

阿里淘系
2021
技术年货

2021 技术人的 百宝黑皮书

2021
新趋势

技术
解决
方案

实战
经验

本书包含

- 淘系最新实战经验&技术解决方案
- 年度必看的技术&行业报告2021新趋势
- 技术人必备工具书单、实用项目、最新paper等

前言

当一项技术逐渐“看不见”的时候，就是这项技术真正走向成熟的时候。

站在2021年时间点的我们，多模态、3D XR、音视频、认知计算与知识图谱等技术已经无声无息地浸润到了我们的生活。

越是身处一线的技术工作者，大概越能深切感知到：技术真正在产业中发挥作用，不是去解决一个惊天动地的大问题，而是去打磨10000个琐碎的小问题。同时在足够的时间和空间里，去不断迭代人类自身与技术、与世界的关系。未来的世界里，技术会像暖流一样浸润到生活的每一处，变得可触摸、可享受、可亲近。

站在天猫双11的第13个年头的大淘宝技术，我们不仅希望为消费者带来购物的快乐，更推动着电商技术不断突破为More fun、More valuable、More future的硬核科技。

同时，我们也关注技术新动向和行业新趋势，不断登顶在创新前沿，去探索数字时代最理想的生活方式。

在这个不断攀索的过程中，我们和你相遇了。

这里是阿里巴巴淘系技术2021一整年的精华技术内容合集，涵盖了大前端、音视频、推荐系统、客户端架构、服务端架构、技术质量、以及3D&AI&AR类等多个技术领域，沉淀了双11大促、淘宝直播、闲鱼APP等多个业务的技术解决方案，并有2021年度必看的技术&行业报告、开源项目、顶会 paper 等多角度知识与价值的输出。

更有趣的，我们今年新增了很多接地气的程序员学习成长经验和人生话题，在似曾相识的人生经验里，能带给你相视一笑的启发和触动。

本书内容页数 860+，全部内容将近 40w 字。希望你喜欢，并分享给身边的朋友。

新年快乐！



扫一扫，关注公众号【淘系技术】
不错过下一年的技术风景

CONTENTS

目录

第一部分

年度精选技术栈内容

后端篇

技术经典总结

谈谈Java接口Result设计	02
DDD系列第五讲：聊聊如何避免写流水账代码	13
浅谈系统实现层面稳定性保障	43
Guava Cache 原理分析与最佳实践	75
一种低延迟的超时中心实现方式	89
Java类加载器 — classloader 的原理及应用	102
如何在业务逻辑当中优雅引入重试机制	113

后端人精彩问答

Q: 怎么提高自己的系统设计和架构理论水平?	118
Q: 公司项目并发量都特小，自己如何实际接触高并发项目?	124
Q: 为什么Go的web框架速度还不如Java?	127

算法篇

技术经典总结

SIGIR2021 超越l2l和向量内积，淘宝新一代召回范式：PDN模型	133
SIGKDD2021 淘宝搜索向量化召回实践	146
情感计算在ugc应用进展	159
EdgeRec：电商信息流的端上推荐系统	171
端智能研发核心套件，MNN 工作台深度解析	191

算法人精彩问答

Q: 计算广告与推荐系统有哪些区别？	204
Q: 目前工业界常用的推荐系统模型有哪些？	207
Q: 2021年，作为算法工程师的你们会在CV业务落地上用Transformer吗？	210

前端篇

技术经典总结

前端智能化看“低代码/无代码”	212
下一代 Web 建站技术栈——Jamstack	221
前端页面双向滚动方案	227
图片上传系统在淘系中的实践	238

前端人精彩问答

Q: 无代码编程会是以后的趋势吗？	249
Q: 2021年了，前端最应该学什么？	259
Q: 如何看待 svelte 这个前端框架？	266

客户端篇

技术经典总结

iOS 性能检测新方式——AnimationHitches	274
淘宝客户端诊断体系升级实践	295
阿里的统一跨端 API 方案 —— Uni API	310
Flutter企业级应用开发在闲鱼的落地	318
一个 Crash 引发的血案	323

测试篇

技术经典总结

软件测试需要学什么？	341
基于强化学习的质量AI在淘系互动业务的实践之路	347
“真正的机器人测试” - 淘宝泛终端机器人自动化测试实践	353
淘系产业级移动应用智能测试最新方案MonkeyBot	361

音视频与图像技术篇

技术经典总结

TaoAudio: 淘宝直播短视频音频解决方案	371
淘宝直播低延迟架构演进和实践	382
淘系的音视频编辑方案：非线性编辑引擎	393
视频内容理解在淘宝逛逛中的应用与落地	406

淘系2021技术资讯重点

2021双11淘系技术九大技术亮点发布	421
淘系发布业界首个基于神经渲染的3D建模产品Object Drawer	423
淘系技术内容互动算法团队入选ACM MM的4篇论文介绍	431
淘系多媒体算法CVPR2021竞赛3项国际冠军方案介绍	437

技术人的通用干货tips

程序命名的原则与重构	445
如何用“心流”提升编码工作效率？	462
如何做好一份程序员的工作汇报ppt？	473
2021阿里淘系工程师推荐书单	480

第二部分

职场成长技巧与解惑

职场成长技巧&故事

从文科生到前端专家 – 在转行时我想过的问题	495
淘系工程师亲述经历：如何避免成为“技术工具人”？	505
仅9天拿下CVPR竞赛冠军，家里的狗狗都改了作息	518

淘系工程师圆桌讨论

小项目需要前后端分离吗？	524
程序员是要专精，还是要广度？	529
程序员如何在业余时间提升自己？	533
程序员如何快速上手一个新项目？	539
“程序员从幼稚到成熟的标志是什么”？	548

一线职场&学习问答实录

Q: 为什么阿里巴巴超级喜欢java开发？	565
Q: 阿里P7、P8、P9级别需要多少年的工作经验，还是因能力而定？	567
Q: 去阿里淘系做技术怎么样？值得2021应届毕业生去吗？	577
Q: 面试官如何判断面试者的机器学习水平？	580
Q: 简历中怎么写「项目经历」最好？为什么？	585
Q: 我想学习后端，后端的学习路线是怎么样子的呢？	591

第三部分

技术人必备工具书籍

前端篇	598
后端篇	599
客户端篇	600
服务端篇	601
测试篇	602
多媒体技术篇	603
计算机视觉篇	604
数据挖掘与数据分析篇	605
推荐系统篇	606
3D&AR篇	607

第四部分

GitHub上的实用练手项目

推荐21个GitHub上好用又有趣的移动端项目（涵盖初中高阶）	609
推荐26个GitHub上好玩又有挑战的前端项目（涵盖初中高阶）	618
推荐10个GitHub上适合练手的后端项目（涵盖初中高阶）	630

第五部分

2021淘系A类顶会论文精选

数据挖掘与机器学习领域

ICDE 2021

Large-scale Fake Click Detection for E-commerce Recommendation Systems 638

ATNN: Adversarial Two-Tower Neural Network for New Item's Popularity Prediction in Ecommerce 650

Billion-scale Pre-trained E-commerce Product Knowledge Graph Model 662

KDD 2021

SEMI: A Sequential Multi-Modal Information Transfer Network for E-Commerce MicroVideo Recommendations 674

AliCoCo2: Commonsense Knowledge Extraction, Representation and Application in Ecommerce 685

Live-Streaming Fraud Detection: A Heterogeneous Graph Neural Network Approach 694

Embedding-based Product Retrieval in Taobao Search 703

机器学习、计算机视觉及多媒体领域

NIPS 2021

Mining the Benefits of Two-stage and One-stage HOI Detection 712

ACM MM 2021

Understanding Chinese Video and Language via Contrastive Multimodal Pre-Training 724

Pre-training Graph Transformer with Multimodal Side Information for Recommendation 734

Shape Controllable Virtual Try-on for Underwear Models 743

CVPR 2021

Residential floor plan recognition and reconstruction 753

ICCV 2021

Conditional Diffusion for Interactive Segmentation 769

交叉/综合/新兴领域

AAAI 2021

Exploiting Diverse Characteristics and Adversarial Ambivalence for Domain Adaptive Segmentation 779

Personalized Adaptive Meta Learning for Cold-start User Preference Prediction 788

Capturing Delayed Feedback in Conversion Rate Prediction via Elapsed-Time Sampling 797

KGDet: Keypoint-Guided Fashion Detection 805

数据库、知识管理、信息检索领域

CIKM 2021

Gated Heterogeneous Graph Representation Learning for Shop Search in E-Commerce 814

VLDB 2021

EdgeDIPN: a Unified Deep Intent Prediction Network Deployed at the Edge 818

SIGIR 2021

A General Method For Automatic Discovery of Powerful Interactions In Click-Through Rate Prediction 827

Path-based Deep Network for Candidate Item Matching in Recommenders 837

第六部分

年度必看的技术&行业报告

2021热门技术新动向报告	853
2021热门行业新趋势报告	856
2021热门主题特别篇：你值得关注的【元宇宙】	858
附录	860

第一部分 年度精选技术栈内容

- 后端篇
- 算法篇
- 前端篇
- 客户篇
- 测试篇
- 音视频与图像技术篇
- 淘系2021技术资讯重点
- 技术人的通用干货tips

后端篇

• 谈谈Java接口Result设计

- 作者：书牧
- 出品：淘系技术

这篇文章酝酿了很久，一直想写，却一直觉得似乎要讲的东西有点杂，又不是很容易讲清楚，又怕争议的地方很多，就一拖再拖。但是，每次看到不少遇到跟这个设计相关导致的问题，又忍不住跟人讨论，但又很难一次说清楚，于是总后悔没有及早把自己的观点写成文章。不管怎样，观点还是要表达的，无论对错。

故障的推手——“Result”

先说结论：接口方法，尤其是对外HSF（开源版本即dubbo）api，接口异常建议不要使用Result，而应该使用异常。阿里内部的java编码，已经习惯性对外API一股脑儿使用“Result”设计——这是导致许多故障的重要原因！

一个简化的例子

```
1 // 用户查询的HSF服务API，使用了Result做为返回结果
2 public interface UserService {
3     Result<User> getUserById(Long userId);
4 }
```

```
1 // 一段客户端应用facade的调用示例。读写缓存逻辑部分省略，仅做示意
2 public User testGetUser(Long userId) {
3     String userKey = "userId-" + userId;
4     // 先查缓存，如果命中则返回缓存中的user
5     // cacheManager.get(123, userKey);
6     // ...
7
8     try{
9         Result<User> result = userService.getUserById(userId);
10        if (result.isSuccess()) {
11            cacheManager.put(123, userKey, result.getData());
12        }
13    }
```

```
12     return result.getData();
13 }
14 // 否则缓存空对象，代表用户不存在
15 cacheManager.put(123, userKey, NullCacheObject.getInstance(), 3600);
16 return null;
17 } catch (Exception e) {
18     // TODO log
19     throw new DemoException("getUserById error. userId=" + userId, e);
20 }
21 }
```

上面的代码很简单，客户端应用对User查询服务做了个缓存。有些同学可能一眼就看出来，这里隐藏的bug：第10行的“result.isSuccess()”为false的实际含义是什么？是服务端系统异常吗？还是用户不存在？光看API是很难确定的。不得不去找服务提供方或文档确认其逻辑，根据错误码进行区分。如果是服务端系统异常，那么第15行将导致线上bug，因为后续1小时对该用户的请求都认为用户不存在了。

严谨点的写法

如果要写正确逻辑，那么代码可能会变成这样：

```
1 public User testGetUser(Long userId) {
2     String userKey = "userId-" + userId;
3     // 先查缓存，如果命中则返回缓存中的user
4     // cacheManager.get(123, userKey);
5     // ...
6
7     try{
8         Result<User> result = userService.getUserById(userId);
9         if (result.isSuccess()) {
10             cacheManager.put(123, userKey, result.getData());
11             return result.getData();
12         }
13         if ("USER_NOT_FOUND".equals(result.getCode())) {
14             // 否则缓存空对象，代表用户不存在
15             cacheManager.put(123, userKey, NullCacheObject.getInstance(), 3600);
16         } else {
17             // 可能是SYSTEM_ERROR、DB_ERROR等一些系统性的异常，TODO log
18             throw new DemoException("getUserById error. userId=" + userId + ", result=" + result);
19         }
20     } catch (Exception e) {
21         // TODO log
22         throw new DemoException("getUserById error. userId=" + userId, e);
23     }
24 }
```

```
19     }
20   } catch (DemoException e) {
21     throw e;
22   } catch (Exception e) {
23     // TODO log
24     throw new DemoException("getUserById error. userId=" + userId, e);
25   }
26   return null;
27 }
```

很显然，代码变得复杂起来了，加上对外部调用的try catch异常处理，实际代码变相当复杂繁琐。

不使用Result的例子

```
1  public interface UserService {
2    User getUserById(Long userId) throws DemoAppException;
3  }
```

```
1  public User testGetUser(Long userId) {
2    String userKey = "userId-" + userId;
3    // 先查缓存，如果命中则返回缓存中的user
4    // cacheManager.get(123, userKey);
5    // ...
6
7    try {
8      User user = userService.getUserById(userId);
9      if (user != null) {
10        cacheManager.put(123, userKey, user);
11        return user;
12      } else {
13        // 否则缓存空对象，代表用户不存在
14        cacheManager.put(123, userKey, NullCacheObject.getInstance(), 3600);
15        return null;
16      }
17    } catch (Exception e) {
18      // TODO log
19      throw new DemoException("getUserById error. userId=" + userId, e);
20    }
21  }
```

这样一看，代码简洁清晰很多，也更符合对普通API的调用习惯。

使用Result的几个问题

1. 调用成本高：虽然通过对依赖的API深入了解异常设计，可以写出严谨的代码以避免出现bug，但是简单的逻辑，代码却变得复杂。换言之，调用的成本变高。但是很可惜，我们忘记判断而写成“一个简化的例子”这样是往日常事。
2. 无意义错误码：SYSTEM_ERROR、DB_ERROR等系统异常的错误码，虽然放在Result中了，但是调用方除了日志和监控作用外，业务逻辑永远不会关心，也永远处理不了。而些错误码的处理分支，实际与抛异常的处理逻辑一样。既然如此，为何要将这些错误码放在返回值里？

关于阿里巴巴开发规约

我们看《阿里巴巴Java开发手册》的“异常处理”小节第13条：

【推荐】对于公司外的http/api开放接口必须使用“错误码”；跨应用间HSF调用优先考虑使用Result方式，封装isSuccess()方法、“错误码”、“错误简短信息”；而应用内部推荐异常抛出。

这条推荐非常具有误导性，在2016年孤尽对于这条规范进行调研时的帖子：《【开发规约热议投票02】HSF服务接口定义时，是Result+isSuccess方式返回，还是抛异常的方式？》有部分同学不建议使用Result，但大部分同学推荐了Result的做法。

为什么说这条规约具有误导性？

因为这个问题本身没有讲清楚“对什么东西的处理”要用Result还是异常的方式，即这里没有讲清楚我们要解决的问题是什么。事实上我们常说的“失败”，往往混淆了2种含义：

1. 系统异常：比如网络超时、DB异常、缓存超时等，调用方一般不太可能基于这些错误类型做不同的业务逻辑，常用于日志和监控，方便定位排查。
2. 业务状态：比如业务规则拦截导致的失败，比如发权益时库存不足、用户限领等，为方便后文叙述和理解，暂时称为“业务失败”。这类“失败”，从机器层面来看，严格来说不能算做是失败，这只是一种正常的业务结果，这和“调用成功”这个业务结果对系统来说没有任何区别，只是一个业务状态而已。调用方往往可能关心对应的错误码，以完成不同的业务逻辑。

有经验的开发，都会意识到这2种含义的区别，这对于帮助我们理解接口的异常设计非常重要！对这条开发规约而言，如果是第2种，并没有什么大的问题，但如果是第1种，我则持相反的意见，因为这违背了java语言的基本设计，不符合java编码直觉，会潜移默化造成前面案例所示的理解和使用成本的问题。



为什么针对HSF?

当我们讨论要用Result代替Exception时，经常会以这是HSF接口为由，因为性能开销等等。我们常说HSF这种RPC框架，设计的目的是为了看起来像本地调用。那么，这个“看起来像本地调用”到底指的是哪方面像呢？显然，编码时像，运行时不像。所以我们写调用HSF接口的代码时，感觉像在调用本地方法，那么我们的编码直觉和习惯也都应该是符合java的规范的。因此，至少有几点理由，对于系统异常，我们的HSF接口更应该使用Exception，而非Result的方式：

1. 只有同样遵循本地方法调用的设计，来设计HSF的api，才能更好做到“像本地调用一样”，更符合HSF设计的初衷。
2. HSF接口是往往用于对外部团队提供服务，更应该遵循java语法的设计，提供清晰的接口语义，降低调用方的使用成本，减少出bug的概率。
3. **Result**并无统一规范，而Exception则是语言标准 有利于中间件、框架代码的监控发现和异常重试等逻辑生效。

当然，由于“运行时不像”，对于HSF封装带来的抽象泄露，我们在使用异常时，需要关注几点问题：

1. 异常要在接口显式声明，否则客户端可能会反序列化失败。
2. 尽可能不带原始堆栈，否则客户端也可能反序列化失败，或者堆栈过大导致性能问题。可以考虑异常中定义错误码以方便定位问题。

结论

无论是HSF接口，还是内部的API，都应该遵循java语言的编码直觉和习惯，业务结果（无论成功还是失败）都应该通过返回值返回，而系统异常，则应该使用抛出Exception的方式来实现。

关于Checked Exception

讲到这里，我们发现，java的Checked Exception的设计，作用上和反映业务失败的Result很像。Result是强制调用方进行判断和识别，并根据不同的错误码进行判断和处理。而Checked Exception也是强制调用方进行处理，并且可能要对不同的异常做不同的处理。但是，基于前面的结论，业务失败应该通过返回值来表达，而不是异常；而异常是不应该用于做业务逻辑判断的，那么java的Checked Exception就变成奇怪的存在了。这里我明确我的观点，我们应该尽可能不使用Checked Exception。另外，《Thinking in Java》的作者 Bruce Eckel就曾经公开表示，Java语言中的Checked Exception是一个错误的决定，Java应该移除它。C#之父Anders Hejlsberg也认同这个观点，因此C#中是没有Checked Exception的。

Result 的实质是什么？



我们看看一个java方法的签名（省略修饰符部分）：

1. 方法名：用于表达这个方法的功能
2. 参数：方法的输入
3. 返回值类型：方法的输出
4. 异常：方法中意外出现的错误

所以，返回值和方法功能必须是配套的，返回值类型，就是这个方法的功能执行结果的准确表达，即返回值必须正好就是当前这个方法要做的事情的结果，必须满足这个方法语义，而不应该有超出这个语义外的东西存在。而异常，所说的“意外”，则是指超出这个方法语义之外的部分。这几句话有点拗口，举个例子来说，上面这个用户接口，语义就是要通过用户id查询用户，那么当服务端发生DB超时错误时，对于“通过用户id查询用户”这个语义来说，“DB超时错误”没有任何意义，使用异常是恰好合适的，如果我们把这个错误做为错误码放在返回值的Result里，那么就是增加了这个方法的使用成本。

Result的由来

到底为什么会有“Result”这样的东西诞生呢？如果设计的方法返回值是Result类型，那么它必须能准确反应这个方法调用的结果。实际上，以上面的例子为例，这个时候的Result就是User类本身，User.status相当于Result.code。这听起来可能有点和直觉不符，这是为什么？

```
1 public class UserRegisterResult {
2     private String errorCode;
3     private String errorMsg;
4     private Long userId;
5     // ...
6
7     public boolean isSuccess() {
8         return errorCode == null;
9     }
10    // ...
11 }
```



```
1 UserRegisterResult registerUser(User user) throws DemoAppException;
```

我们再来看看上面这个“注册用户”的方法声明，会发现，这个方法定义一个Result显得很合适。这是因为前一个例子，我们的方法是一个查询方法，返回值刚好可以用领域对象类型本身，而这个“注册用户”的方法，显然没有现成合适的类型可以使用，所以需要定义一个新的类型来表达方法的执行结果。看到这里，我们会以为，对于“写”与“读”类型的方法有所差异，但实际上，对于java语言或者机器来说，并无二致，第二个方法UserRegisterResult的和第一个方法的User是同等地位。所以，最重要的还是一点：需要有一个合适的类型，做为返回值，用于准确表达方法执行的功能结果。而偏“写”类型，或者带业务校验的读接口，往往因为没有现成的类型可用，为了方便，常常会使用Result来代替。

是否有必要统一Result?

讲到这里，想想，当我们这种“需要Result”的方法有多个时，我们会说“我需要一个统一的Result类”时，实际上说的什么呢？

- 1.我希望各种接口方法都统一同样的Result，方便使用
- 2.我希望有个类复用errorCode、errorMsg以及相关的getter/setter等代码

显然，第1点理由经不起推敲，为何“统一就方便使用”了？如果各种方法返回类型都一样，那就违背了“返回值要和方法功能配套”的结论，也不符合高内聚的设计原则。恰相反，返回值越是设计得专用，对调用方来说理解和使用成本越低。所以，我们实际想要的，仅仅是如何“偷懒”，也就是第2点理由。所以我们真正要做的是，只是在当前领域范围内，如何既满足让每个方法返回值专用以便使用，同时又可以偷懒复用部分代码即可。因此，绝不要求大家都统一使用同一个Result类型。

接口返回设计建议

根据前文的结论，我们知道，对于接口方法的返回值和异常处理，最重要的是需要遵循方法的语义进行设计。以下是我梳理的一些设计上的原则和建议。

对响应合理分类

接口响应按有业务结果和未知业务结果分类，业务结果不管是业务成功还是业务规则导致的失败，都通过返回值返回；未知结果一般是系统性的异常导致，不要通过返回值错误码表达，而是通过抛出异常来表达。这里最关键一点，就是如何理解和区分某个“失败”是属于业务失败，还是属于系统异常。由于有时候这个区分并不是很容易，我们可以有一个比较简单的判断标准来确定：

1.如果一个错误，调用方只能通过人工介入的方式才能恢复，比如修改代码、改配置，或数据订正等处理，则必然属于异常

2.如果调用方无法使用代码逻辑处理消化使得自动恢复，而是只能通过重试的方式，依赖下游的恢复才能恢复，则属于异常

找到合适的场景

普通查询接口，如无必要，不要使用Result包装返回值。可以简单分为3类做为参考：

- 普通读接口

查询结果即是领域对象，无其他业务规则导致的失败：建议直接用领域对象类型做为返回值。如：

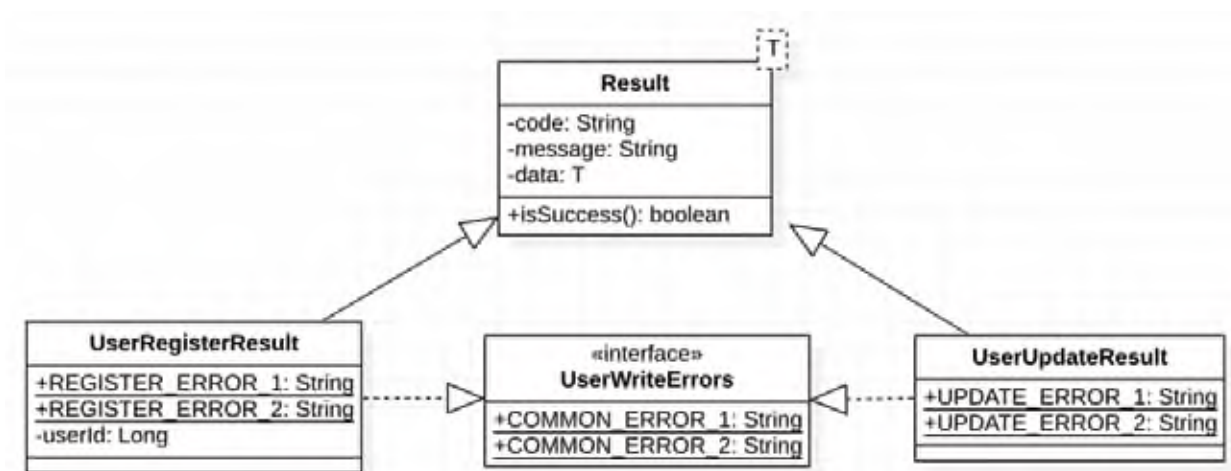
```
1 User getUserById(Long userId) throws DemoAppException;
```

- 写接口

或者带业务规则的读接口：

- 1.理想情况是专门封装一个返回值类，以降低调用方的使用成本。
- 2.可考虑将返回值类继承Result，以复用errorCode和errorMsg等代码，减轻开发工作量。但注意这不是必要的。
- 3.将本方法的错误码，直接定义到这个返回值类上（高内聚原则）。
- 4.若有多个方法有共同的错误码，可以考虑通过将这部分错误码定义到一个Interface中，然后实现该接口。

```
1 // UserRegisterResult、UserUpdateResult可以继承Result类，减少工作量，但调用方不需要感知Result类的存在
2 UserRegisterResult registerUser(User user) throws DemoAppException;
3
4 UserUpdateResult updateUser(User user) throws DemoAppException;
```



- 带业务规则的领域对象读接口

完全遵循上面第2点，会给方法提供者带来一定的开发成本，权衡情况下可以考虑，套Result包装领域对象做为返回值。注意，对外不建议，可考虑用于内部方法。如下接口，“没有权限”是一个正常的业务失败，调用方可能会判断并做一定的业务逻辑处理：

```
1 // 查询有效用户，如果用户存在但状态非有效状态则返回“用户状态错误”的错误码，如果不存在则返回null
2 Result<User> getEffectiveUserWithStatusCheck(Long userId) throws DemoAppException;
```

内外部区分

对外接口，尤其是HSF，由于变更成本高，更要遵循前面的原则；内部方法，方法众多，如果完全遵循需要编码成本，这里需要做权衡，根据代码规模和发展阶段不断重构和调整即可。

避免直接包装原生类型

我们对外的接口，返回值要避免出现直接使用Result包装一个原生类型。比如：

```
1 Result<Long> registerUser(User user) throws DemoAppException;
```

这样设计导致的结果是，扩展性很差。如果registerUser方法需要增加返回除了userId以外的其他字段时，就面临几个选择：

- 1.让Result支持扩展参数，通过map来传递额外字段：可读性和使用成本很高
- 2.开发一个新的registerUser方法：显然，成本很高

避免所有错误码定义在一个类中

有人建议，做一个全局的错误码定义，以做统一，方便排查和定位。但这样做真的方便吗？这样做实际上有几个问题：

- 1.完全违背了高内聚、低耦合的设计原则。这个“统一的定义”将与各个域都有耦合，同时对于某单个接口而言，则不够内聚。
- 2.这个统一定义的错误码，一定会爆炸式增长，即便我们对其进行分类（非常依赖人的经验），迟早也会变得难以维护和理解。
- 3.不要将系统异常类的错误码和业务失败错误码放在一起，这点其实和方法响应分类设计是一回事。

我们在设计拉菲2权益平台的错误码时，就犯了这样的错误。现在这个“统一的”错误码已经超过400个，揉合了管理域、投发放域、离线域等各种不同域的业务失败、系统异常的错误码，不要说调用方，即便我们自己，也梳理不清楚了。而实际上，每个域、每个方法自己的业务失败是非常有限的，它的增长一定是随着业务需求本身的变化而增长的。现在如果有个业务方来问我，拉菲2的发放接口，有哪些错误码（这问的实际是业务失败，他也只关心业务失败），我几乎难以回答。很可惜，这块目前即便重构，难度也很大。

异常处理机制

- 异常错误码

前面我们讲到，即便是抛异常的形式，我们也可以为我们的异常类设计错误码，异常错误码的增加会很快，往往也和当前业务语义无关，因此千万不要和业务失败的错误码定义在一起。异常内的错误码主要用于日志、监控等，核心原则就是，要方便定位问题。

- 避免层层try catch

到处充满异常处理的代码，会导致整个程序可读性变差，写起来也非常繁琐，可以遵循一定的原则：

- 1.在原始发生错误的地方try catch，比如调用HSF接口的Facade层代码，主要目的是为了记录原始的错误以及出入参，方便定位问题，一般会打日志，并转换成本应用的异常类上抛
- 2.在应用的最顶层catch异常，打印统一日志，并根据“为什么针对HSF？”小节中的建议，处理成合适的异常后再抛出。对于HSF接口，可以直接实现HSF的“ServerFilter”来统一在框架层面处理。
- 3.中间层的代码，不必再层层catch，比如domain层，可以让代码逻辑更加清晰。

- 异常错误码

抛异常的场景，除了前面说的系统性异常外，参数错误也推荐使用异常。原因如下：

1.参数正确一般是我们当前上下文执行的前提条件，我们一般可以使用assert来保证其正确。即我们的后续逻辑是认为，当前的参数是不可能错误的，我们没必要为此写过繁琐的防御性代码。

2.一旦发生参数错误，则一定是调用方有代码bug，或者配置bug，应该通过抛出异常的方式，充分提前在开发或测试阶段暴露。

3.参数错误对调用方来说，是无法处理的，程序不可能自动恢复，一定是会需要人工介入才可能恢复，调用方不可能会“判断如果是xx参数错误，我就做某个业务逻辑”这样的代码，因此通过返回值定义参数错误码没有意义。

- 系统异常和业务结果转换

系统性异常并非一定是异常，因为有些层可能有能力处理某些异常，比如对于弱依赖的接口，异常是可以吞掉，转换成一个业务结果；相反，有些接口返回的一些业务失败，但调用方认为该业务失败不可能出现，出现也无法处理，那么这一层可以将其转换成异常。

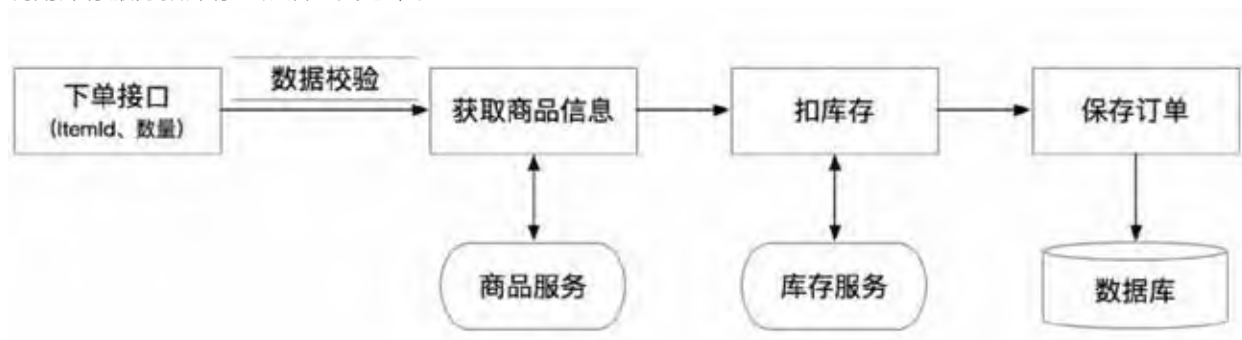
• DDD系列第五讲： 聊聊如何避免写流水账代码

- 作者：殷浩
- 出品：淘系技术

过去一年里我们团队做了大量的老系统重构和迁移，其中有大量的代码属于流水账代码，通常能看到是开发在对外的API接口里直接写业务逻辑代码，或者在一个服务里大量的堆接口，导致业务逻辑实际无法收敛，接口复用性比较差。所以这讲主要想系统性的解释一下如何通过DDD的重构，将原有的流水账代码改造为逻辑清晰、职责分明的模块。

案例简介

这里举一个简单的常见案例：下单链路。假设我们在做一个checkout接口，需要做各种校验、查询商品信息、调用库存服务扣库存、然后生成订单：



一个比较典型的代码如下：

```
1  @RestController
2  @RequestMapping("/")
3  public class CheckoutController {
4
5      @Resource
6      private ItemService itemService;
7
8      @Resource
9      private InventoryService inventoryService;
10
11     @Resource
12     private OrderRepository orderRepository;
13 }
```

```
14 @PostMapping("checkout")
15 public Result<OrderDO> checkout(Long itemId, Integer quantity) {
16     // 1) Session管理
17     Long userId = SessionUtils.getLoginUserId();
18     if (userId <= 0) {
19         return Result.fail("Not Logged In");
20     }
21
22     // 2) 参数校验
23     if (itemId <= 0 || quantity <= 0 || quantity >= 1000) {
24         return Result.fail("Invalid Args");
25     }
26
27     // 3) 外部数据补全
28     ItemDO item = itemService.getItem(itemId);
29     if (item == null) {
30         return Result.fail("Item Not Found");
31     }
32
33     // 4) 调用外部服务
34     boolean withholdSuccess = inventoryService.withhold(itemId, quantity);
35     if (!withholdSuccess) {
36         return Result.fail("Inventory not enough");
37     }
38
39     // 5) 领域计算
40     Long cost = item.getPriceInCents() * quantity;
41
42     // 6) 领域对象操作
43     OrderDO order = new OrderDO();
44     order.setItemId(itemId);
45     order.setBuyerId(userId);
46     order.setSellerId(item.getSellerId());
47     order.setCount(quantity);
48     order.setTotalCost(cost);
49
50     // 7) 数据持久化
51     orderRepository.createOrder(order);
52
53     // 8) 返回
54     return Result.success(order);
55 }
56 }
```

为什么这种典型的流水账代码在实际应用中会有问题呢？其本质问题是违背了SRP（Single Responsibility Principle）单一职责原则。这段代码里混杂了业务计算、校验逻辑、基础设施、和通信协议等，在未来无论哪一部分的逻辑变更都会直接影响到这段代码，长期当后人不断的在上面叠加新的逻辑时，会造成代码复杂度增加、逻辑分支越来越多，最终造成bug或者没人敢重构的历史包袱。

所以我们才需要用DDD的分层思想去重构一下以上的代码，通过不同的代码分层和规范，拆分出逻辑清晰，职责明确的分层和模块，也便于一些通用能力的沉淀。

主要的几个步骤分为：

分离出独立的Interface接口层，负责处理网络协议相关的逻辑

从真实业务场景中，找出具体用例（Use Cases），然后将具体用例通过专用的Command指令、Query查询、和Event事件对象来承接

分离出独立的Application应用层，负责业务流程的编排，响应Command、Query和Event。每个应用层的方法应该代表整个业务流程中的一个节点

处理一些跨层的横切关注点，如鉴权、异常处理、校验、缓存、日志等

下面会针对每个点做详细的解释。

Interface接口层

随着REST和MVC架构的普及，经常能看到开发同学直接在Controller中写业务逻辑，如上面的典型案例，但实际上MVC Controller不是唯一的重灾区。以下的几种常见的代码写法通常都可能包含了同样的问题：

- HTTP 框架：如Spring MVC框架，Spring Cloud等
- RPC 框架：如Dubbo、HSF、gRPC等
- 消息队列MQ的“消费者”：比如JMS的 onMessage，RocketMQ的MessageListener等
- Socket通信：Socket通信的receive、WebSocket的onMessage等
- 文件系统：WatcherService等
- 分布式任务调度：SchedulerX等

这些的方法都有一个共同的点就是都有自己的网络协议，而如果我们业务代码和网络协议混杂在一起，则会直接导致代码跟网络协议绑定，无法被复用。

所以，在DDD的分层架构中，我们单独会抽取出来Interface接口层，作为所有对外的门户，将网络协议和业务逻辑解耦。

接口层的组成

接口层主要由以下几个功能组成：

- 1.网络协议的转化：通常这个已经由各种框架给封装掉了，我们需要构建的类要么是被注解的bean，要么是继承了某个接口的bean。
- 2.统一鉴权：比如在一些需要AppKey+Secret的场景，需要针对某个租户做鉴权的，包括一些加密串的校验
- 3.Session管理：一般在面向用户的接口或者有登陆态的，通过Session或者RPC上下文可以拿到当前调用的用户，以便传递给下游服务。
- 4.限流配置：对接口做限流避免大流量打到下游服务
- 5.前置缓存：针对变更不是很频繁的只读场景，可以前置结果缓存到接口层
- 6.异常处理：通常在接口层要避免将异常直接暴露给调用端，所以需要在接口层做统一的异常捕获，转化为调用端可以理解的数据格式
- 7.日志：在接口层打调用日志，用来做统计和debug等。一般微服务框架可能都直接包含了这些功能。

当然，如果有一个独立的网关设施/应用，则可以抽离出鉴权、Session、限流、日志等逻辑，但是目前来看API网关也只能解决一部分的功能，即使在有API网关的场景下，应用里独立的接口层还是有必要的。

在interface层，鉴权、Session、限流、缓存、日志等都比较直接，只有一个异常处理的点需要重点说下。

返回值和异常处理规范，Result vs Exception

注：这部分主要还是面向REST和RPC接口，其他的协议需要根据协议的规范产生返回值。

在我见过的一些代码里，接口的返回值比较多样化，有些直接返回DTO甚至DO，另一些返回Result。

接口层的核心价值是对外，所以如果只是返回DTO或DO会不可避免的面临异常和错误栈泄漏到使用方的情况，包括错误栈被序列化反序列化的消耗。所以，这里提出一个规范：

规范：Interface层的HTTP和RPC接口，返回值为Result，捕捉所有异常

规范：Application层的所有接口返回值为DTO，不负责处理异常

Application层的具体规范等下再讲，在这里先展示Interface层的逻辑。

举个例子：

```
1  @PostMapping("checkout")
2  public Result<OrderDTO> checkout(Long itemId, Integer quantity) {
3      try {
4          CheckoutCommand cmd = new CheckoutCommand();
5          OrderDTO orderDTO = checkoutService.checkout(cmd);
6          return Result.success(orderDTO);
7      } catch (ConstraintViolationException cve) {
8          // 捕捉一些特殊异常，比如Validation异常
9          return Result.fail(cve.getMessage());
10     } catch (Exception e) {
11         // 兜底异常捕获
12         return Result.fail(e.getMessage());
13     }
14 }
```

当然，每个接口都要写异常处理逻辑会比较烦，所以可以用AOP做个注解

```
1  @Target(ElementType.METHOD)
2  @Retention(RetentionPolicy.RUNTIME)
3  public @interface ResultHandler {
4
5  }
6
7  @Aspect
8  @Component
9  public class ResultAspect {
10     @Around("@annotation(ResultHandler)")
11     public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {
12         Object proceed = null;
13         try {
14             proceed = joinPoint.proceed();
15         } catch (ConstraintViolationException cve) {
16             return Result.fail(cve.getMessage());
17         } catch (Exception e) {
18             return Result.fail(e.getMessage());
19         }
20         return proceed;
21     }
22 }
```

然后最终代码则简化为：

```
1  @PostMapping("checkout")
2  @ExceptionHandler
3  public Result<OrderDTO> checkout(Long itemId, Integer quantity) {
4      CheckoutCommand cmd = new CheckoutCommand();
5      OrderDTO orderDTO = checkoutService.checkout(cmd);
6      return Result.success(orderDTO);
7  }
```

接口层的接口的数量和业务间的隔离

在传统REST和RPC的接口规范中，通常一个领域的接口，无论是REST的Resource资源的GET/POST/DELETE，还是RPC的方法，是追求相对固定的，统一的，而且会追求统一个领域的方法放在一个领域的服务或Controller中。

但是我发现在实际做业务的过程中，特别是当支撑的上游业务比较多时，刻意去追求接口的统一通常会导致方法中的参数膨胀，或者导致方法的膨胀。举个例子：假设有一个宠物卡和一个亲子卡的业务公用一个开卡服务，但是宠物需要传入宠物类型，亲子的需要传入宝宝年龄。

```
1  // 可以是RPC Provider 或者 Controller
2  public interface CardService {
3
4      // 1) 统一接口，参数膨胀
5      Result openCard(int petType, int babyAge);
6
7      // 2) 统一泛化接口，参数语意丢失
8      Result openCardV2(Map<String, Object> params);
9
10     // 3) 不泛化，同一个类里的接口膨胀
11     Result openPetCard(int petType);
12     Result openBabyCard(int babyAge);
13 }
```

可以看出来，无论是怎么操作，都有可能导致CardService这个服务未来越来越难以维护，方法越来越多，一个业务的变更有可能会整个服务/Controller的变更，最终变得无法维护。我曾经参与过的一个服务，提供了几十个方法，上百万行代码，可想而知无论是使用方对接口的理解成本还是对代码的维护成本都是极高的。所以，这里提出另一个规范：

规范：Interface层的HTTP和RPC接口，返回值为Result，捕捉所有异常

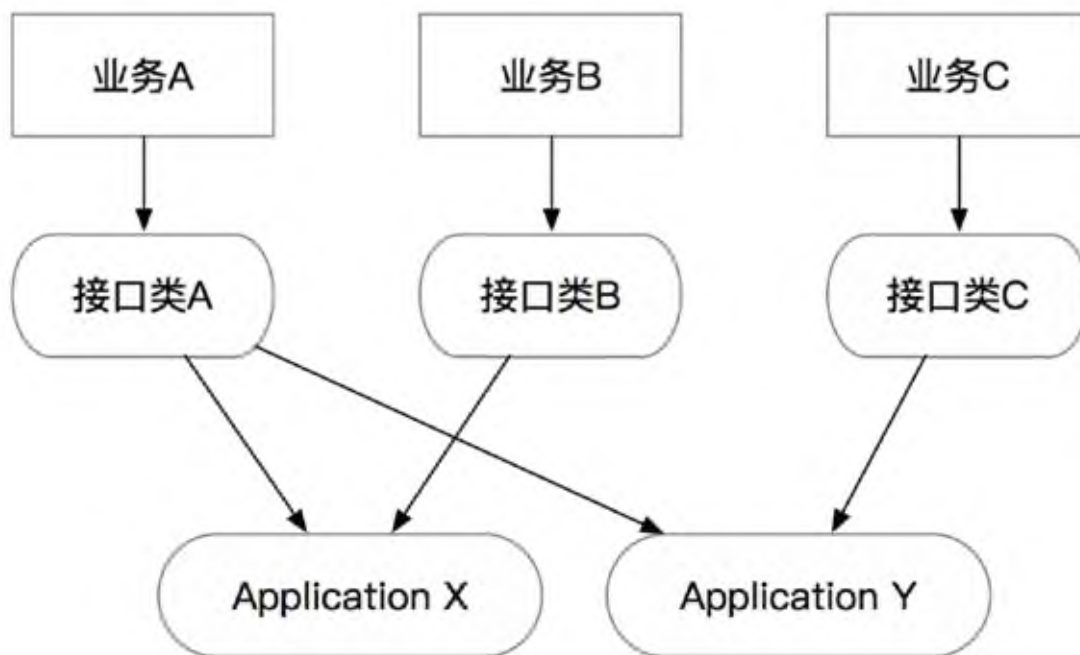
规范：Application层的所有接口返回值为DTO，不负责处理异常

基于上面的这个规范，可以发现宠物卡和亲子卡虽然看起来像是类似的需求，但并非是“同样需求”的，可以预见到在未来的某个时刻，这两个业务的需求和需要提供的接口会越走越远，所以需要将这两个接口类拆分开：

```
1 public interface PetCardService {  
2     Result openPetCard(int petType);  
3 }  
4  
5 public interface BabyCardService {  
6     Result openBabyCard(int babyAge);  
7 }
```

这个的好处是符合了Single Responsibility Principle单一职责原则，也就是说一个接口类仅仅会因为一个（或一类）业务的变化而变化。一个建议是当一个现有的接口类过度膨胀时，可以考虑对接口类做拆分，拆分原则和SRP一致。

也许会有人问，如果按照这种做法，会不会产生大量的接口类，导致代码逻辑重复？答案是不会，因为在DDD分层架构里，接口类的核心作用仅仅是协议层，每类业务的协议可以是不同的，而真实的业务逻辑会沉淀到应用层。也就是说Interface和Application的关系是多对多的：



因为业务需求是快速变化的，所以接口层也要跟着快速变化，通过独立的接口层可以避免业务间相互影响，但我们希望相对稳定的是Application层的逻辑。所以我们接下来看一下Application层的一些规范。

Application层

Application层的组成部分

Application层的几个核心类：

- ApplicationService应用服务：最核心的类，负责业务流程的编排，但本身不负责任何业务逻辑
- DTO Assembler：负责将内部领域模型转化为可对外的DTO
- Command、Query、Event对象：作为ApplicationService的入参
- 返回的DTO：作为ApplicationService的出参

Application层最核心的对象是ApplicationService，它的核心功能是承接“业务流程”。但是在讲ApplicationService的规范之前，必须要先重点的讲几个特殊类型的对象，即Command、Query和Event。

Command、Query、Event对象

从本质上来看，这几种对象都是Value Object，但是从语义上来看有比较大的差异：

- Command指令：指调用方明确想让系统操作的指令，其预期是对一个系统有影响，也就是写操作。通常来讲指令需要有一个明确的返回值（如同步的操作结果，或异步的指令已经被接受）。
- Query查询：指调用方明确想查询的东西，包括查询参数、过滤、分页等条件，其预期是对一个系统的数据完全不影响的，也就是只读操作。
- Event事件：指一件已经发生过的既有事实，需要系统根据这个事实作出改变或者响应的，通常事件处理都会有一定的写操作。事件处理器不会有返回值。这里需要注意一下的是，Application层的Event概念和Domain层的DomainEvent是类似的概念，但不一定是同一回事，这里的Event更多是外部一种通知机制而已。

简单总结下：

	Command	Query	Event
语意	”希望“能触发的操作	各种条件的查询	已经发生过的事情
读/写	写	只读	通常是写
返回值	DTO 或 Boolean	DTO 或 Collection	Void

- 为什么要用CQE对象？

通常在很多代码里，能看到接口上有多个参数，比如上文中的案例：

```
1 Result<OrderDO> checkout(Long itemId, Integer quantity);
```

如果需要在接口上增加参数，考虑到向前兼容，则需要增加一个方法：

```
1 Result<OrderDO> checkout(Long itemId, Integer quantity);  
2 Result<OrderDO> checkout(Long itemId, Integer quantity, Integer channel);
```

或者常见的查询方法，由于条件的不同导致多个方法：

```
1 List<OrderDO> queryById(Long itemId);  
2 List<OrderDO> queryBySellerId(Long sellerId);  
3 List<OrderDO> queryBySellerIdWithPage(Long sellerId, int currentPage, int pageSize);
```

可以看出来，传统的接口写法有几个问题：

- 1.接口膨胀：一个查询条件一个方法
- 2.难以扩展：每新增一个参数都有可能需要调用方升级
- 3.难以测试：接口一多，职责随之变得繁杂，业务场景各异，测试用例难以维护

但是另外一个最重要的问题是：这种类型的参数罗列，本身没有任何业务上的”语意“，只是一堆参数而已，无法明确的表达出来意图。

- CQE的规范

所以在Application层的接口里，强力建议的一个规范是：

规范：ApplicationService的接口入参只能是一个Command、Query或Event对象，CQE对象需要能代表当前方法的语意。唯一可以的例外是根据单一ID查询的情况，可以省略掉一个Query对象的创建

按照上面的规范，实现案例是：

```
1 public interface CheckoutService {
2     OrderDTO checkout(@Valid CheckoutCommand cmd);
3     List<OrderDTO> query(OrderQuery query);
4     OrderDTO getOrder(Long orderId); // 注意单一ID查询可以不用Query
5 }
6
7 @Data
8 public class CheckoutCommand {
9     private Long userId;
10    private Long itemId;
11    private Integer quantity;
12 }
13
14 @Data
15 public class OrderQuery {
16     private Long sellerId;
17     private Long itemId;
18     private int currentPage;
19     private int pageSize;
20 }
```

这个规范的好处是：提升了接口的稳定性、降低低级的重复，并且让接口入参更加语义化。

• CQE vs DTO

从上面的代码能看出来，ApplicationService的入参是CQE对象，但是出参却是一个DTO，从代码格式上来看都是简单的POJO对象，那么他们之间有什么区别呢？

- CQE：CQE对象是ApplicationService的输入，是有明确的”意图“的，所以这个对象必须保证其”正确性“。
- DTO：DTO对象只是数据容器，只是为了和外部交互，所以本身不包含任何逻辑，只是贫血对象。

但可能最重要的一点：因为CQE是”意图“，所以CQE对象在理论上可以有”无限“个，每个代表不同的意图；但是DTO作为模型数据容器，和模型一一对应，所以是有限的。

• CQE的校验

CQE作为ApplicationService的输入，必须保证其正确性，那么这个校验是放在哪里呢？

在最早的代码里，曾经有这样的校验逻辑，当时写在了服务里：

```
1  if (itemId <= 0 || quantity <= 0 || quantity >= 1000) {  
2      return Result.fail("Invalid Args");  
3  }
```

这种代码在日常非常常见，但其最大的问题就是大量的非业务代码混杂在业务代码中，很明显的违背了单一职责原则。但因为当时入参仅仅是简单的int，所以这个逻辑只能出现在服务里。现在当入参改为了CQE之后，我们可以利用java标准JSR303或JSR380的Bean Validation来前置这个校验逻辑。

| 规范：CQE对象的校验应该前置，避免在ApplicationService里做参数的校验。可以通过JSR303/380和Spring Validation来实现

前面的例子可以改造为：

```
1  @Validated // Spring的注解  
2  public class CheckoutServiceImpl implements CheckoutService {  
3      OrderDTO checkout(@Valid CheckoutCommand cmd) { // 这里@Valid是JSR-303/380的注解  
4          // 如果校验失败会抛异常，在interface层被捕捉  
5      }  
6  }  
7  
8  @Data  
9  public class CheckoutCommand {  
10  
11      @NotNull(message = "用户未登陆")  
12      private Long userId;  
13  
14      @NotNull  
15      @Positive(message = "需要是合法的itemId")  
16      private Long itemId;  
17  
18      @NotNull  
19      @Min(value = 1, message = "最少1件")  
20      @Max(value = 1000, message = "最多不能超过1000件")  
21      private Integer quantity;  
22  }
```

这种做法的好处是，让ApplicationService更加清爽，同时各种错误信息可以通过Bean Validation的API做各种个性化定制。

- 避免复用CQE

因为CQE是有“意图”和“语意”的，我们需要尽量避免CQE对象的复用，哪怕所有的参数都一样，只要他们的语意不同，尽量还是要用不同的对象。

- 规范：针对于不同语意的指令，要避免CQE对象的复用

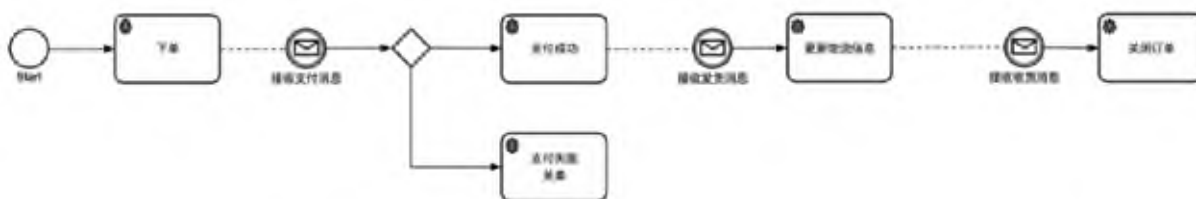
✘ 反例：一个常见的场景是“Create创建”和“Update更新”，一般来说这两种类型的对象唯一的区别是一个ID，创建没有ID，而更新则有。所以经常能看见有的同学用同一个对象来作为两个方法的入参，唯一区别是ID是否赋值。这个是错误的用法，因为这两个操作的语意完全不一样，他们的校验条件可能也完全不一样，所以不应该复用同一个对象。正确的做法是产出两个对象：

```
1  public interface CheckoutService {
2      OrderDTO checkout(@Valid CheckoutCommand cmd);
3      OrderDTO updateOrder(@Valid UpdateOrderCommand cmd);
4  }
5
6  @Data
7  public class UpdateOrderCommand {
8
9      @NotNull(message = "用户未登陆")
10     private Long userId;
11
12     @NotNull(message = "必须要有OrderID")
13     private Long orderId;
14
15     @NotNull
16     @Positive(message = "需要是合法的itemId")
17     private Long itemId;
18
19     @NotNull
20     @Min(value = 1, message = "最少1件")
21     @Max(value = 1000, message = "最多不能超过1000件")
22     private Integer quantity;
23
24 }
```

ApplicationService

ApplicationService负责业务流程的编排，是将原有业务流水账代码剥离了校验逻辑、领域计算、持久化等逻辑之后剩余的流程，是“胶水层”代码。

参考一个简易的交易流程：



在这个案例里可以看出来，交易这个领域一共有5个用例：下单、支付成功、支付失败关单、物流信息更新、关闭订单。这5个用例可以用5个Command/Event对象代替，也就是对应了5个方法。

我见过3种ApplicationService的组织形态：

1

一个ApplicationService类是一个完整的业务流程，其中每个方法负责处理一个Use Case。这种的好处是可以完整的收敛整个业务逻辑，从接口类即可对业务逻辑有一定的掌握，适合相对简单的业务流程。坏处就是对于复杂的业务流程会导致一个类的方法过多，有可能代码量过大。这种类型的具体案例如：

```
1 public interface CheckoutService {
2
3     // 下单
4     OrderDTO checkout(@Valid CheckoutCommand cmd);
5
6     // 支付成功
7     OrderDTO payReceived(@Valid PaymentReceivedEvent event);
8
9     // 支付取消
10    OrderDTO payCanceled(@Valid PaymentCanceledEvent event);
11
12    // 发货
13    OrderDTO packageSent(@Valid PackageSentEvent event);
14
15    // 收货
```

```
16 OrderDTO delivered(@Valid DeliveredEvent event);
17
18 // 批量查询
19 List<OrderDTO> query(OrderQuery query);
20
21 // 单个查询
22 OrderDTO getOrder(Long orderId);
23 }
```

2

针对于比较复杂的业务流程，可以通过增加独立的CommandHandler、EventHandler来降低一个类中的代码量：

```
1 @Component
2 public class CheckoutCommandHandler implements CommandHandler<CheckoutCommand, OrderDTO> {
3     @Override
4     public OrderDTO handle(CheckoutCommand cmd) {
5         //
6     }
7 }
8
9 public class CheckoutServiceImpl implements CheckoutService {
10     @Resource
11     private CheckoutCommandHandler checkoutCommandHandler;
12
13     @Override
14     public OrderDTO checkout(@Valid CheckoutCommand cmd) {
15         return checkoutCommandHandler.handle(cmd);
16     }
17 }
```

3

比较激进一点，通过CommandBus、EventBus，直接将指令或事件抛给对应的Handler，EventBus比较常见。具体案例代码如下，通过消息队列收到MQ消息后，生成Event，然后由EventBus做路由到对应的Handler：

```
1 // Application层
2 // 在这里框架通常可以根据接口识别到这个负责处理PaymentReceivedEvent
3 // 也可以通过增加注解识别
4 @Component
5 public class PaymentReceivedHandler implements EventHandler<PaymentReceivedEvent> {
6     @Override
7     public void process(PaymentReceivedEvent event) {
8         //
9     }
10 }
11
12 // Interface层, 这个是RocketMQ的Listener
13 public class OrderMessageListener implements MessageListenerOrderly {
14
15     @Resource
16     private EventBus eventBus;
17
18     @Override
19     public ConsumeOrderlyStatus consumeMessage(List<MessageExt> msgs, ConsumeOrderlyContext context) {
20
21         PaymentReceivedEvent event = new PaymentReceivedEvent();
22         eventBus.dispatch(event); // 不需要指定消费者
23
24         return ConsumeOrderlyStatus.SUCCESS;
25     }
26 }
```

⚠ 不建议

这种做法可以实现Interface层和某个具体的ApplicationService或Handler的完全静态解耦，在运行时动态dispatch，做的比较好的框架如AxonFramework。虽然看起来很便利，但是根据我们自己业务的实践和踩坑发现，当代码中的CQE对象越来越多，handler越来越复杂时，运行时的dispatch缺乏了静态代码间的关联关系，导致代码很难读懂，特别是当你需要trace一个复杂调用链路时，因为dispatch是运行时的，很难摸清楚具体调用到的对象。所以我们虽然曾经有过这种尝试，但现在已经不建议这么做了。

- Application Service 是业务流程的封装，不处理业务逻辑

虽然之前曾经无数次重复ApplicationService只负责业务流程串联，不负责业务逻辑，但如何判断一段代码到底是业务流程还是逻辑呢？

举个之前的例子，最初的代码重构后：

```
1  @Service
2  @Validated
3  public class CheckoutServiceImpl implements CheckoutService {
4
5      private final OrderDtoAssembler orderDtoAssembler = OrderDtoAssembler.INSTANCE;
6      @Resource
7      private ItemService itemService;
8      @Resource
9      private InventoryService inventoryService;
10     @Resource
11     private OrderRepository orderRepository;
12
13     @Override
14     public OrderDTO checkout(@Valid CheckoutCommand cmd) {
15         ItemDO item = itemService.getItem(cmd.getItemId());
16         if (item == null) {
17             throw new IllegalArgumentException("Item not found");
18         }
19
20         boolean withholdSuccess = inventoryService.withhold(cmd.getItemId(), cmd.getQuantity());
21         if (!withholdSuccess) {
22             throw new IllegalArgumentException("Inventory not enough");
23         }
24
25         Order order = new Order();
26         order.setBuyerId(cmd.getUserId());
27         order.setSellerId(item.getSellerId());
28         order.setItemId(item.getItemId());
29         order.setItemTitle(item.getTitle());
30         order.setItemUnitPrice(item.getPriceInCents());
31         order.setCount(cmd.getQuantity());
32
33         Order savedOrder = orderRepository.save(order);
34
35         return orderDtoAssembler.orderToDTO(savedOrder);
36     }
37 }
```

- 判断是否业务流程的几个点：

1

不要有if/else分支逻辑：也就是说代码的Cyclomatic Complexity（循环复杂度）应该尽量等于1

通常有分支逻辑的，都代表一些业务判断，应该将逻辑封装到DomainService或者Entity里。但这不代表完全不能有if逻辑，比如，在这段代码里：

```
1 boolean withholdSuccess = inventoryService.withhold(cmd.getItemId(), cmd.getQuantity());
2 if (!withholdSuccess) {
3     throw new IllegalArgumentException("Inventory not enough");
4 }
```

虽然CC > 1，但是仅仅代表了中断条件，具体的业务逻辑处理并没有受影响。可以把它看作为Precondition。

2

不要有任何计算：

在最早的代码里有这个计算：

```
1 // 5) 领域计算
2 Long cost = item.getPriceInCents() * quantity;
3 order.setTotalCost(cost);
```

通过将这个计算逻辑封装到实体里，避免在ApplicationService里做计算

```
1 @Data
2 public class Order {
3
4     private Long itemUnitPrice;
5     private Integer count;
6
7     // 把原来一个在ApplicationService的计算迁移到Entity里
8     public Long getTotalCost() {
9         return itemUnitPrice * count;
10    }
11 }
```

```
12  
13 order.setItemUnitPrice(item.getPriceInCents());  
14 order.setCount(cmd.getQuantity());
```

3

一些数据的转化可以交给其他对象来做：

比如DTO Assembler，将对象间转化的逻辑沉淀在单独的类中，降低ApplicationService的复杂度

```
1 OrderDTO dto = orderDtoAssembler.orderToDTO(savedOrder);
```

• 常用的ApplicationService “套路”

我们可以看出来，ApplicationService的代码通常有类似的结构：AppService通常不做任何决策（Precondition除外），仅仅是把所有决策交给DomainService或Entity，把跟外部交互的交给Infrastructure接口，如Repository或防腐层。

一般的“套路”如下：

- 准备数据：包括从外部服务或持久化源取出相对应的Entity、VO以及外部服务返回的DTO。
- 执行操作：包括新对象的创建、赋值，以及调用领域对象的方法对其进行操作。需要注意的是这个时候通常都是纯内存操作，非持久化。
- 持久化：将操作结果持久化，或操作外部系统产生相应的影响，包括发消息等异步操作。

如果涉及到对多个外部系统（包括自身的DB）都有变更的情况，这个时候通常处在“分布式事务”的场景里，无论是用分布式TX、TCC、还是Saga模式，取决于具体场景的设计，在此处暂时略过。

DTO Assembler

一个经常被忽视的问题是 ApplicationService应该返回 Entity 还是 DTO？这里提出一个规范，在DDD分层架构中：

| ApplicationService应该永远返回DTO而不是Entity

为什么呢？

1.构建领域边界：ApplicationService的入参是CQE对象，出参是DTO，这些基本上都属于简单的POJO，来确保Application层的内外互相不影响。

2.降低规则依赖：Entity里面通常会包含业务规则，如果ApplicationService返回Entity，则会导致调用方直接依赖业务规则。如果内部规则变更可能直接影响到外部。

3.通过DTO组合降低成本：Entity是有限的，DTO可以是多个Entity、VO的自由组合，一次性封装成复杂DTO，或者有选择的抽取部分参数封装成DTO可以降低对外的成本。

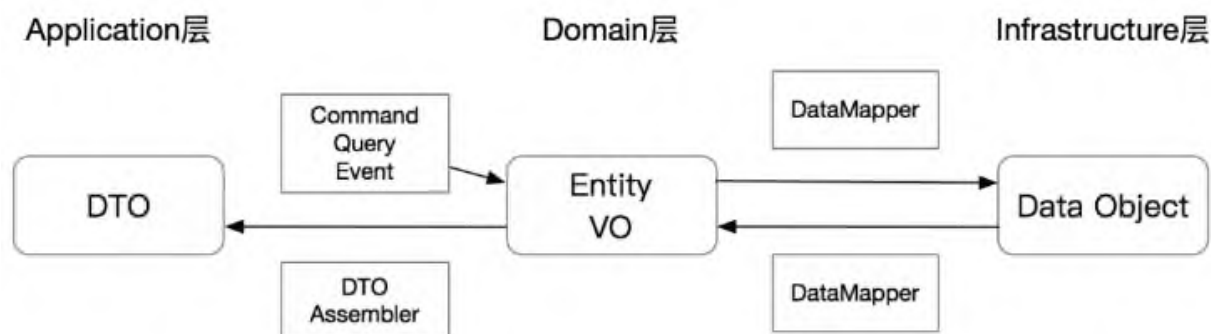
因为我们操作的对象是Entity，但是输出的对象是DTO，这里就需要一个专属类型的对象叫DTO Assembler。DTO Assembler的唯一职责是将一个或多个Entity/VO，转化为DTO。注意：DTO Assembler通常不建议有反操作，也就是不会从DTO到Entity，因为通常一个DTO转化为Entity时是无法保证Entity的准确性的。

通常，Entity转DTO是有成本的，无论是代码量还是运行时的操作。手写转换代码容易出错，为了节省代码量用Reflection会造成极大的性能损耗。所以这里我还是不遗余力的推荐MapStruct这个库。MapStruct通过静态编译时代码生成，通过写接口和配置注解就可以生成对应的代码，且因为生成的代码是直接赋值，其性能损耗基本可以忽略不计。

通过MapStruct，代码即可简化为：

```
1  import org.mapstruct.Mapper;
2  @Mapper
3  public interface OrderDtoAssembler {
4      OrderDtoAssembler INSTANCE = Mappers.getMapper(OrderDtoAssembler.class);
5      OrderDTO orderToDTO(Order order);
6  }
7
8  public class CheckoutServiceImpl implements CheckoutService {
9      private final OrderDtoAssembler orderDtoAssembler = OrderDtoAssembler.INSTANCE;
10
11      @Override
12      public OrderDTO checkout(@Valid CheckoutCommand cmd) {
13          // ...
14          Order order = new Order();
15          // ...
16          Order savedOrder = orderRepository.save(order);
17          return orderDtoAssembler.orderToDTO(savedOrder);
18      }
19  }
```


结合之前的Data Mapper，DTO、Entity和DataObject之间的关系如下图：



Result vs Exception

最后，上文曾经提及在Interface层应该返回Result，在Application层应该返回DTO，在这里再次重复提出规范：

Application层只返回DTO，可以直接抛异常，不用统一处理。所有调用到的服务也都可以直接抛异常，除非需要特殊处理，否则不需要刻意捕捉异常

异常的好处是能明确的知道错误的来源，堆栈等，在Interface层统一捕捉异常是为了避免异常堆栈信息泄漏到API之外，但是在Application层，异常机制仍然是信息量最大，代码结构最清晰的方法，避免了Result的一些常见且繁杂的Result.isSuccess判断。所以在Application层、Domain层，以及Infrastructure层，遇到错误直接抛异常是最合理的方法。

Anti-Corruption Layer防腐层

本文仅仅简单描述一下ACL的原理和作用，具体的实施规范可能要等到另外一篇文章。

在ApplicationService中，经常会依赖外部服务，从代码层面对外部系统产生了依赖。比如上文中的：

```
1 ItemDO item = itemService.getItem(cmd.getItemId());
2 boolean withholdSuccess = inventoryService.withhold(cmd.getItemId(), cmd.getQuantity());
```

会发现我们的ApplicationService会强依赖ItemService、InventoryService以及ItemDO这个对象。如果任何一个服务的方法变更，或者ItemDO字段变更，都会有可能影响到ApplicationService的代码。也就是说，我们自己的代码会因为强依赖了外部系统的变化而变更，这个在复杂系统中应该是尽量避免的。那么如何做到对外部系统的隔离呢？需要加入ACL防腐层。

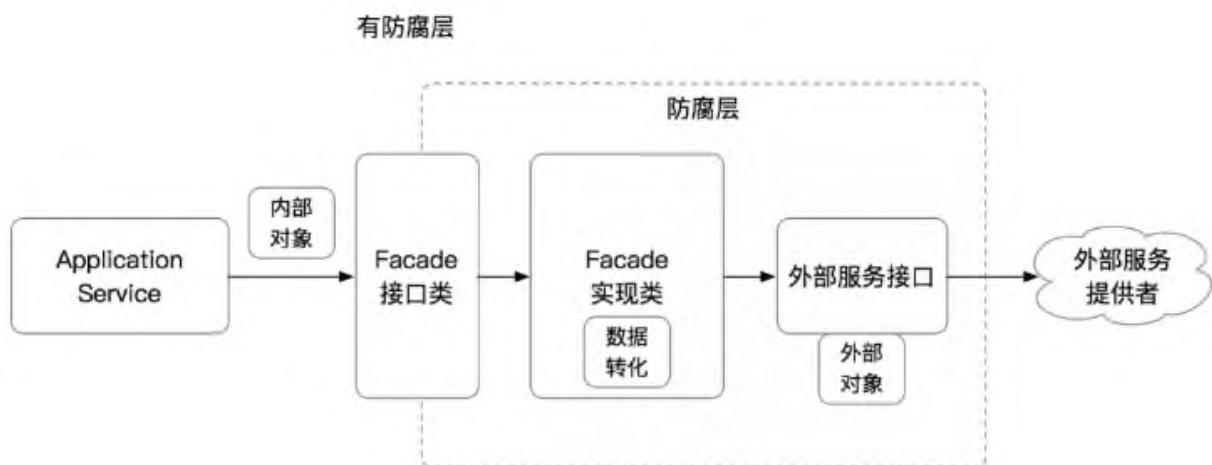
ACL防腐层的简单原理如下：

对于依赖的外部对象，我们抽取所需要的字段，生成一个内部所需的VO或DTO类
构建一个新的Facade，在Facade中封装调用链路，将外部类转化为内部类
针对外部系统调用，同样的用Facade方法封装外部调用链路

无防腐层的情况：



有防腐层的情况：



具体简单实现，假设所有外部依赖都命名为ExternalXXXService：

```
1 // 自定义的内部值类
2 @Data
3 public class ItemDTO {
4     private Long itemId;
5     private Long sellerId;
6     private String title;
7     private Long priceInCents;
8 }
9
10 // 商品Facade接口
11 public interface ItemFacade {
12     ItemDTO getItem(Long itemId);
13 }
14 // 商品facade实现
15 @Service
16 public class ItemFacadeImpl implements ItemFacade {
17
18     @Resource
19     private ExternalItemService externalItemService;
20
21     @Override
22     public ItemDTO getItem(Long itemId) {
23         ItemDO itemDO = externalItemService.getItem(itemId);
24         if (itemDO != null) {
25             ItemDTO dto = new ItemDTO();
26             dto.setItemId(itemDO.getItemId());
27             dto.setTitle(itemDO.getTitle());
28             dto.setPriceInCents(itemDO.getPriceInCents());
29             dto.setSellerId(itemDO.getSellerId());
30             return dto;
31         }
32         return null;
33     }
34 }
35
36 // 库存Facade
37 public interface InventoryFacade {
38     boolean withhold(Long itemId, Integer quantity);
39 }
40 @Service
41 public class InventoryFacadeImpl implements InventoryFacade {
42
43     @Resource
```

```
44 private ExternalInventoryService externalInventoryService;
45
46 @Override
47 public boolean withhold(Long itemId, Integer quantity) {
48     return externalInventoryService.withhold(itemId, quantity);
49 }
50 }
```

通过ACL改造之后，我们ApplicationService的代码改为：

```
1  @Service
2  public class CheckoutServiceImpl implements CheckoutService {
3
4      @Resource
5      private ItemFacade itemFacade;
6      @Resource
7      private InventoryFacade inventoryFacade;
8
9      @Override
10     public OrderDTO checkout(@Valid CheckoutCommand cmd) {
11         ItemDTO item = itemFacade.getItem(cmd.getItemId());
12         if (item == null) {
13             throw new IllegalArgumentException("Item not found");
14         }
15
16         boolean withholdSuccess = inventoryFacade.withhold(cmd.getItemId(), cmd.getQuantity());
17         if (!withholdSuccess) {
18             throw new IllegalArgumentException("Inventory not enough");
19         }
20
21         // ...
22     }
23 }
```

很显然，这么做的好处是ApplicationService的代码已经完全不再直接依赖外部的类和方法，而是依赖了我们自己内部定义的值类和接口。如果未来外部服务有任何的变更，需要修改的是Facade类和数据转化逻辑，而不需要修改ApplicationService的逻辑。

Repository可以认为是一种特殊的ACL，屏蔽了具体数据操作的细节，即使底层数据库结构变更，数据库类型变更，或者加入其他的持久化方式，Repository的接口保持稳定，ApplicationService就能保持不变。

在一些理论框架里ACL Facade也被叫做Gateway，含义是一样的。

Orchestration vs Choreography

在本文最后想聊一下复杂业务流程的设计规范。在复杂的业务流程里，我们通常面临两种模式：Orchestration 和 Choreography。很无奈，这两个英文单词的百度翻译/谷歌翻译，都是“编排”，但实际上这两种模式是完全不一样的设计模式。

Orchestration的编排（比如SOA/微服务的服务编排Service Orchestration）是我们通常熟悉的用法，Choreography是最近出现了事件驱动架构EDA才慢慢流行起来。网上可能会有其他的翻译，比如编制、编舞、协作等，但感觉都没有真正的把英文单词的意思表达出来，所以为了避免误解，在下文我尽量还是用英文原词。如果谁有更好的翻译方法欢迎联系我。

模式简介

Orchestration：通常出现在脑海里的是一个交响乐团（Orchestra，注意这两个词的相似性），如下图。交响乐团的核心是一个唯一的指挥家Conductor，在一个交响乐中，所有的音乐家必须听从Conductor的指挥做操作，不可以独自发挥。所以在Orchestration模式中，所有的流程都是由一个节点或服务触发的。我们常见的业务流程代码，包括调用外部服务，就是Orchestration，由我们的服务统一触发。



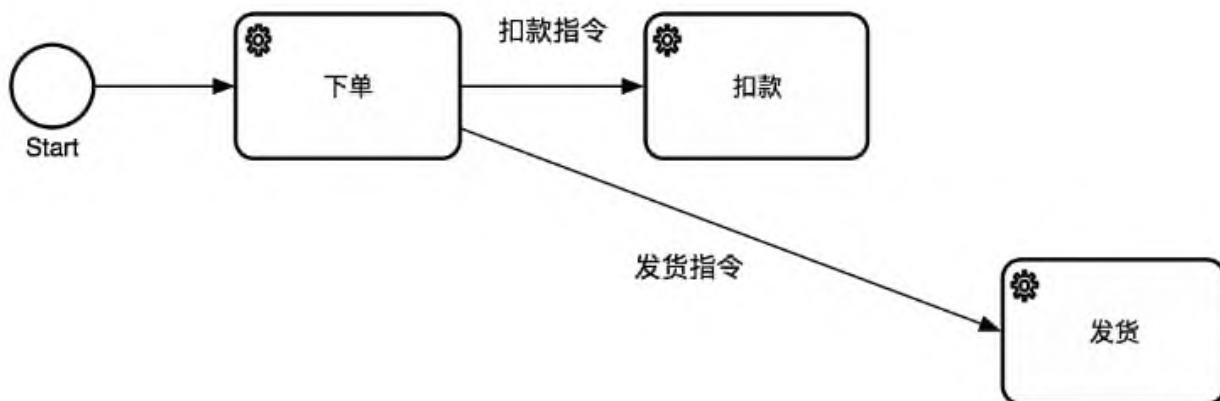
Choreography：通常会出现脑海的场景是一个舞剧（来自于希腊文的舞蹈，Choros），如下图。其中每个不同的舞蹈家都在做自己的事，但是没有一个中心化的指挥。通过协作配合，每个人做好自己的事，整个舞蹈可以展现出一个完整的、和谐的画面。所以在Choreography模式中，每个服务都是独立的个体，可能会响应外部的一些事件，但整个系统是一个整体。



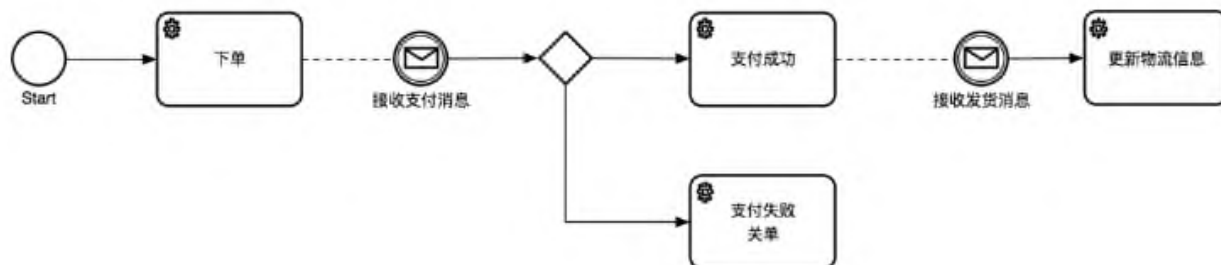
案例

用一个常见的例子：下单后支付并发货

如果这个案例是Orchestration，则业务逻辑为：下单时从一个预存的账户里扣取资金，并且生成物流单发货，从图上看是这样的：



如果这个案例是Choreography，则业务逻辑为：下单，然后等支付成功事件，然后再发货，类似这样：



模式的区别和选择

虽然看起来这两种模式都能达到一样的业务目的，但是在实际开发中他们有巨大的差异：

- 从代码依赖关系来看：

Orchestration：涉及到一个服务调用到另外的服务，对于调用方来说，是强依赖的服务提供方。

Choreography：每一个服务只是做好自己的事，然后通过事件触发其他的服务，服务之间没有直接调用上的依赖。但要注意的是下游还是会依赖上游的代码（比如事件类），所以可以认为是下游对上游有依赖。

- 从代码灵活性来看：

Orchestration：因为服务间的依赖关系是写死的，增加新的业务流程必然需要修改代码。

Choreography：因为服务间没有直接调用关系，可以增加或替换服务，而不需要改上游代码。

- 从调用链路来看：

Orchestration：是从一个服务主动调用另一个服务，所以是Command-Driven指令驱动的。

Choreography：是每个服务被动的被外部事件触发，所以是Event-Driven事件驱动的。

- 从业务职责来看：

Orchestration：有主动的调用方（比如：下单服务）。无论下游的依赖是谁，主动的调用方都需要为整个业务流程和结果负责。

Choreography：没有主动调用方，每个服务只关心自己的触发条件和结果，没有任何一个服务会为整个业务链路负责

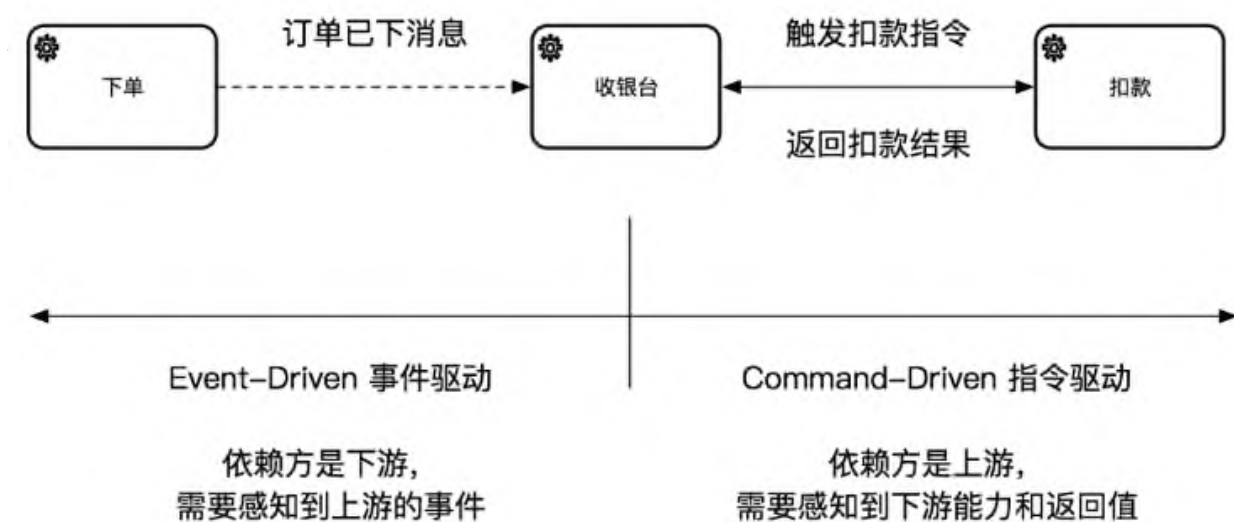
• 小结:

	Orchestration	Choreography
驱动力	指令驱动Command-Driven	事件驱动Event-Driven
调用依赖	上游强依赖下游	无直接调用依赖 但是有代码依赖 可以认为是下游依赖上游
灵活性	较差	较高
业务职责	上游为业务负责	无全局责任人

另外需要重点明确的：“指令驱动”和“事件驱动”的区别不是“同步”和“异步”。指令可以是同步调用，也可以是异步消息触发（但异步指令不是事件）；反过来事件可以是异步消息，但也完全可以是进程内的同步调用。所以指令驱动和事件驱动差异的本质不在于调用方式，而是一件事情是否“已经”发生。

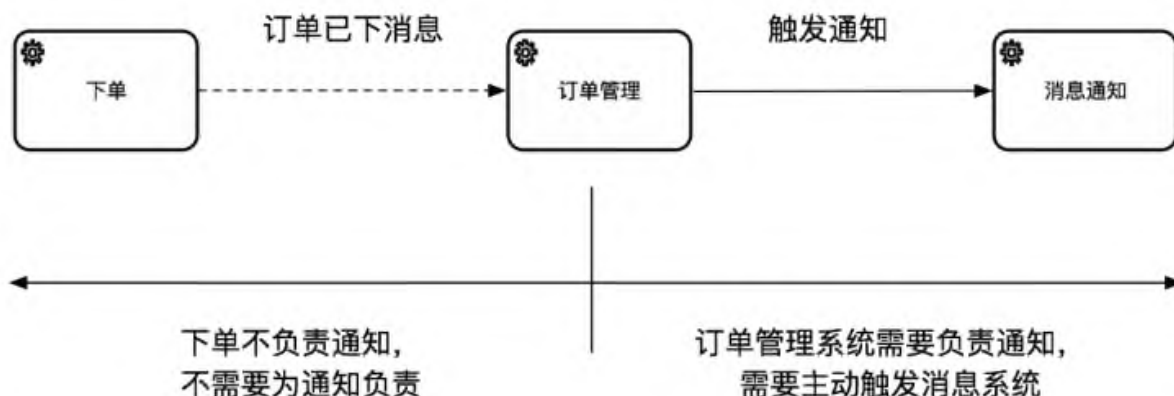
• 所以在日常业务中当你碰到一个需求时，该如何选择是用Orchestration还是Choreography?

这里给出两个判断方法：



在代码中的依赖是比较明确的：如果你是下游，上游对你无感知，则只能走事件驱动；如果上游必须要对你有感知，则可以走指令驱动。反过来，如果你是上游，需要对下游强依赖，则是指令驱动；如果下游是谁无所谓，则可以走事件驱动。

2. 找出业务中的“负责人”：



第二种方法是根据业务场景找出其中的“负责人”。比如，如果业务需要通知卖家，下单系统的单一职责不应该为消息通知负责，但订单管理系统需要根据订单状态的推进主动触发消息，所以是这个功能的负责人。

在一个复杂业务流程里，通常两个模式都要有，但也很容易设计错误。如果出现依赖关系很奇怪，或者代码里调用链路/负责人梳理不清楚的情况，可以尝试转换一下模式，可能会好很多。

- 哪个模式更好？

很显然，没有最好的模式，只有最合适自己业务场景的模式。

✘ 反例：最近几年比较流行的Event-Driven Architecture（EDA）事件驱动架构，以及Reactive-Programming响应式编程（比如RxJava），虽然有很多创新，但在一定程度上是“当你有把锤子，所有问题都是钉子”的典型案列。他们对一些基于事件的、流处理的问题有奇效，但如果拿这些框架硬套指令驱动的业务，就会感到代码极其“不协调”，认知成本提高。所以在日常选型中，还是要先根据业务场景梳理出来是哪些流程中的部分是Orchestration，哪些是Choreography，然后再选择相对应的框架。

跟DDD分层架构的关系

最后，讲了这么多O vs C，跟DDD有啥关系？很简单：

O&C其实是Interface层的关注点，Orchestration = 对外的API，而Choreography = 消息或事件。当你决策了O还是C之后，需要在interface层承接这些“驱动力”。

无论O&C如何设计，Application层都“无感知”，因为ApplicationService天生就可以处理Command、Query和Event，至于这些对象怎么来，是Interface层的决策。

所以，虽然Orchestration 和 Choreography是两种完全不同的业务设计模式，但最终落到Application层的代码应该是一致的，这也是为什么Application层是“用例”而不是“接口”，是相对稳定的存在。

总结

只要是做业务的，一定会需要写业务流程和服务编排，但不代表这种代码一定质量差。通过DDD的分层架构里的Interface层和Application层的合理拆分，代码可以变得优雅、灵活，能更快的响应业务但同时又能更好的沉淀。本文主要介绍了一些代码的设计规范，帮助大家掌握一定的技巧。

Interface层：

- 职责：主要负责承接网络协议的转化、Session管理等
- 接口数量：避免所谓的统一API，不必人为限制接口类的数量，每个/每类业务对应一套接口即可，接口参数应该符合业务需求，避免大而全的入参
- 接口出参：统一返回Result
- 异常处理：应该捕捉所有异常，避免异常信息的泄漏。可以通过AOP统一处理，避免代码里有大量重复代码。

Application层：

- 入参：具像化Command、Query、Event对象作为ApplicationService的入参，唯一可以的例外是单ID查询的场景。
- CQE的语意化：CQE对象有语意，不同用例之间语意不同，即使参数一样也要避免复用。
- 入参校验：基础校验通过Bean Validation api解决。Spring Validation自带Validation的AOP，也可以自己写AOP。
- 出参：统一返回DTO，而不是Entity或DO。
- DTO转化：用DTO Assembler负责Entity/VO到DTO的转化。
- 异常处理：不统一捕捉异常，可以随意抛异常。

部分Infra层：

- 用ACL防腐层将外部依赖转化为内部代码，隔离外部的影响

业务流程设计模式：

- 没有最好的模式，取决于业务场景、依赖关系、以及是否有业务“负责人”。避免拿着锤子找钉子。

前瞻预告

- CQRS是Application层的一种设计模式，是基于Command和Query分离的一种设计理念，从最简单的对象分离，到目前最复杂的Event-Sourcing。这个topic有很多需要深入的点，也经常可以被用到，特别是结合复杂的Aggregate。后面单独会拉出来讲，标题暂定为《CQRS的7层境界》
- 在当今复杂的微服务开发环境下，依赖外部团队开发的服务是不可避免的，但强耦合带来的成本（无论是变更、代码依赖、甚至Maven Jar包间接依赖）是一个复杂系统长期不可忽视的点。ACL防腐层是一种隔离理念，将外部耦合去除，让内部代码更加纯粹。ACL防腐层可以有很多种，Repository是一种特殊的面相数据持久化的ACL，K8S-sidecar-istio 可以说是一种网络层的ACL，但在Java/Spring里可以有比Istio更高效、更通用的方法，待后文介绍。
- 当你开始用起来DDD时，会发现很多代码模式都非常类似，比如主子订单就是总分模式、类目体系的CPV模式也可以用到一些活动上，ECS模式可以在互动业务上发挥作用等等。后面会尝试总结出一些通用的领域设计模式，他们的设计思路、可以解决的问题类型、以及实践落地的方法。

● 浅谈系统实现层面稳定性保障

- 作者：高丙寅
- 出品：淘系技术

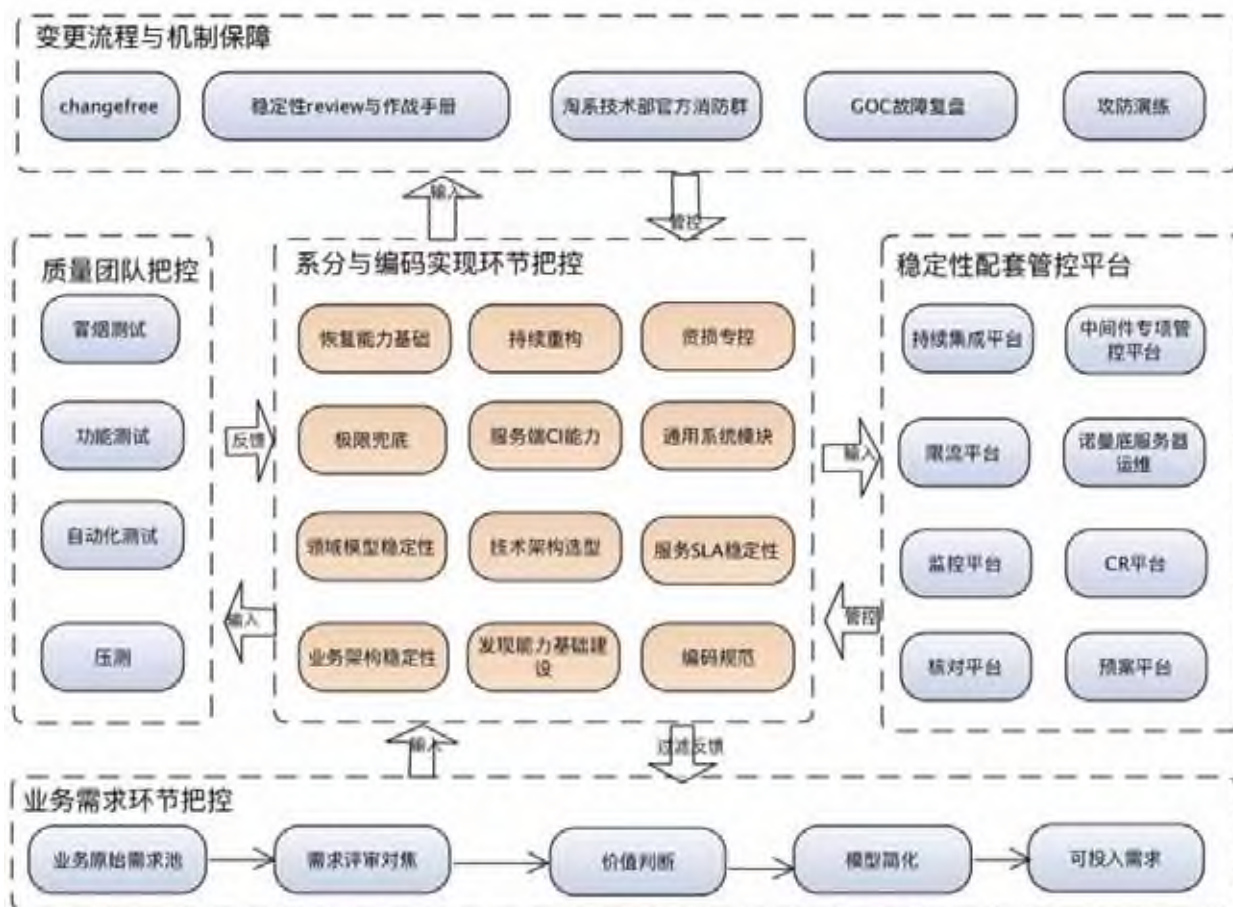
导读

上海外滩建筑群包括古典主义风格的亚细亚大楼（1915年），英国古典式的上海总会大楼（1911年），欧洲古典折中主义的海关大楼（1925年），仿意大利文艺复兴风格的汇中饭店大楼（1906年）等，这些建筑历经百年风雨，仍巍然屹立固若堡垒，保持其原本风貌，让今人得以深切领略一个世纪以前的壮丽繁华。这一切，除得益于人为保护之外，最主要原因是建筑自身主体结构具备较高的稳定可靠性。

而相比传统建筑工程，软件工程有两个显著特点，一是具备规模化快速复制扩散的能力，二是在竣工之后依然可以被改造并保持高速进化。这也就导致了软件工程一点点的不足也可能被快速扩散、无限放大造成大规模损失，进化中既有的稳定性结构可能会被不断打破，可能导致大量的救火投入，疲于奔命，极大地消耗有效而宝贵研发资源，阻碍软件的进行甚至带来灾难性后果。所以对于任何一家上规模的软件开发的企业来说，稳定性保障都是必须面对和解决好的课题。我个人加入阿里之初是在国际支付宝核心团队长期负责金融系统稳定性保障，其后扎根淘系技术三年有余，参与了多种不同类型系统设计与稳定性建设，以及大促稳定性保障工作。对比总结下来无论电商、金融、物流、ERP型软件工程，其稳定性保障策略是有较多共性经验的。本文主要结合金融、电商两种场景下的个人以及所在团队实践，谈谈稳定性保障的一些思路方法。

稳定性保障围绕的核心

稳定性保障涉及机房、网络、硬件部署到业务场景、交互设计，再到应用架构代码质量、流量与封网管控、攻防复盘等，是一项非常系统化的工程。而分工协作使得上述大部分工作流程化标准化，比如硬件问题、网络问题、Jvm监控等均可以由专业化团队提供统一配套方案，做为业务开发主要是利用这些工具更快的发现问题，干预度相对有限。但具体业务系统实现距离拖拽式批量生产尚需时日，最终产出实现方案差异可能很大，所以所有稳定性工作的核心还是围绕系统与代码本身来开展，如下图所示，无论是质量团队、各方管控平台，还是研发流程保障机制，最终的稳定性还是体现在系统代码上。因此，对于业务开发的来说，稳定性保障更应该着眼于系统设计实现环节的控制，如下橙色部分：



而正是需求不受控、架构局限、代码扩展性不足、监控日志不统一、下游依赖不统一之类问题导致了每到大促前夕的紧张慌乱，重复的梳理、反复的打补丁，需要额外的进行压力测试，预案配置演练，消耗巨大且效果有限。所以我们需要贯穿于系统实现各环节，来制定合理的稳定性策略。系统实现核心环节与稳定性关系如下图所示：



接下来着重针对上图所示各环节，做一下总结。

业务大环境与业务架构稳定性

一个人的命运必须要考虑历史进程影响，谈系统稳定性则不可以忽略大的业务环境，业务环境是指所在的部门业务现状，以及自己所在业务部门与其它业务部门的合作现状，可能发生的业务调整重组。

有时候为了满足业务上的发展目标，老板们不得不频繁的做业务调整。随之而来的是技术团队与系统架构的拥抱变化，系统拆分、合并、重建一些列工作，必然会极大地打乱既有的稳定性保障策略，所以必须要从业务环境业务架构的层面来审视并调整以跟上变化。

比如天猫技术部和淘宝技术部合并、拆分，洋淘业务下线，逛逛品牌上线，躺平业务独立，导致原有支持的统一社区运营工作台不再work，各自垂直业务线因发展需要或者被迫不得不独立出自己的运营后台管理，自己的索引，迁移兼容老数据，影响不可谓不深远。

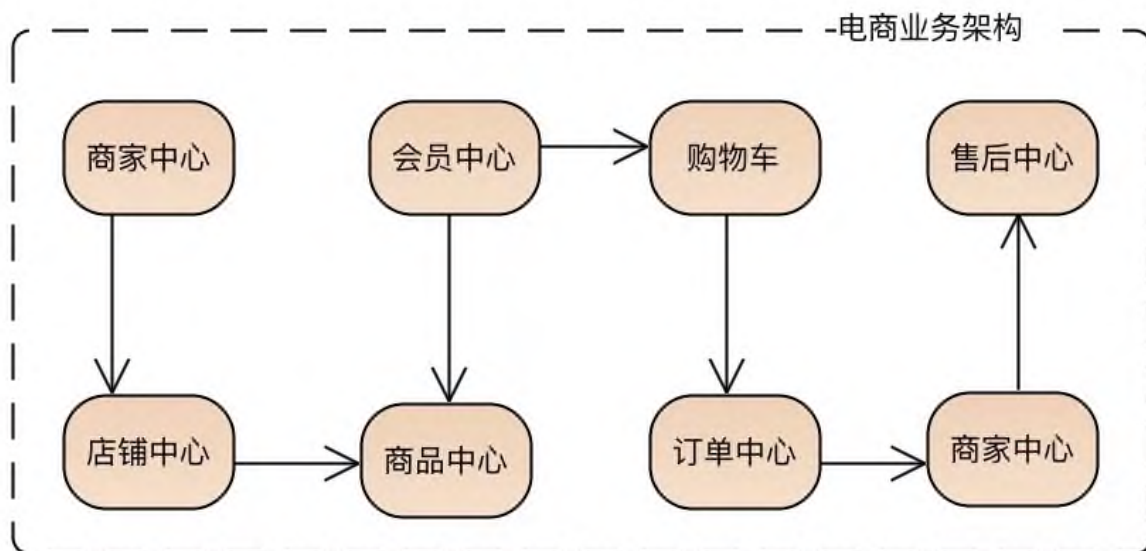
再比如淘宝社交账号体系原来是作为淘宝内部的一个体系，仅仅面向淘宝会员服务，但随着闲鱼、躺平的独立，需要延伸出新的特性化需求，那么以现有的形态继续支持跨BU的业务发展是否是最优的方案。如果继续支持面临人力、机器资源的保障的投入，那这些与当前BU的价值与方向是否吻合？这些是我们业务上去要考虑的问题。

平台化产品思维下，所有人都想建平台并急于让更多的人接入进来，以发挥自己平台影响力，体现自身价值。而一旦业务目标调整，团队方向跟着调整，平台这一块儿不再是重点，角色反转，之前苦苦拉过来的客户被告知不再提供服务，甚至限期迁移，这对自己、对客户方的系统稳定性都极其不利。所以，越往上层的团队，建平台越审慎，先做好自己的主业而不是盲目扩展边界。一个基于短期目的，或者缺乏长远规划的所谓平台直接提供给别人用，也是不负责任的表现。

而对于一个新的业务来说，我们也需要要求产品运营不仅仅是提出一个商业需求商业模式，而是要系统化思考一个业务的心智及演变趋势。从天马行空的想法到一个可行性方案，技术人员需要发挥自身优势，和产品运营同学一起思考，将商业模式进行细化，想清楚，想透彻，往往业务上一点点的权衡或者调整对技术上的改变将会带来四两拨千斤的效果。

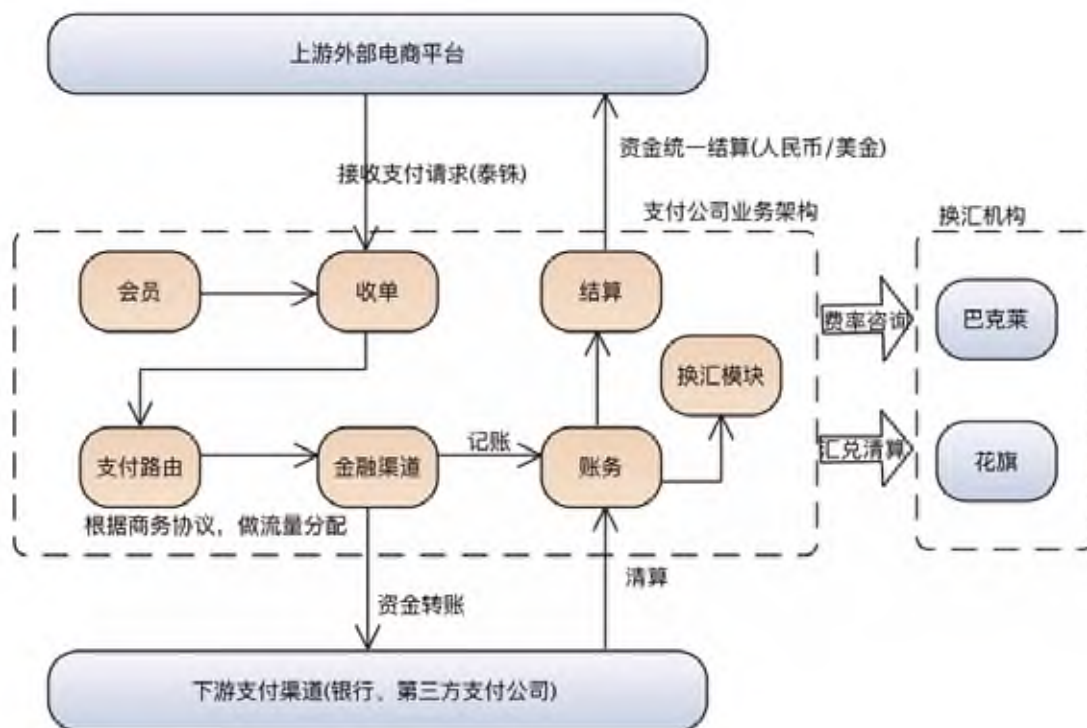
系统是业务的直观反馈，业务架构很大程度上决定了进一步的技术架构。如果产研团队对业务理解不深刻不透彻，只着眼于未来一年甚至几个月短期需求与利益，想到哪里做到哪里，那么技术层面就无法做好提前的布局与设计，堆砌、重复就会随之而来。大的环境背景不稳则限制了天花板，后续实现过程上无论怎么努力只能治标难以治本！所以对当前身处的业务的环境整体大图轮廓的认知，是建立全局稳定性思考的前提。

电商系统业务架构示意



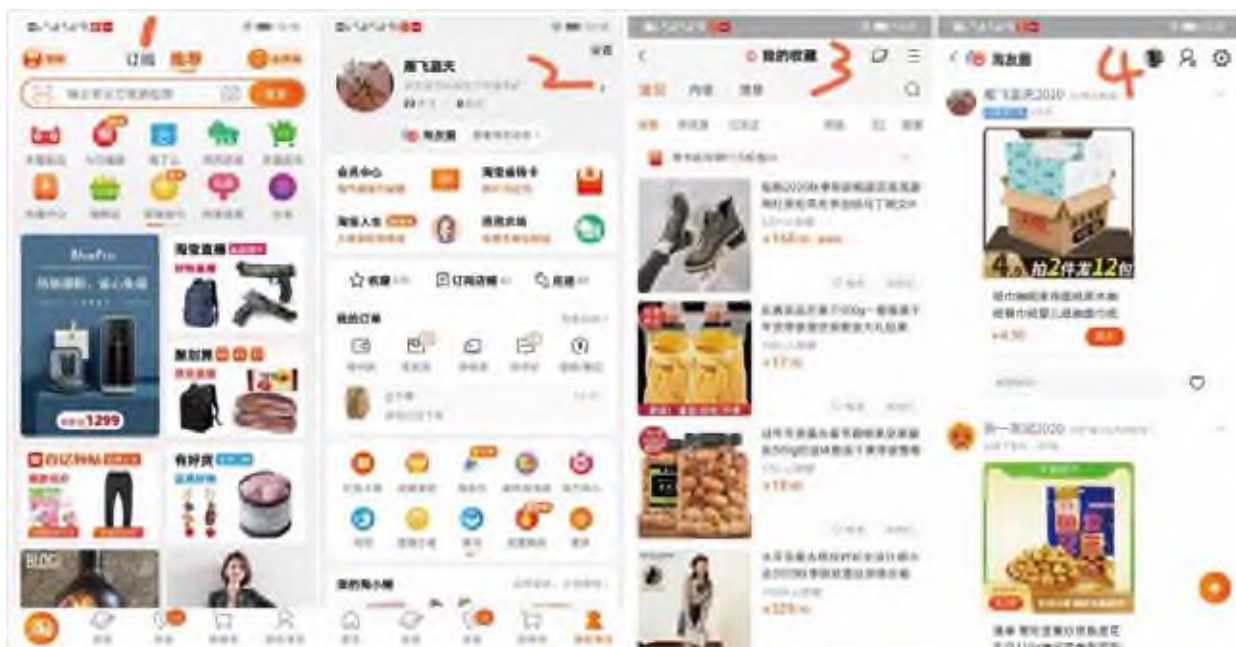
如果想做一个最小型的电商网站，至少也需要包含这几个业务模块，会员、商户库存、下单模块、店铺管理中心，规模稍大之后每个模块可能要支持独立的运营，可能还要考虑支持商业化接入，还可能延伸出物流、售后、资金管理等延伸模块，那么这些模块将怎么去协作，这些东西我们需要在业务大图中有一定预见。

电商系统业务架构示意



比如泰国用户小A在登录速卖通(www.aliexpress.com)后,使用支付宝支付泰铢,购买了一双鞋子。那么这个支付过程至少就会涉及到上面几方。在这个过程中,速卖通是做为一个外部商户之一,支付宝要服务好千万个这样的商户。就需要独立的收单对接团队,独立出收单模块。在用户支付的过程中,还可能根据与银行之间支付服务费差异,营销活动等决策要给用户推荐的支付方式,那么就需要一个运营可干预的支付路由模块,而下游同样提供泰铢支付的泰国支付机构也有很多家,我们对于每一家的接入都是要有商务协议和技术准入,那么把金融渠道看做一个业务单元,既然有资金流动那么独立的财务核算模块就是必不可少的,包括围绕这个支付活动可以沉淀下来的会员体系,决定了在支付平台中要有用户模块。

导购场页面架构示意



类似于手机淘宝中首页、我的淘宝、收藏夹这类通用的强入口,这种兵家必争之地更需要技术、产品、运营一起考虑清楚业务架构。尽可能提供通用的UI标准与接入方案,前瞻性考虑页面业务架构,能保证我们的页面复杂度不随业务方呈现线性增强趋势,业务接入可以批量化、可配置化。而如果局限于应付单次的特性化需求堆砌,几经迭代,业务逻辑越来越臃肿,每次调整将牵一发而动全身,最终难以为继。产品拥抱变化,加之人员调整频繁,新一批的产品、技术没有人能讲清楚这个页面的真实运行情况。

业务架构并不只是部门老板的运筹决策,也不是必须面对特别大的场景才需要考虑。即使我们只是一个小页面甚至小区块的产品、开发,那么对这个页面这个组件也可以有很好的业务规划,比如如下区块虽小,但做为一个产品的流量入口来说却具有决定性意义。



和产品设计达成一致后可以抽象出稳定模型如下：



可分为：logo区、数字标、小红点、头像区，引导文案区、图片氛围区、色彩氛围区。

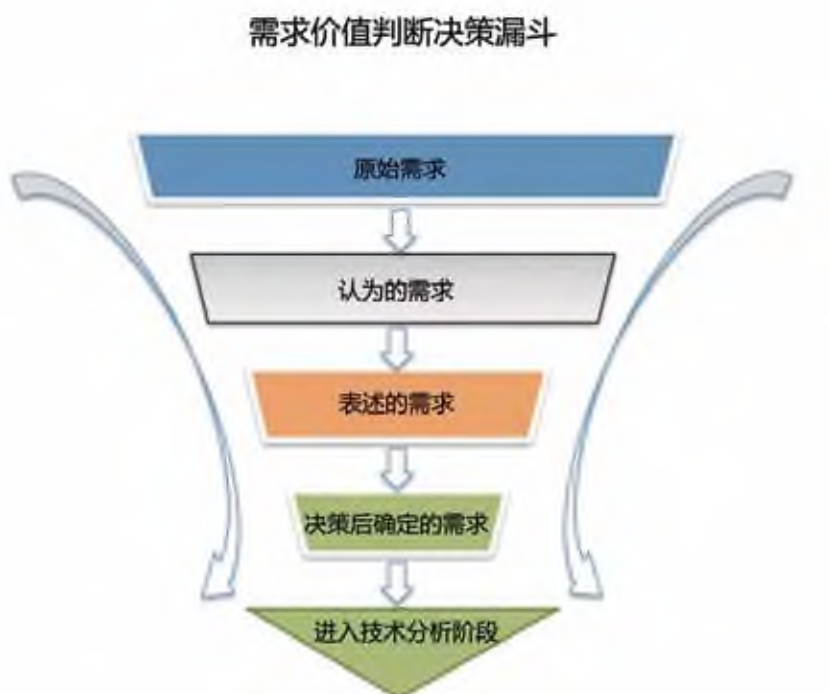
基于这个稳定的区块交互架构，我们定义统一标准协议，之后服务端就可以前瞻性规划，考虑各类营销场景、AB方案，大做文章以支持多样化业务运营需求且不会因为频繁的前后端联动修改引入稳定性问题。

先要清晰认识了解所在部门大的业务环境、背景、未来趋势。确保我们和业务方能建立业务架构上的共识，目标一致。以此为基石，保证方向正确的前提下，贴合实际去考虑系统架构与边界问题，做配套的稳定性方案，才具有更好的可行性！

业务需求把控与稳定性关系

从业务需求层面考虑稳定性，主要是做两个方面，一是业务需求过滤(价值判断)，二是需求模型简化。

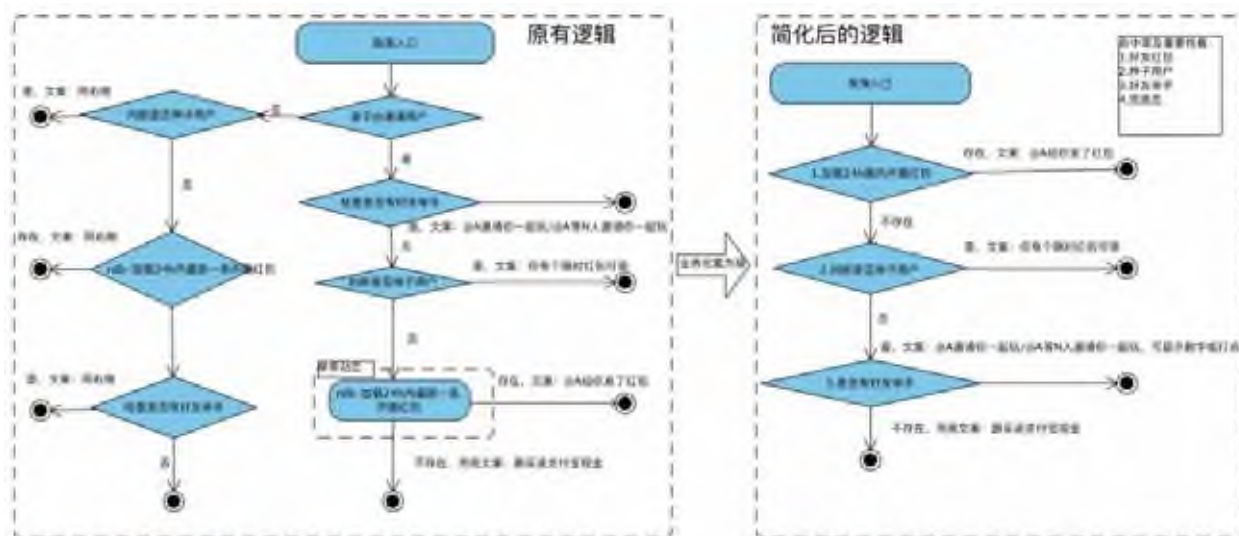
关于需求价值判断，尤其是对于创新型产品，产品设计同学思维很发散，天马星空idea层出不穷，这是创新源动力，是好事，但作为技术人员精力很有限，必须脚踏实地的思考可行性问题，必须要对需求方原始需求进行合理质疑，砍掉一些表面上的浮华，实际没有核心价值的伪需求，同时需求精简模型，将有限的研发精力投入到真正有业务价值的地方。



做价值判断，必要时候用数据分析、数据驱动手段去证明，很可能分析结论发现整个盘子的天花板就在那里，那么一些需求就自然没有存在的理由，无价值的需求上去了除了浪费开发资源，还会带来系统复杂度的无意义升高，稳定性风险自然就升高。

有些时候业务方倾向于把一个需求方案复杂化，我们需要做模型简化，考虑是不是有必要设计如此复杂的规则。如果我们把规则简化到20%以下，是否可以满足90%以上的需求了，而剩下10%是否可以有更轻量方式去解决。

如下针对淘友圈所在的我的淘宝入口，和业务上达成一致后，简化后的逻辑实现复杂度下降50%，那么稳定性风险也会下降50%，且对业务上带来的是同样的用户体验，体感如下：



如今纷繁复杂的无线页面形态并不是越绚越好，而是需要真正找到对用户有吸引力，有价值的点。正如逍遥子和蒋凡在2020年双11所要求的，简单、好逛。很多时候，技术必须要协助产品去做减法，而业务上的去伪存真化繁为简对于系统复杂度往往带来决定性影响，做稳定性保障必须重视这一环节的把控。

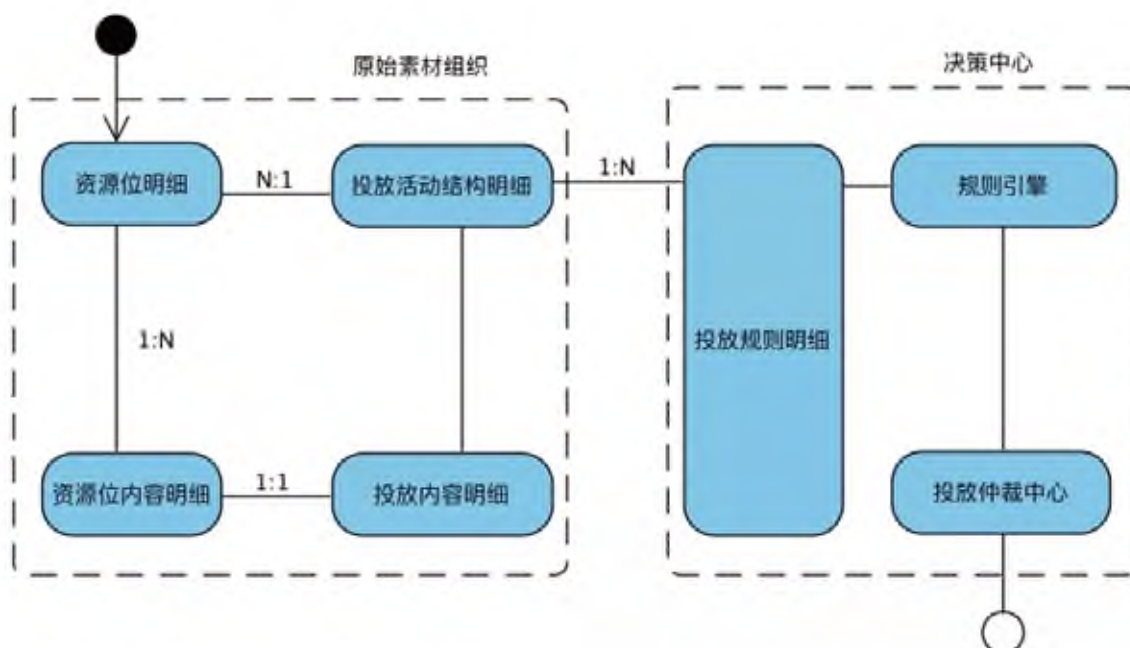
业务领域模型稳定性

基于对业务的理解把控，我们拿到一个相对靠谱的业务需求，开始抽象领域模型。领域模型是从纯粹客户需求转化为技术人员可理解的语言，是对业务的高度抽象，根本目的是帮助我们理解和分析业务，以指导进一步的技术实现。

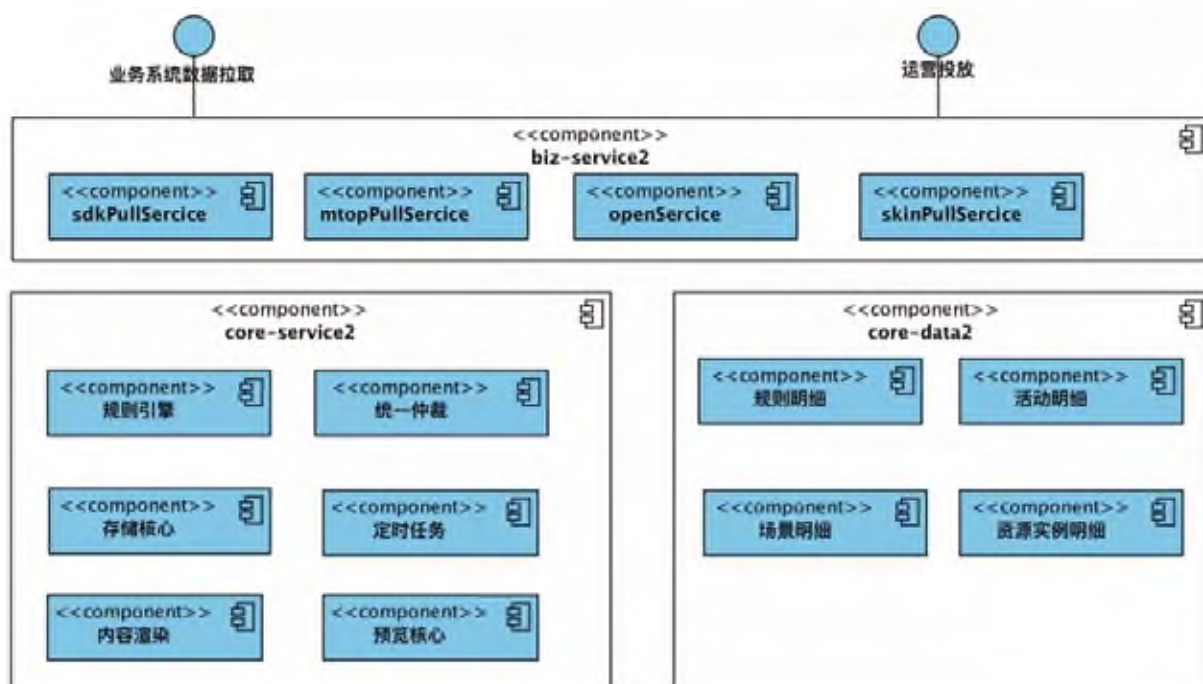
这个阶段需要与产品经理反复对焦，充分理解题意深挖出潜在逻辑(实际必须要支持的逻辑，但是局限于需求方认知在需求阶段没有提到)，使用uml工具梳理出面向对象编程中的对象，以及对象之间的转化关系，需要抓住整体而不是一上来就陷入细节。

比如我们建一个面向页面资源位的投放系统，分析后可以得到如下：

领域模型示意



系统领域模块划分示意



领域模型分析不等于数据库ER图设计，但有了清晰准确的领域模型，再细化出ER图就变得很确定性了。

这一步是从微观层面理解业务，基于业务架构抽象出了业务的核心主体及其主体之间的协作关系，确保清晰准确的理解了需求现在要做，未来要做，未来可能做的事情有哪些，基于这个充分的理解，设计稳定可靠可扩展的模型，确保业务领域模型的稳定。

技术架构与选型稳定性

技术架构与选型这一步需要确定编程语言，数据库，系统拆分，以及你的系统之间大致是如何产生关联作用，并最终提供完整的业务能力。

技术选型主要包括：技术框架选型，数据存储索引选型，数据交互流转选型等大的模块。对配置有专门中间件团队的公司，出于效率与技术统一性考虑，代码框架一般都有一套成熟稳定的配套方案，业务团队无需在这个上面过渡投入。比如在阿里，可以在一站式研发协同平台aone上一键创建最新应用。主要精力是花在数据存储与数据交互流转技术的选择上，需要结合自身业务特点来做选择。基于业务架构与领域建模、数据规模、未来趋势、团队能力限制综合决策权衡选择合适的架构，没有最好的架构，只有最合适的架构。同时需要有对业务的预判能力，至少产品主线上可能出现的较大变化，要预留架构上的可能性。

比如系统是否有必要一开始就应该考虑到读写分离，拆分为几个系统。比如基于业务特性拆分成了读写分离的A，B两个系统，A主要做了门面抗流量，B主要异步任务，定时写入，这样避免因为瞬时异步流量过高影响生产读服务。

但因为项目人员变动较大，时间紧任务重，导致没有坚守，有些服务似乎直接写在A中更加节省工作量，妥协一下，这样A与B的职责越来越模糊，久而久之A系统中也有了较多的瞬时流量风险，风险就不可控。这就要求必须从架构上确定系统职责边界，该撕的一定要撕，如果表面总是一团和气，必然是在某些方面做了一些放弃，把所有的毛刺都按在床底下，日积月累总有一天会爆发。

而如果拆成读写分离，势必会导致有些其实都强依赖的模块，不太好界定的模块也要远程通信。那么可以达成共识，统一提供一个非client类型的公共二方包，在B中开发，同时供给A、B使用。

而如果你做为上游入口，是否考虑提供spi机制，避免后续接入N个下游，就要引入N个二方包的情况。

比如考虑一致性保障策略是用分布式事务，还是使用差错补偿机制来处理。

比如关于存储，我们是需要用nosql存储还是关系型数据库，还是按照业务特性写多份异构存储来提供更好的性能。做这一步决策就需要综合理解mysql，lindorm特性及应用场景，lindorm与redis做为索引的差异化，以及opensearch的可靠性、延时性问题。数据库设计考虑不足，导致容量瓶颈。数据库的分库分表，则应该考虑未来5至10年的业务规模。关于存储与索引技术的选型，是整个技术选型中的核心，后续将另谋篇幅详谈。

技术架构及选型直接决定了我们的系统结构上是否稳定合理，决定了在未来可预期的时间段内是否会被推到重来，是系统稳定的基础。

代码实现规范稳定性

业务架构->领域建模->技术架构与选型决定了我们整个工程的宏观可行性与各个关键节点的解决方案。接下来进入到编码环节，但这个环节来说，不同的施工队、不同的人员操作上是不一致的，可能一两个点的细节看不出主要影响，但是两边引起质变，最终会决定成败。我们需要有一套规范来保障细节的可控与标准化，来确保系统微观层面的稳定性，比如：

- 比如包装类型与基本类型使用场景，判断对象相等方法，对象做json序列化注意事项。
- 金额字段处理，统一规范的封装工具类。金额统一收口服务端处理。
- 响应给上游的result到底是代表通信成功，还是业务成功。这个在注释上必要详尽说明，避免异常情况下扯皮。
- 数据库，核心资源操作应始终对照：一锁二判三更新的基本原则。
- 尽量避免使用|、&、异或(^)、位运算(<<)，因为可读性较差，代码的可读性可维护性是除了代码本身业务价值外，技术人员对公司最大的贡献。

关于代码规范的制定，推荐《阿里巴巴Java开发手册》一书，它是阿里内部Java工程师所遵循的开发规范，涵盖编程规约、单元测试规约、异常日志规约、MySQL规约、工程规约、安全规约等，这是近万名阿里Java开发人员经验总结，并经历了多次大规模一线实战检验及完善，具备较强的参考意义。

代码通用模块稳定性

结合自身系统特点，理清楚变与不变。把可以反复使用的部分、易出错的部分以及系统核心引擎抽象出来，做为系统不变的部分。把伴随着业务的变化或者新业务方接入而不断调整的部分提取出来，做为系统中可变的部分。

对于不变的部分，运用设计模式及常用套路将其固化下来形成公共模块，降低类似功能重复编写带来的风险，提高增量业务迭代效率。这部分代码投入核心精力让它像工具类一样稳定可靠，之后反复运行。这部分的抽象决定了系统的核心代码层次，保障大楼的上相似的模块统一稳定，有统一的管控手段。

对于变的部分，提供扩展点。这部分是会随着业务的变化而不断迭代，同时要考虑让变化的部分具备隔离性。即当改动一个子业务需求时，尽可能从从架构上限定住它的影响范围。

追求高内聚，低耦合，满足开闭原则易扩展易维护的代码层次结构。

当然这里要避免一个误区，即滥用设计模式，或者为了模式而模式。比如在一个小小的项目里本来简单一个方法调用就能实现，确要过度套用设计模式去编码，折腾出来好几层，开发成本高且给代码的维护带来了困难。私下练手

可以。但在工程层面，则要更多的考虑实用价值，实际一定要结合场景需要。

下面简要列举几个常见的点：

统一对外服务层异常兜底

对于多数应用来说为上游提供hsf是其主要职责。标准是当service层抛异常，我们需要自己处理掉异常，以Result结果中的结果码的形式与外界通信，而不是直接抛运行时异常给业务方。那么我们可以抽象出AOP层，对Service层的异常统一捕获，返回兜底错误码给上游。

统一抽象摘要日志模块

方法的核心出入参，是我们监控的关键。但是穿插在业务代码中打印，总是容易遗漏而且侵入性很强。那么可以抽象摘要日志注解模块，无侵入的打印方法的摘要日志。

统一下游依赖模块

在淘系，我们做导购型产品，基本绕不开淘系3C，即UIC(用户中心)、TC(交易中心)、IC(商品中心)，可能还会有一个SC(店铺中心)，而这些中心又因为历史的原因提供了多套对接查询方法。曾经看到一个系统中对IC的直接依赖有10+处之多，同样是查询商品对象，但由于不同的开发人员依赖了不同的方法。

那么每当大促链路梳理的时候，或者IC包做升级的时候，就需要梳理回归多个入口，给系统的链路梳理带来了极大的不确定性。

正确的做法是我们对于同一个下游入口，包装出唯一的代理类，唯一的方法，统一维护、监控，反复使用。这里的下游不仅仅指业务系统，同时包括对于一些中间件的依赖，比如：针对hbase、redis、ldb、opensearch、odps的访问封装。

Java线程池使用

单个应用对异步线程的管理，应该有统一的类收口，使用者只需要传递线程池名及所需要的变量即可。

这样通过统一的监控配置一目了然就能看到整个应用对异步线程池的使用管理情况，快速诊断出是哪里的线程池使用不合理导致的系统线程数飙高报警，也便于后续的交接维护。这个类就可以完成对线程池的所有幻想,屏蔽掉对线程池工具类的直接访问，至少包括这些类的行为：

- 1) 创建指定线程池
- 2) 并发执行/单个执行
- 3) 同步执行/异步执行

统一的线程池管理工具类示例：

```
1  /**
2   * 说明:通用线程池工具类
3   */
4  public class CommonExecutorManager {
5
6      public static final String POOL_DEFAULT = "POOL_DEFAULT";
7
8      /**
9       * 线程池map
10      */
11     private Map<String, ExecutorService> threadPoolMap = null;
12
13     /**
14      * 默认线程池
15      */
16     private ExecutorService defaultExecutor = null;
17
18     @PostConstruct
19     public void init() {
20         threadPoolMap = new HashMap<>(2);
21
22         /**
23          * 默认线程池
24          */
25         ThreadFactory defFactory = new com.google.common.util.concurrent.ThreadFactoryBuilder().setNameFormat(
26             "AsyncManager-default-%d").build();
27         this.defaultExecutor = new ThreadPoolExecutor(8, 16,
28             100L, TimeUnit.MILLISECONDS, new LinkedBlockingQueue<>(1024), defFactory,
29             new ThreadPoolExecutor.DiscardPolicy());
30         threadPoolMap.put(POOL_DEFAULT, defaultExecutor);
31     }
32
33     @PreDestroy
34     public void close() {
35         if (defaultExecutor != null) {
36             defaultExecutor.shutdown();
```



```
37     }
38     for (ExecutorService pool : threadPoolMap.values()) {
39         pool.shutdown();
40     }
41 }
42
43 /**
44  * 异步执行supplier
45  *
46  * @param poolName 线程池
47  * @param supplier supplier
48  * @param <T>      结果类型
49  * @return
50  */
51 public <T> CompletableFuture<T> supplyAsync(String poolName, Supplier<T> supplier) {
52     if (poolName == null) {
53         poolName = POOL_DEFAULT;
54     }
55     ExecutorService executorService = threadPoolMap.getOrDefault(poolName, defaultExecutor);
56     return CompletableFuture.supplyAsync(supplier, executorService);
57 }
58
59 /**
60  * 并发执行supplier, 并等待结束
61  *
62  * @param poolName 线程池
63  * @param supplierList supplier list
64  * @param <T>      结果类型
65  * @return 结果列表
66  */
67 public <T> List<T> supplyListSync(String poolName, List<Supplier<T>> supplierList) {
68     if (poolName == null) {
69         poolName = POOL_DEFAULT;
70     }
71     ExecutorService executorService = threadPoolMap.getOrDefault(poolName, defaultExecutor);
72     List<CompletableFuture<T>> futures = supplierList.stream()
73         .map(supplier -> CompletableFuture
74             .supplyAsync(supplier, executorService))
75         .collect(Collectors.toList());
76     return futures.stream()
77         .map(CompletableFuture::join)
78         .collect(Collectors.toList());
79 }
```

```
80  /**
81   * 查询线程池
82   *
83   * @param poolName 线程池名称
84   * @return
85   */
86  public ExecutorService getExecutorService(String poolName) {
87      return threadPoolMap.getDefault(poolName, defaultExecutor);
88  }
89  /**
90   * 创建线程池
91   * @param corePoolSize 核心线程数
92   * @param maxSize 最大线程数
93   * @param threadName 线程名称
94   * @param daemon
95   * @return
96   */
97  public static ExecutorService createThreadPool(int corePoolSize, int maxSize, String threadName, boolean d
98      aemon) {
99      ExecutorService executorService = new ThreadPoolExecutor(corePoolSize, maxSize, 5, TimeUnit.MINUTES,
100      new
101      LinkedBlockingQueue<>(), new com.google.common.util.concurrent.ThreadFactoryBuilder().setNameFo
102      rmat(
103      threadName + "-%d").setDaemon(daemon)
104      .build());
105      return executorService;
106  }
107  /**
108   * @param nThreads
109   * @param threadName
110   * @return
111   */
112  public static ScheduledThreadPoolExecutor createScheduledPool(int nThreads, String threadName) {
113
114      return new ScheduledThreadPoolExecutor(nThreads,
115      new com.google.common.util.concurrent.ThreadFactoryBuilder().setNameFormat(threadName + "-%d").
116      build());
117  }
118  public static ExecutorService newCachedThreadPool(ThreadFactory threadFactory) {
```

```
119     return new ThreadPoolExecutor(0, Integer.MAX_VALUE,  
120         60L, TimeUnit.SECONDS,  
121         new SynchronousQueue<Runnable>(),  
122         threadFactory);  
123     }  
124 }
```

定时任务使用

同管理线程池一样，单个应用对定时任务的使用，是可以有通用的部分抽象出来，比如对于网格任务中的根任务与子任务的识别。具体的定时任务实例只需要聚焦在子任务的特性读取与单条任务的计算处理上。

这样通过统一的监控配置一目了然就能看到整个应用定时任务执行的情况，快速诊断出是哪个定时任务的运行导致系统资源报警。统一的job抽象类示例：

```
1  public abstract class AbstractMapJob extends MapJobProcessor {  
2  
3      @Override  
4      public ProcessResult process(JobContext context) throws Exception {  
5          try {  
6              String taskName = context.getTaskName();  
7              Object task = context.getTask();  
8  
9              if (isRootTask(context)) {  
10                 //生成子任务  
11                 List<IndexChildTask> childTasks = generateChildTasks(MsgSwitch.childTaskCount, context);  
12                 return map(childTasks, getChildTaskName());  
13             } else if (getChildTaskName().equals(taskName)) {  
14                 try {  
15                     //子任务处理  
16                     handleChildTask(task);  
17                     return new ProcessResult(true);  
18                 } catch (Throwable t) {  
19                     log.error("handleChildTask error.", t);  
20                     return new ProcessResult(false);  
21                 }  
22             }  
23         } catch (Exception e) {  
24             log.error("process error", e);  
25         }  
26         return new ProcessResult(true);  
27     }  
28 }
```

```
27     }
28
29     /**
30     * 根据总记录数拆分成子任务
31     * @param taskCount 要拆分的子任务个数
32     * @return
33     * @throws Exception
34     */
35     private List<IndexChildTask> generateChildTasks(int taskCount, JobContext context) throws Exception {
36         long totalCount = getTotalCount(context);
37         log.info("total count: {}", totalCount);
38         if (totalCount == 0) {
39             return Collections.emptyList();
40         }
41         long taskSize = (long) Math.ceil(totalCount * 1.0 / taskCount);
42         List<IndexChildTask> taskList = new ArrayList<>();
43         for (long i = 0; i < taskCount; i++) {
44             long start = i * taskSize;
45             IndexChildTask childTask;
46             if (i == taskCount - 1) {
47                 childTask = new IndexChildTask(start, totalCount - (i * taskSize));
48             } else {
49                 childTask = new IndexChildTask(start, taskSize);
50             }
51             taskList.add(childTask);
52         }
53         return taskList;
54     }
55
56     /**
57     * 获取要处理的总记录数
58     * @return
59     * @throws Exception
60     */
61     public abstract long getTotalCount(JobContext context) throws Exception;
62
63     /**
64     * 处理子任务
65     * @param task
66     */
67     public abstract void handleChildTask(Object task);
68
69     /**
```

```
70    * 获取子任务名称
71    * @return
72    */
73    public abstract String getChildTaskName();
74 }
```

消息订阅处理

将消息接收的注册、序列化为业务类型，异常处理，监控日志做统一的封装，收口，使得消息的消费监控变得简单易执行，同时便于后续的开发维护,示例类：

```
1  public abstract class MetaqCommonConsumer<T> implements MessageListenerConcurrently {
2
3      private MetaPushConsumer consumer;
4
5      private String topic;
6      private String tag;
7      private String consumerGroup;
8
9      /**
10     * 具体的消息实体类型
11     */
12     protected Class messageObjClass;
13
14     public MetaqCommonConsumer() {
15         // 获取父类带泛型的类型
16         Type genericSuperclassType = getClass().getGenericSuperclass();
17         ParameterizedType parameterizedType = (ParameterizedType)genericSuperclassType;
18         Type[] actualTypeArguments = parameterizedType.getActualTypeArguments();
19         messageObjClass = (Class)actualTypeArguments[0];
20     }
21
22     @Override
23     public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt> list, ConsumeConcurrentlyContext context) {
24         if (list == null || list.isEmpty()) {
25             return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
26         }
27         for (MessageExt msg : list) {
28             return consumeSingleMsg(msg, context);
29         }
30     }
31 }
```

```
29     }
30
31     return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
32 }
33
34 /**
35  * 消费消息
36  *
37  * @param msg
38  * @return
39  */
40 private ConsumeConcurrentlyStatus consumeSingleMsg(MessageExt msg, ConsumeConcurrentlyContext context) {
41     if (msg == null) {
42         return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
43     }
44     try {
45         String bodyJsonStr = new String(msg.getBody(), "UTF-8");
46         // 转换为实际类型
47         T messageObj = (T)JSON.parseObject(bodyJsonStr, messageObjClass);
48         return process(messageObj, msg, context);
49     } catch (Exception e) {
50         log.error("consumeMessage error, {}", JSON.toJSONString(msg), e);
51     }
52     return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
53 }
54
55 /**
56  * 消息处理方法
57  *
58  * @param messageObj 转换为具体类型的消息obj
59  * @param messageExt 完整的消息体
60  * @return
61  */
62 protected abstract ConsumeConcurrentlyStatus process(T messageObj, MessageExt messageExt,
63                                                         ConsumeConcurrentlyContext context);
64
65 /**
66  * 初始化
67  */
68 public void start() {
69     try {
70         consumer = new MetaPushConsumer(consumerGroup);
```

```
71
72     consumer.subscribe(topic, tag);
73     consumer.registerMessageListener(this);
74
75     consumer.start();
76     log.info("start success consumerGroup: {}, topic: {}, tag: {}", consumerGroup, topic, tag);
77 } catch (Exception e) {
78     log.error("start consumer error consumerGroup: {}, topic: {}, tag: {} , errorMsg: {}", consumerGroup, topic,
79         tag, e.getMessage());
80 }
81 }
82
83 public void setTopic(String topic) {
84     this.topic = topic;
85 }
86
87 public void setTag(String tag) {
88     this.tag = tag;
89 }
90
91 public void setConsumerGroup(String consumerGroup) {
92     this.consumerGroup = consumerGroup;
93 }
```

报文网关交互

比如报文网关系统无论和外部怎样的通信区别变化的部分在于报文格式不同需要有不同的行处理逻辑，不变的是通过http进行request、response的通信逻辑，以及对json、xml格式报文的解析方式。

文件解析模块抽取

而文件网关系统则一定是先将文件转换成流读取到内存，再转换成一行为，针对这一行做特性化处理，最后关闭流操作，那么我们可以把文件处理的逻辑抽成模板方法，后续来一种新的文件，只需要实现模板方法，专注处理某一行的业务逻辑就可以了。

通过通用模块的抽取，我们可以极大的规避同一类的风险，也为后续的统一发现能力的建设提供了基础。

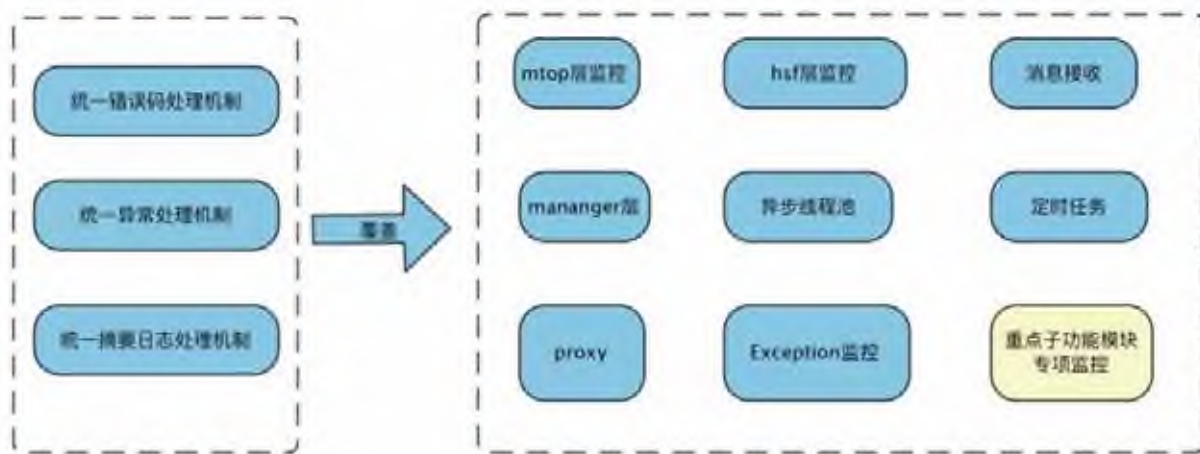
代码实现规范稳定性

这里主要强调为了做发现能力建设，系统需要满足怎样的设计更合理，围绕代码实现谈谈建设思路。

在前述架构合理、分层得当、代码规范到位、兜底保护完善的情况下确保我们项目可以顺利验收上线运行。

但在运行过程中都会受到各种因素的影响，难免还是会出问题，出问题并不可怕，关键是能否尽可能早的在第一时间发现问题并解决问题，而发现问题往往比解决问题更具有挑战性。试想如果不该打印日志的地方打印了日志，不该配置报警的地方配置报警，错误码粒度杂乱无法区分，那么真正的问题就不能被及时发现。

对于一个直面无线app的服务端应用，系统层面很容易得到如下通用视角：



抽象来看系统层面发现能力建设主要围绕以下4个要素：



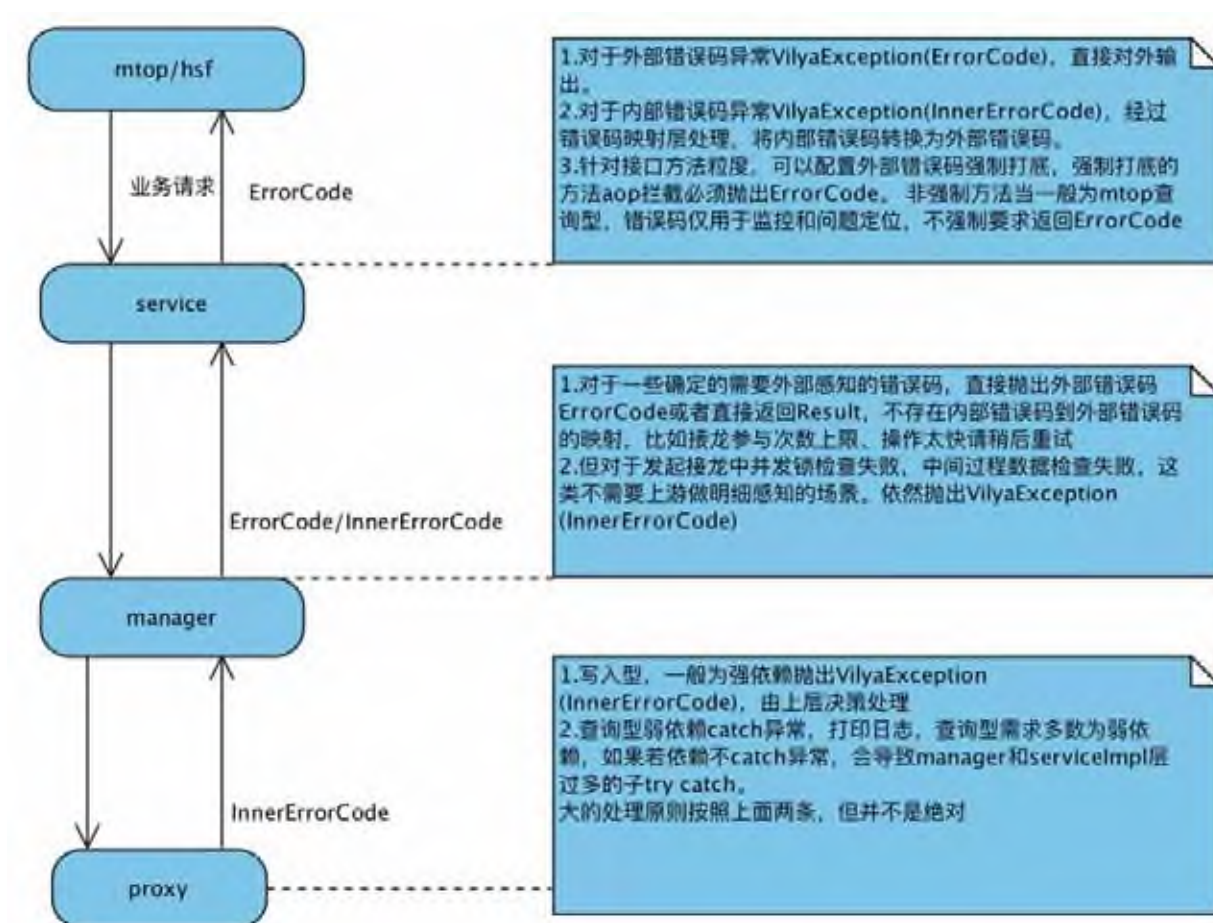
绝大部分的稳定工作都是围绕上述四个要素在投入，梳异常链路、映射错误码、加日志、加监控。针对这些问题，我们需要有统一的、尽可能低成本低侵入的解决方案和标准，在日常的研发过程中就尽量随时准备好。

统一错误码标准

需要建立错误码标准，分外部错误码、内部错误码，建立错误码统一维护策略。

需要上游业务感知，或者方便上下游之间定位问题，可以定义外部错误码，这部分错误码的变化需要非常谨慎，特别是上游已有引用已有理解的情况下。而内部错误码是指系统的各个层次之间内部执行异常得到的错误码，通常不对外输出，主要用于业务场景监控与异常定位。

错误码的设计要结合自身业务特点，制定适合自身业务的错误码。比如银行的结果码细分品类繁多，不得不用数字码来替代，但导购应用需要前端感知的结果码很有限，这个时候可读性占据主导地位，就不建议输出使用数字编码，这样的格式机器可读，但人不可读。



统一异常处理机制

面向异常编程，定义统一业务异常类，通过业务错误码进行异常分级分类。当抛出原生运行时必须打印完整堆栈。当抛出系统业务异常，对于业务预期之中的则权衡是否需要堆栈信息。

异常分类的方式有两种，一种是通过继承设计不同的异常类来做异常区分，另一种是通过在异常中增加异常结果码。实践上，推荐使用异常结果码这种比较灵活和方便的方式。

统一日志输出规范

日志是我们发现、定位问题的基础，几乎绝大部分的稳定性工作都是在围绕着日志进行。

基本原则

- 1) 鹰眼id、压测标、ip、时间、线程等业务无关的必要信息，由日志框架层统一解决。
- 2) 日志文件功能分级，在默认application.log之外至少分出方法xflush摘要日志文件、异常日志文件。
- 3) 统一监控日志格式，针对xflush日志文件，定义行列结构
- 4) 谨慎打印日志，不多打，也不少打，注意日志打爆磁盘的问题。

设计示例

格式化摘要日志：

日期|eagleid|ip|压测标识位|日志级别|预留1|预留2|线程号|((层级^场景)(类名^方法名^业务成功表示位^结果码列^异常根原因^耗时)(监控属性1^监控属性2^监控属性3)(业务属性1^业务属性2^业务属性N)

其中监控属性区：存放比如用户是否种子用户,返回动态条数等，可能用于监控分析的字段，主要用于监控。

业务属性区：存放比如userId，动态id，红包id之类的字段，主要用于异常定位。

灵活而业务无侵入的使用方式：

```
1  /****
2   * 构造开圈页面入口的动态消息
3   * @param friendPageDTO
4   * @return
5   */
6   @Digest(logger = "COMMON-XFLUSH",layer = "manager", success = "!ret?.isEmpty()",monitor = {"ret?.size()"},
7   attrs= {
8       "userId"})
9   public List<OpenTipsVO> getTaoMomentsOpenPageTips(Long userId, Optional<FriendPageDTO>friendPageD
10  TO) {
11      //do something
12  }
```

输出效果:

```
2020-12-22 16:40:44.618|0b13045816086264438145928e77fb|11.23.98.0||WARN |HSFBizPro-
cessor-DEFAULT-8-thread-47|(manager^-)(TaoMomentsEnterTipsManager^getTaoMoment-
sOpenPageTips^Y^-^-^86ms)(2^-^-^-^-)(2200782267981)
```

统一贴合业务的监控机制

监控是手段和目标，需要在错误码、异常、日志三个要素的基础之上基于业务特点细化到每一段核心代码逻辑，能够及时发现预期之外的运行逻辑，报警精准程度则直接决定了线上运维成本。根据适用场景可以把分为通知型与大盘型，通知型是指直接通过某种通讯工具报警触达至接收者，而大盘型是指用户定期主动浏览，通过实时或者T+1(h+1)报表能够进行业务统揽以发现关键问题。

通用监控告警层

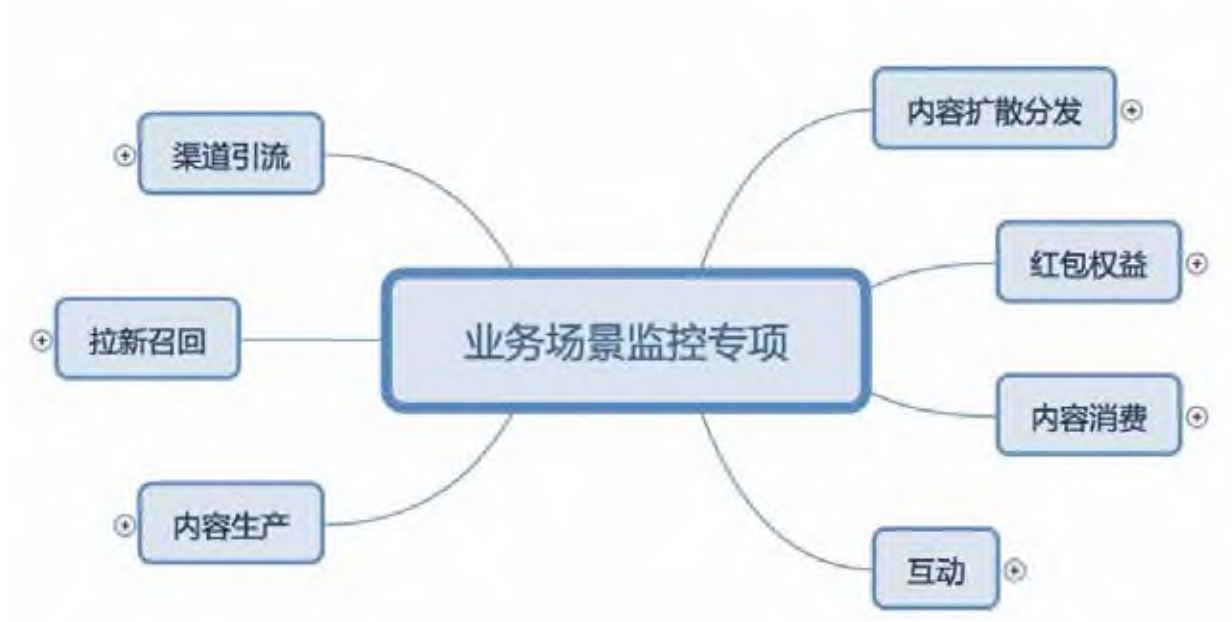
这里抛开硬件及基础设施监控先不谈，仅看系统代码层面的情况。

通用监控是指具有业务无关性，只要是无线服务端应用都会需要的监控。

比如mtop层方法执行超过500ms、错误码环比大涨、异常量大涨、出现未知异常。这类异常应该由统一解决方案，无需按照业务模块重复建设。

特定业务场景告警层

结合系统所承载的业务本身，对关键链路做特定监控，比如淘友圈特定业务场景监控：



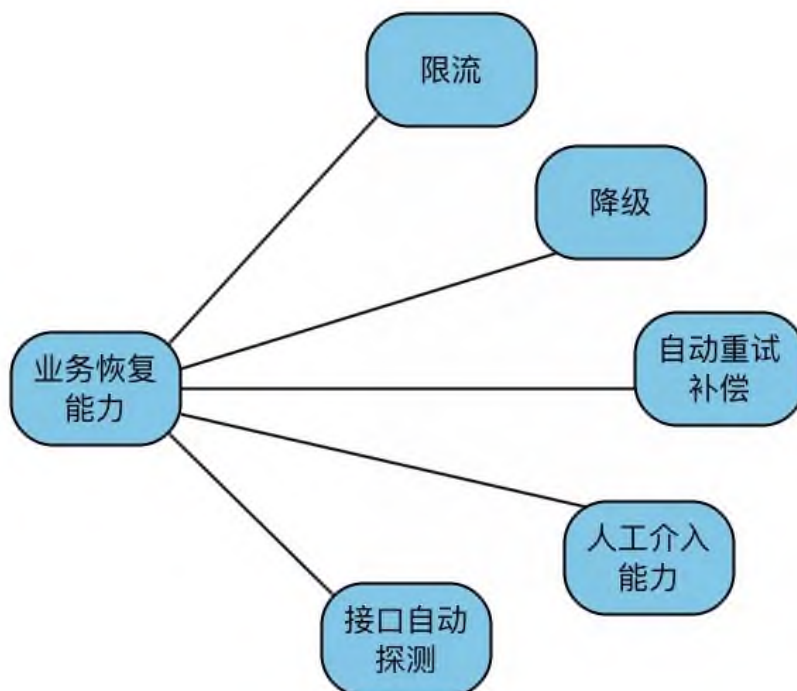
建立业务系统统一大盘

针对一个系统，我们会有aop监控层、各个子业务场景监控点。随着功能的叠加监控点会越来越多。这些监控点会在具体场景出问题时报警给我们。但假设就在某一个时刻，想知道系统各业务是否都运行正常，就需要有全局视角的业务大盘。在这个大盘上可以覆盖系统的主要业务场景，承担系统晴雨表的效果，关键时候可以一目了然、心中有数。

恢复能力基础建设

这里主要强调为了做发现能力建设，系统需要满足怎样的设计更合理。

当通过发现能力发现问题后，进一步面临的是如何恢复的问题，一般有如下几种手段：



业务降级

当发现服务接口超时，定位到是某一个下游依赖所致。且判断该依赖并非产品核心流程，此时需要一键关闭掉对该服务的调用，进而保障主流程能够正常工作。要求我们在编码的时候能够梳理强弱依赖，对弱依赖增加必要的降级开关，并形成预案。

重试补偿与人工介入

重试补偿分自动重试补偿和人工补偿两种情形。

自动重试如消息重试，线程内循环重试，一般是因为机器进程中断，网络延时等原因导致的差错场景，这类场景经过系统自动重试机制是可以完成的。

而人工补偿主要面对人为引起的脏数据，或者依赖外部机构带来的不可控因素。

比如一个客户的支付单据，因为国外银行端的接口bug，导致返回了不确定的流水单号或者金额数据或者指定导致校验不通过，因而支付状态未推进。这种情况可能必须要通过客服手段去联系银行人员，因为确认后我们需要手工触发单据到支付完成状态。如果偶尔几笔可以由开发人员通过数据订正手段完成，如果这样的情况时有发生，那么就需要考虑是否开发差错恢复功能，让业务运营可以在限定条件下直接介入推进单据状态，完成状态补偿。

自动切换

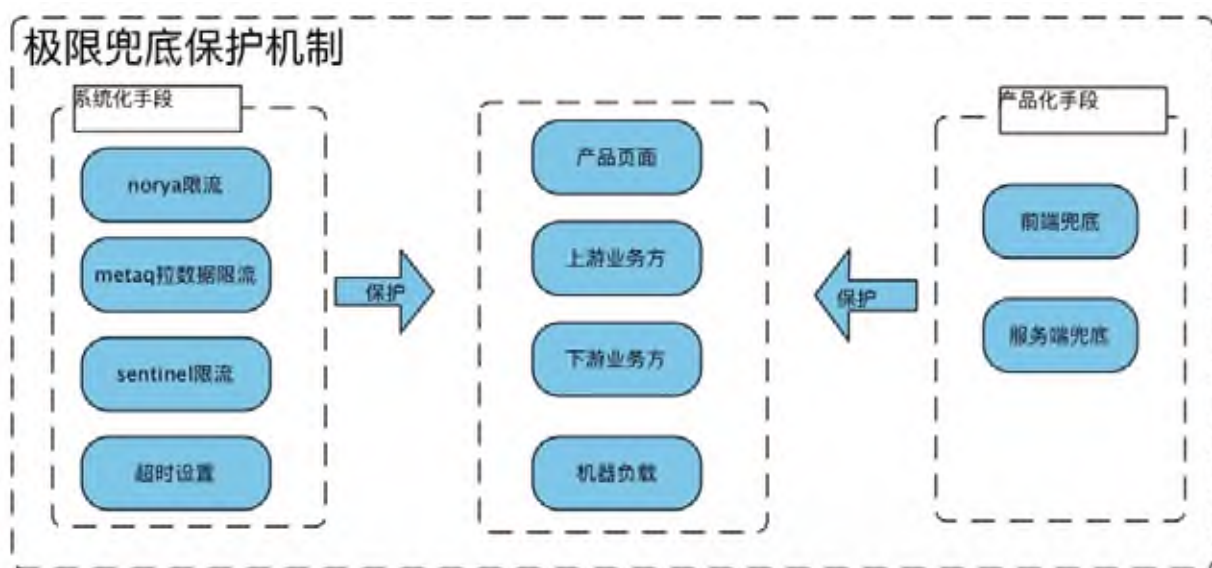
这里想说的其实类似于分布式服务发现Zookeeper的自动探测能力，zk通过探测容器是否存活决定是否将容器从可用列表中移除或增加。而对于业务接口我们有时也需要这种探测能力，通过探测接口的可用性，决定是否启用。尤其在有多个下游服务竞争，且稳定性得不到充分保障的情况下诉求较为强烈，比如用户钱包有N个银行卡可用，当某个银行接口出现故障的时候，我们应该具备自动识别的能力。比如超过一定的错误阈值及失败率，自动将这个银行接口切换为不可用将支付方式置灰并给用户以文案提示，每隔一定的时间再做自动的探测尝试，发现成功后再自动切换为可用状态。

极限兜底保护稳定性

汽车的安全气囊，虽然阻止不了车祸，但很可能会挽救一次生命。兜底保护相当于一些安全手段，比如紧急情况事故无可避免，但是我们可以把损失降低到最小，比如无论在任何情况下，当前端或者服务端出现任何异常的情况下，不允许用户看到诸如500，404，空白页之类的页面。这需要全面的异常分支分析+与产品人员的充分对焦+充分的兜底场景测试。

1) 单机qps限流2) norya自适应系统load限流3) 消息接收处理线程数控制，就是无论什么时候，要保障你的系统是活着的，研发人员必须建立这样的认识。

使用meta的线程数来进行削峰平谷，保护应用。



资金操作稳定性

资金操作稳定性保障即资损防控

1) 如何定义资损？

广义的资损应该包括资金流转活动中，资金未按照业务规则预期流动的情况，导致业务参与方中的任何一方或多方遭受了资金损失。比如红包活动中，如果因为技术故障给用户少发了5毛或者多发了5毛，无论谁吃了亏，沾了光，只要是不符合预期的，均定义为资损，都应该界定为资损。

2) 资损也可以归属于稳定性范畴，为什么要单独拎出来？

个人理解资损是稳定性问题中危害最大波及面最广的一种，尤其是对金融系统来说。

我曾经在支付宝金融核心组参与研发工作4年，看过公司内外不少案例。深刻体会到资金风险可能给业务及公司带来的灾难性后果，金融软件要求我们在所有的地方都做到最好，容不得半点损失，多一分钱少一分钱都要复盘追责。彼时资损案例的发生直接与绩效直接挂钩，每一行代码都需要被自动化脚本覆盖到，花5分钟写的代码，可能需要花2小时来编写、运行测试脚本，要求极其严苛，但在金融系统中，这一切都是值得的。

3) 电商团队资损防控看法

近两年以来，淘系对于资损重视程度逐渐收紧，不少防控机制从蚂蚁引入。因为虽然电商不属于金融系统，但做为

一个经济体，任何细小的波动都可能给品牌形象带来巨大破坏和影响。但无论如何，其要求的严苛程度与金融软件是不一样的，不能所有的问题都一刀切，不断叠加的机制、流程会带来新的问题。尤其针对电商创新型业务，是需要结合具体业务场景具体分析，在稳定性保障与业务创新迭代效率之间找到一个平衡点，定义最合适粒度的资损防控策略。

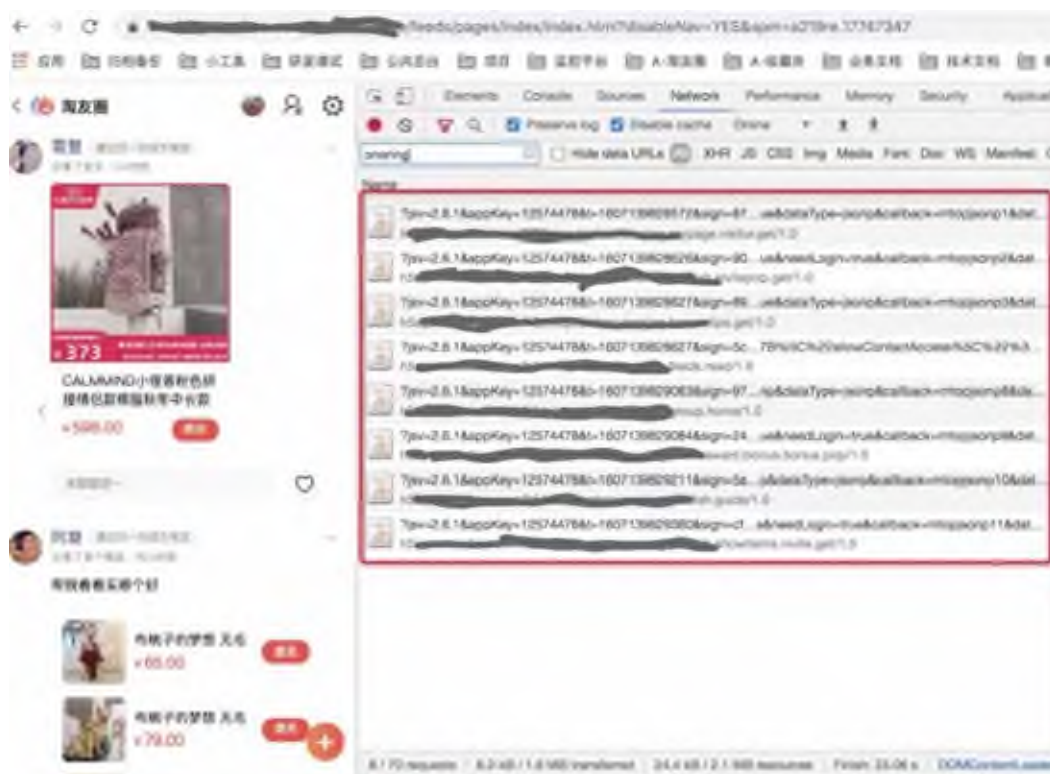
持续重构与稳定性的关系

在业务先赢的时代背景下，特定时候不得不做妥协，不得不上临时方案，这样导致即使原本优雅稳定的系统实现，依然会被一点点扭曲，形成越来越多的技术债。这些技术债会成为越来越大的绊脚石，除了会障碍业务快速发展，更会直接带来额外稳定性风险和巨大的人力开销。我们必须要做持续的重构来偿还，下面举几个示例：

无线页面接口爆炸问题

比如下图所示的某产品首页，一旦进去首页会同时打开8个接口，而这种暴涨增长的接口，势必给后续的开发迭代及稳定性工作带来较大挑战，同时也影响了低端机用户体验。你可能说这一个业务很复杂，所以不得不使用8个。但我们抓取手淘首页接口，发现那么复杂的页面结构，只要一个接口搞定了，因为统一的协议标准定义的科学、通用。

而抽象出这样的接口，是需要对产品形态及未来趋势、前后端研发协作机制有充分考虑。



系统对商品服务的依赖问题

比如对IC、主搜索依赖，全站点下对商品的状态、价格统一处理的问题，这些问题在一两个功能下都不需考虑太多问题。但功能点越来越多，问题就会逐步出现，就需要进行统一收口，把复杂易出错的问题集中在一个地方，集中精力解决一次。

feeds流加载逻辑问题：

做为一个朋友圈形态的电商内容场的页面，其feeds块类型包括商品、红包、游戏、玩法各种类型，需要从商品、营销、关系、互动获取各种类型数据来透出，那么怎样去组织代码结构，才能在后续尽量小的侵入去接入新品类，需要斟酌推敲。

我的淘宝动态入口读写模式问题：

当你进入我的淘宝，过去的做法是直接pull的形式，拉取你的好友的动态，为了保护我淘页面的性能，超时设置为100ms。但随着动态类型越来越多，超时的情况时有发生，这个时候就需要基于稳定性与可持续扩展层，提出新的方案并改进既有代码。

诸如上述所列，有些重构可能和短期的业务目标无直接的联系，尤其是产品运营同学不能直接感知的情况，很容易被忽略，得不到重视。但如果不做，坑就会在那里越积越多，引发线上的故障概率就会越来越大。

可压测性能力建设

这里主要强调为了支持可压测，系统需要满足怎样的设计更合理。

导购型业务流量大是其最主要特点，当我们开发新功能的时候，很容易因为节奏的问题而急于上线，完全不考虑这个代码在双11的情况下是否OK。导致到了压测的时候才发现这段逻辑没有识别压测流量，可能会导致将压测数据写入到正式库中。理想的状况是在项目一开始，就考虑对压测的支持方案,预留扩展点，项目时间紧可以理解，但是一定要能提前布局。否则会因为突击压测能力改造导致代码侵入极强，时间上比较被动，最后临近封网，有些流程其实没有经过压测，也只好硬着头皮上线，具有较大不确定性。

服务端CI能力建设

这里主要强调服务端CI能力，依然是我们系统工程中需要代码实现的部分。

对于服务端开发来说，自动化回归主要关注单元测试和接口集成测试。单元测试关注单个类单个方法的逻辑正确

性。接口集成测试一般关注单个系统单个service方法的逻辑正确性。具体这个层面做到什么程度需要依赖业务场景而定，比如高风险、模型稳定的核心系统就必须需要严格要求自动化覆盖率，而创新型快速迭代的新业务讲究快跑覆盖核心流程比较合适。

比如支付宝多数系统都与资金相关稳定性要求极高，要求每次代码变更的自动化行覆盖率甚至分支覆盖率不能下降，且手工测试不计入内，似乎为了规避风险无论怎样严苛的卡控都是一种政治正确。曾有人做过局部统计，发现写1行业务代码需要5行测试代码来覆盖，改一行代码需要重跑整个系统的数百个用例可能耗时一天，以保障其业务系统高度稳定可用，但弊端是降低了生产效率。

而电商内容业务创新型团队，业务变化极快。产品经理idea天马星空，更看重的是业务快速上线快速试错的能力，我们需要两害相权取其轻做决策。针对这样情况建议梳理出业务核心主流程，针对主流程建立自动化回归用例持续运行。

长远来看，服务端CI自动化能力是保障系统核心模块稳定性不可获取的手段，但需要视场景、合理规划。忌不切合和业务实际的一刀切要求。

上下游SLA稳定性

这里主要强调系统之间的SLA，自身做的再好，合作方上下游没有规范好是不够的，依然隐患重重。这里上下游是指使用你系统服务的前端、上游服务端，以及被你调用的下游服务。

SLA只是一种约定，目的是确保接口提供方和使用方能够对接口有一致的理解，能够更好的保障业务活动进行，同时在发生线上故障必须要进行定责的时候，是一份参照标准，一份免责声明。个人认为可以不局限于形式，但一定要可追溯，文档或者清晰的接口注释都具备同样效力。与上下游之间做好SLA协议签订，在金融级系统几乎是必须要做的事情。

但在淘系这边业务开发的过程中，发现有时候两个团队的开发一碰头，或者口头约定或者钉钉上沟通，就开始调用接口了。也不管result.success到底代表是业务执行成功，业务受理成功，还是不确定。或者当时沟通清楚了，但是也没有关键性文档沉淀下来，就发上线了，潜在风险很大。

比如A依赖B以check是否可以给B用户发红包，那就必须对B的入参、出参，结果码有100%的理解，且有明确文档落下来。

```
1  /**
2   *用户资质检查服务
3   */
4   public interface CheckService {
5       Result check(QueryParam queryParam, UserInfo userInfo);
6   }
```

对比一下，如下Result，就很清晰明了，让人一看就很有确定性：

```
1  /**
2   * 开放API2.0_活动投放开放API
3   */
4   public interface DeliveryOpenService {
5       /**
6        * 投放活动-支持单资源位单实例
7        * 1)result.isSuccess=true代表业务上数据写入成功,此时data为工匠平台的活动id,建议业务方做存储，便于后续问题的
        定位排查
8        * 2)result.isSuccess=false情况下，errorCode会有相信错误信息,业务方需做监控并采取必要的处理
9        * @param singleTypeActivityRequest 活动对象
10       * @return <activityId,#activityId#>
11       */
12       OpenResult<Map<String,String>> createActivity(SingleTypeActivityRequest singleTypeActivityRequest);
```

总结

上医治未病,中医治欲病,下医治已病。本文大致围绕系统设计实现各环节，分析了各环节对稳定性保障的影响，同时介绍了对应的一些套路和经验。着眼点主要在于未病、欲病阶段的思考投入，尽可能把一些问题的解决和规避前置均摊到业务需求受理把控、系统分析设计与编码实现阶段，同时持续改进健全，最终达到提高全局稳定性保障工作效能的目的。

• Guava Cache 原理分析与最佳实践

- 作者：梓川
- 出品：淘系技术

在大部分互联网架构中 Cache 已经成为了必不可少的一环。常用的方案有大家熟知的 NoSQL 数据库（Redis、Memcached），也有大量的进程内缓存比如 EhCache、Guava Cache、Caffeine 等。

本系列文章会选取本地缓存和分布式缓存（NoSQL）的优秀框架比较他们各自的优缺点、应用场景、项目中的最佳实践以及原理分析。本文主要针对本地 Cache 的老大哥 Guava Cache 进行介绍和分析。

基本用法

Guava Cache 通过简单好用的 Client 可以快速构造出符合需求的 Cache 对象，不需要过多复杂的配置，大多数情况就像构造一个 POJO 一样的简单。这里介绍两种构造 Cache 对象的方式：CacheLoader 和 Callable

CacheLoader

构造 LoadingCache 的关键在于实现 load 方法，也就是在需要访问的缓存项不存在的时候 Cache 会自动调用 load 方法将数据加载到 Cache 中。这里你肯定会想假如有多个线程过来访问这个不存在的缓存项怎么办，也就是缓存的并发问题如何怎么处理是否需要人工介入，这些在下文中也会介绍到。

除了实现 load 方法之外还可以配置缓存相关的一些性质，比如过期加载策略、刷新策略。

```
1 private static final LoadingCache<String, String> CACHE = CacheBuilder
2   .newBuilder()
3   // 最大容量为 100 超过容量有对应的淘汰机制，下文详述
4   .maximumSize(100)
5   // 缓存项写入后多久过期，下文详述
6   .expireAfterWrite(60 * 5, TimeUnit.SECONDS)
7   // 缓存写入后多久自动刷新一次，下文详述
8   .refreshAfterWrite(60, TimeUnit.SECONDS)
9   // 创建一个 CacheLoader，load 表示缓存不存在的时候加载到缓存并返回
10  .build(new CacheLoader<String, String>() {
11      // 加载缓存数据的方法
12      @Override
13      public String load(String key) {
14          return "cache [" + key + "];"
```

```
15     }  
16   });  
17  
18   public void getTest() throws Exception {  
19     CACHE.get("KEY_25487");  
20   }
```

Callable

除了在构造 Cache 对象的时候指定 load 方法来加载缓存外，我们亦可以在[获取缓存项时指定载入缓存的方法](#)，并且可以根据使用场景在不同的位置采用不同的加载方式。

比如在某些位置可以通过二级缓存加载不存在的缓存项，而有些位置则可以直接从 DB 加载缓存项。

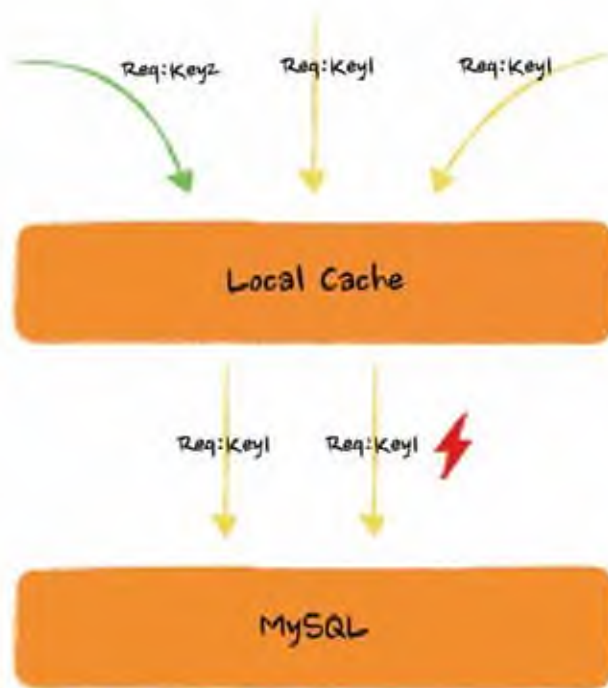
```
1  // 注意返回值是 Cache  
2  private static final Cache<String, String> SIMPLE_CACHE = CacheBuilder  
3    .newBuilder()  
4    .build();  
5  
6  public void getTest1() throws Exception {  
7    String key = "KEY_25487";  
8    // get 缓存项的时候指定 callable 加载缓存项  
9    SIMPLE_CACHE.get(key, () -> "cache [" + key + "]");  
10 }
```

缓存项加载机制

如果某个缓存过期了或者缓存项不存在于缓存中，而恰巧此此时有大量请求过来请求这个缓存项，如果没有保护机制就会导致[大量的线程同时请求数据源加载数据并生成缓存项](#)，这就是所谓的“缓存击穿”。

举个简单的例子，某个时刻有 100 个请求同时请求 KEY_25487 这个缓存项，而不巧这个缓存项刚好失效了，那么这 100 个线程（如果有这么多机器和流量的话）就会同时从 DB 加载这个数据，很可怕的点在于就算某一个线程率先获取到数据生成了缓存项，[其他的线程还是继续请求 DB 而不会走到缓存](#)。

缓存击穿



【缓存击穿图例】

看到上面这个图或许你已经有方法解决这个问题了，如果多个线程过来如果我们只让一个线程去加载数据生成缓存项，其他线程等待然后读取生成好的缓存项岂不是就完美解决。那么恭喜你在这个问题上，和 Google 工程师的思路是一致的。不过采用这个方案，问题是解了但没有完全解，后面会说到它的缺陷。

其实 Guava Cache 在 load 的时候做了并发控制，在多个线程请求一个不存在或者过期的缓存项时保证只有一个线程进入 load 方法，其他线程等待直到缓存项被生成，这样就避免了大量的线程击穿缓存直达 DB。不过试想下如果有上万 QPS 同时过来会有大量的线程阻塞导致线程无法释放，甚至会出现线程池满的尴尬场景，这也是说为什么这个方案解了“缓存击穿”问题但又没完全解。

上述机制其实就是 expireAfterWrite/expireAfterAccess 来控制的，如果你配置了过期策略对应的缓存项在过期后被访问就会走上述流程来加载缓存项。

缓存项刷新机制

缓存项的刷新和加载看起来是相似的，都是让缓存数据处于最新的状态。区别在于：

1. 缓存项加载是一个被动的过程，而缓存刷新是一个主动触发动作。如果缓存项不存在或者过期只有下次 get 的时候才会触发新值加载。而缓存刷新则更加主动替换缓存中的老值。

2. 另外一个很重要点的在于，缓存刷新的项目一定是存在缓存中的，他是对老值的替换而非是对 NULL 值的替换。

由于缓存项刷新的前提是该缓存项存在于缓存中，那么缓存的刷新就不用像缓存加载的流程一样让其他线程等待而是允许一个线程去数据源获取数据，其他线程都先返回老值直到异步线程生成了新缓存项。

这个方案完美解决了上述遇到的“缓存击穿”问题，不过他的前提是已经生成缓存项了。在实际生产情况下我们可以做缓存预热，提前生成缓存项，避免流量洪峰造成的线程堆积。

这套机制在 Guava Cache 中是通过 refreshAfterWrite 实现的，在配置刷新策略后，对应的缓存项会按照设定的时间定时刷新，避免线程阻塞的同时保证缓存项处于最新状态。

但他也不是完美的，比如他的限制是缓存项已经生成，并且如果恰巧你运气不好，大量的缓存项同时需要刷新或者过期，就会有大量的线程请求 DB，这就是常说的“缓存血崩”。

缓存项异步刷新机制

上面说到缓存项大面积失效或者刷新会导致雪崩，那么就只能限制访问 DB 的数量了，位置有三个地方：

1. 源头：因为加载缓存的线程就是前台请求线程，所以如果控制请求线程数量的确是减少大面积失效对 DB 的请求，那这样一来就不存在高并发请求，就算不用缓存都可以。

2. 中间层缓冲：因为请求线程和访问 DB 的线程是同一个，假如在中间加一层缓冲，通过一个后台线程池去异步刷新缓存所有请求线程直接返回老值，这样对于 DB 的访问的流量就可以被后台线程池的池大小控住。

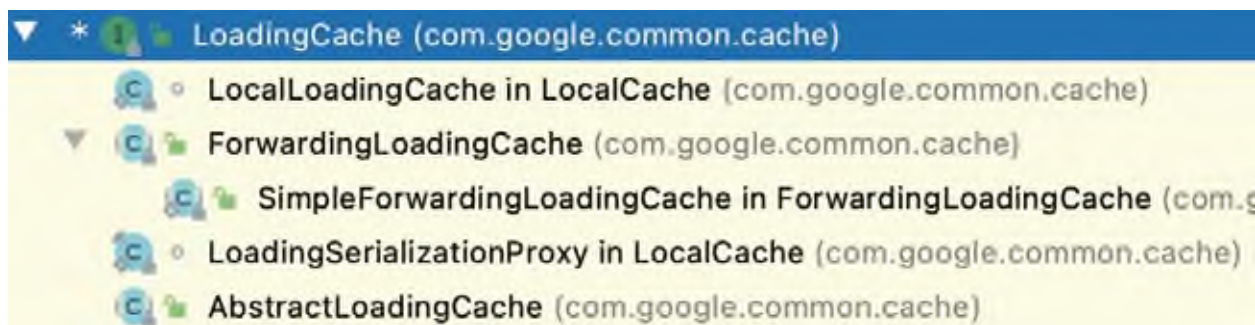
3. 底层：直接控 DB 连接池的池大小，这样访问 DB 的连接数自然就少了，但是如果大量请求到连接池发现获取不到连接程序一样会出现连接池满的问题，会有大量连接被拒绝的异常。

所以比较合适的方式是通过添加一个异步线程池异步刷新数据，在 Guava Cache 中实现方案是重写 Cache-Loader 的 reload 方法。

```
1 private static final LoadingCache<String, String> ASYNC_CACHE = CacheBuilder.newBuilder()
2   .build(
3     CacheLoader.asyncReloading(new CacheLoader<String, String>() {
4       @Override
5       public String load(String key) {
6         return key;
7       }
8
9       @Override
10      public ListenableFuture<String> reload(String key, String oldValue) throws Exception {
11        return super.reload(key, oldValue);
12      }
13    }, new ThreadPoolExecutor(5, Integer.MAX_VALUE,
14      60L, TimeUnit.SECONDS,
15      new SynchronousQueue<>()))
16 );
```

LocalCache 源码分析

先整体看下 Cache 的类结构，下面的这些子类表示了不同的创建方式本质还都是 LocalCache



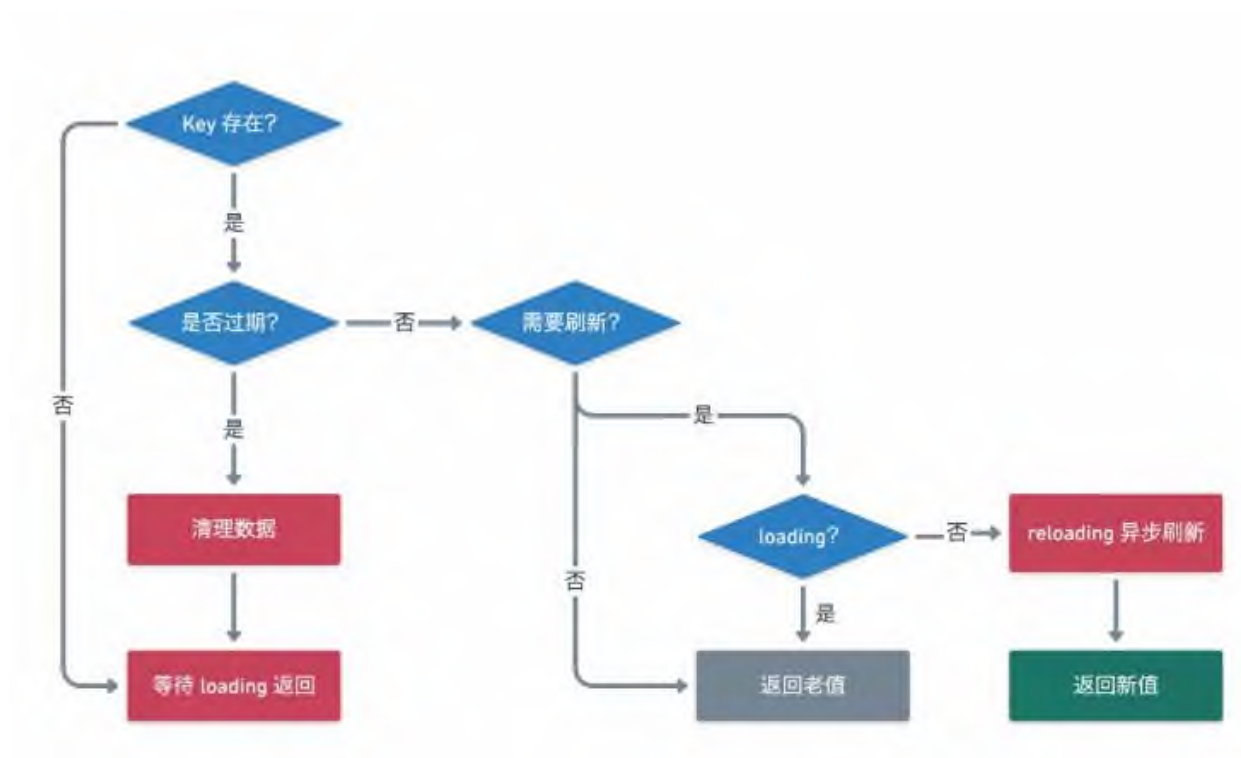
【Cache 类图】

核心代码都在 LocalCache 这个文件中，并且通过这个继承关系可以看出 Guava Cache 的本质就是 ConcurrentMap。

```
@SuppressWarnings("GoodTime") // lots of violations (nanosecond math)
@GwtCompatible(emulated = true)
class LocalCache<K, V> extends AbstractMap<K, V> implements ConcurrentMap<K, V> {
```

【LocalCache 继承与实现】

在看源码之前先理一下流程，先理清思路。如果想直接看源码理解流程可以先跳过这张图～



【get 缓存数据流程图】

这里核心理一下 Get 的流程，put 阶段比较简单就不做分析了。

LocalCache#get

```
1 V get(K key, CacheLoader<? super K, V> loader) throws ExecutionException {
2     int hash = hash(checkNotNull(key));
3     // 根据 hash 获取对应的 segment 然后从 segment 获取具体值
4     return segmentFor(hash).get(key, hash, loader);
5 }
```

Segment#get

```
1 V get(K key, int hash, CacheLoader<? super K, V> loader) throws ExecutionException {
2     checkNotNull(key);
3     checkNotNull(loader);
4     try {
5         // count 表示在这个 segment 中存活的项目个数
6         if (count != 0) {
```

```
7      // 获取 segment 中的元素 (ReferenceEntry) 包含正在 load 的数据
8      ReferenceEntry<K, V> e = getEntry(key, hash);
9      if (e != null) {
10         long now = map.ticker.read();
11         // 获取缓存值，如果是 load，invalid，expired 返回 null，同时检查是否过期了,过期移除并返回 null
12         V value = getLiveValue(e, now);
13         if (value != null) {
14             // 记录访问时间
15             recordRead(e, now);
16             // 记录缓存命中一次
17             statsCounter.recordHits(1);
18             // 刷新缓存并返回缓存值，后面展开
19             return scheduleRefresh(e, key, hash, value, now, loader);
20         }
21         ValueReference<K, V> valueReference = e.getValueReference();
22         // 如果在 loading 等着，后面展开
23         if (valueReference.isLoading()) {
24             return waitForLoadingValue(e, key, valueReference);
25         }
26     }
27 }
28
29 // 走到这说明从来没写入过值 或者 值为 null 或者 过期（数据还没做清理），后面展开
30 return lockedGetOrLoad(key, hash, loader);
31 } catch (ExecutionException ee) {
32     Throwable cause = ee.getCause();
33     if (cause instanceof Error) {
34         throw new ExecutionError((Error) cause);
35     } else if (cause instanceof RuntimeException) {
36         throw new UncheckedExecutionException(cause);
37     }
38     throw ee;
39 } finally {
40     postReadCleanup();
41 }
42 }
```

Segment#scheduleRefresh

```
1 // com.google.common.cache.LocalCache.Segment#scheduleRefresh
2
3 V scheduleRefresh(
4     ReferenceEntry<K, V> entry,
5     K key,
6     int hash,
7     V oldValue,
8     long now,
9     CacheLoader<? super K, V> loader) {
10
11     if (
12         // 配置了刷新策略 refreshAfterWrite
13         map.refreshes()
14         // 到刷新时间了
15         && (now - entry.getWriteTime() > map.refreshNanos)
16         // 没在 loading
17         && !entry.getValueReference().isLoading()) {
18         // 开始刷新，下面展开
19         V newValue = refresh(key, hash, loader, true);
20         if (newValue != null) {
21             return newValue;
22         }
23     }
24     return oldValue;
25 }
26
27
28 // com.google.common.cache.LocalCache.Segment#refresh
29
30 V refresh(K key, int hash, CacheLoader<? super K, V> loader, boolean checkTime) {
31     // 插入 loading 节点
32     final LoadingValueReference<K, V> loadingValueReference =
33         insertLoadingValueReference(key, hash, checkTime);
34
35     if (loadingValueReference == null) {
36         return null;
37     }
38
39     // 异步刷新，下面展开
40     ListenableFuture<V> result = loadAsync(key, hash, loadingValueReference, loader);
41     if (result.isDone()) {
```

```
42     try {
43         return Uninterruptibles.getUninterruptibly(result);
44     } catch (Throwable t) {
45         // don't let refresh exceptions propagate; error was already logged
46     }
47 }
48 return null;
49 }
50
51 // com.google.common.cache.LocalCache.Segment#loadAsync
52
53 ListenableFuture<V> loadAsync(
54     final K key,
55     final int hash,
56     final LoadingValueReference<K, V> loadingValueReference,
57     CacheLoader<? super K, V> loader) {
58     // 通过 loader 异步加载数据，下面展开
59     final ListenableFuture<V> loadingFuture = loadingValueReference.loadFuture(key, loader);
60     loadingFuture.addListener(
61         new Runnable() {
62             @Override
63             public void run() {
64                 try {
65                     getAndRecordStats(key, hash, loadingValueReference, loadingFuture);
66                 } catch (Throwable t) {
67                     logger.log(Level.WARNING, "Exception thrown during refresh", t);
68                     loadingValueReference.setException(t);
69                 }
70             }
71         },
72         directExecutor());
73     return loadingFuture;
74 }
75
76 // com.google.common.cache.LocalCache.LoadingValueReference#loadFuture
77
78 public ListenableFuture<V> loadFuture(K key, CacheLoader<? super K, V> loader) {
79     try {
80         stopwatch.start();
81         // oldValue 指在写入 loading 节点前这个位置的值，如果这个位置之前没有值 oldValue 会被赋值为 UNSET
82         // UNSET.get() 值为 null，所以这个缓存项从来没有进入缓存需要同步 load 具体原因前面提到了，如果通过
83         // 异步 reload，由于没有老值会导致其他线程返回的都是 null
84         V previousValue = oldValue.get();
```

```
85     if (previousValue == null) {
86         V newValue = loader.load(key);
87         return set(newValue) ? futureValue : Futures.immediateFuture(newValue);
88     }
89     // 异步 load
90     ListenableFuture<V> newValue = loader.reload(key, previousValue);
91     if (newValue == null) {
92         return Futures.immediateFuture(null);
93     }
94     // To avoid a race, make sure the refreshed value is set into loadingValueReference
95     // *before* returning newValue from the cache query.
96     return transform(
97         newValue,
98         new com.google.common.base.Function<V, V>() {
99             @Override
100             public V apply(V newValue) {
101                 LoadingValueReference.this.set(newValue);
102                 return newValue;
103             }
104         },
105         directExecutor());
106 } catch (Throwable t) {
107     ListenableFuture<V> result = setException(t) ? futureValue : fullyFailedFuture(t);
108     if (t instanceof InterruptedException) {
109         Thread.currentThread().interrupt();
110     }
111     return result;
112 }
113 }
```

Segment#waitForLoadingValue

```
1  V waitForLoadingValue(ReferenceEntry<K, V> e, K key, ValueReference<K, V> valueReference)
2      throws ExecutionException {
3      // 首先你要是一个 loading 节点
4      if (!valueReference.isLoading()) {
5          throw new AssertionError();
6      }
7
8      checkState(!Thread.holdsLock(e), "Recursive load of: %s", key);
9      // don't consider expiration as we're concurrent with loading
10     try {
```

```
11     V value = valueReference.waitForValue();
12     if (value == null) {
13         throw new InvalidCacheLoadException("CacheLoader returned null for key " + key + ".");
14     }
15     // re-read ticker now that loading has completed
16     long now = map.ticker.read();
17     recordRead(e, now);
18     return value;
19 } finally {
20     statsCounter.recordMisses(1);
21 }
22 }
23
24 // com.google.common.cache.LocalCache.LoadingValueReference#waitForValue
25
26 public V waitForValue() throws ExecutionException {
27     return getUninterruptibly(futureValue);
28 }
29
30 // com.google.common.util.concurrent.Uninterruptibles#getUninterruptibly
31
32 public static <V> V getUninterruptibly(Future<V> future) throws ExecutionException {
33     boolean interrupted = false;
34     try {
35         while (true) {
36             try {
37                 // hang 住，如果该线程被打断了继续回去 hang 住等结果，直到有结果返回
38                 return future.get();
39             } catch (InterruptedException e) {
40                 interrupted = true;
41             }
42         }
43     } finally {
44         if (interrupted) {
45             Thread.currentThread().interrupt();
46         }
47     }
48 }
```

Segment#lockedGetOrLoad

```
1 V lockedGetOrLoad(K key, int hash, CacheLoader<? super K, V> loader) throws ExecutionException {
2     ReferenceEntry<K, V> e;
3     ValueReference<K, V> valueReference = null;
4     LoadingValueReference<K, V> loadingValueReference = null;
5     boolean createNewEntry = true;
6
7     // 要对 segment 写操作，先加锁
8     lock();
9     try {
10         // re-read ticker once inside the lock
11         long now = map.ticker.read();
12         preWriteCleanup(now);
13
14         // 这里基本就是 HashMap 的代码，如果没有 segment 的数组下标冲突了就拉一个链表
15         int newCount = this.count - 1;
16         AtomicReferenceArray<ReferenceEntry<K, V>> table = this.table;
17         int index = hash & (table.length() - 1);
18         ReferenceEntry<K, V> first = table.get(index);
19
20         for (e = first; e != null; e = e.getNext()) {
21             K entryKey = e.getKey();
22             if (e.getHash() == hash
23                 && entryKey != null
24                 && map.keyEquivalence.equivalent(key, entryKey)) {
25                 valueReference = e.getValueReference();
26
27                 // 如果在加载中 不做任何处理
28                 if (valueReference.isLoading()) {
29                     createNewEntry = false;
30                 } else {
31                     V value = valueReference.get();
32                     // 如果缓存项为 null 数据已经被删除，通知对应的 queue
33                     if (value == null) {
34                         enqueueNotification(
35                             entryKey, hash, value, valueReference.getWeight(), RemovalCause.COLLECTED);
36                     // 这个是 double check 如果缓存项过期 数据没被删除，通知对应的 queue
37                     } else if (map.isExpired(e, now)) {
38                         // This is a duplicate check, as preWriteCleanup already purged expired
39                         // entries, but let's accommodate an incorrect expiration queue.
40                         enqueueNotification(
41                             entryKey, hash, value, valueReference.getWeight(), RemovalCause.EXPIRED);
```

```
42         // 再次看到的时候这个位置有值了直接返回
43     } else {
44         recordLockedRead(e, now);
45         statsCounter.recordHits(1);
46         return value;
47     }
48
49     // immediately reuse invalid entries
50     writeQueue.remove(e);
51     accessQueue.remove(e);
52     this.count = newCount; // write-volatile
53 }
54 break;
55 }
56 }
57
58 // 没有 loading , 创建一个 loading 节点
59 if (createNewEntry) {
60     loadingValueReference = new LoadingValueReference<>();
61
62     if (e == null) {
63         e = newEntry(key, hash, first);
64         e.setValueReference(loadingValueReference);
65         table.set(index, e);
66     } else {
67         e.setValueReference(loadingValueReference);
68     }
69 }
70 } finally {
71     unlock();
72     postWriteCleanup();
73 }
74
75 if (createNewEntry) {
76     try {
77         // Synchronizes on the entry to allow failing fast when a recursive load is
78         // detected. This may be circumvented when an entry is copied, but will fail fast most
79         // of the time.
80         synchronized (e) {
81             return loadSync(key, hash, loadingValueReference, loader);
82         }
83     } finally {
84         statsCounter.recordMisses(1);
```



```
85     }  
86   } else {  
87     // The entry already exists. Wait for loading.  
88     return waitForLoadingValue(e, key, valueReference);  
89   }  
90 }
```

总结

结合上面图以及源码我们发现整个流程中 GuavaCache 是**没有额外的线程去做数据清理和刷新的**，基本都是通过 **Get 方法来触发这些动作**，减少了设计的复杂性和降低了系统开销。

简单回顾下 Get 的流程以及在每个阶段做的事情，返回的值。**首先判断缓存是否过期然后判断是否需要刷新**，如果过期了就调用 **loading 去同步加载数据（其他线程阻塞）**，如果是仅仅需要刷新调用 **reloading 异步加载（其他线程返回老值）**。

所以如果 $\text{refreshTime} > \text{expireTime}$ 意味着永远走不到缓存刷新逻辑，缓存刷新是为了在缓存有效期内尽量保证缓存数据一致性所以在配置刷新策略和过期策略时一定要保证 $\text{refreshTime} < \text{expireTime}$ 。

最后关于 Guava Cache 的使用建议 (最佳实践)：

- 1.如果刷新时间配置的较短一定要重载 reload 异步加载数据的方法，传入一个自定义线程池保护 DB
- 2.失效时间一定要大于刷新时间
- 3.如果是常驻内存的一些少量数据失效时间可以配置的较长刷新时间配置短一点 (根据业务对缓存失效容忍度)

• 一种低延迟的超时中心实现方式

- 作者：默达
- 出品：淘系技术

高延迟方案与低延迟方案的区别。

背景

在很多产品中都存在生命周期相关的设计，时间节点到了之后需要做对应的事情。

超时中心（TimeOutCenter，TOC）负责存储和调度生命周期节点上面的超时任务，当超时任务设置的超时时间到期后，超时中心需要立即调度处理这些超时任务。对于一些需要低延迟的超时场景，超时中心调度延迟会给产品带来不可估量的影响。

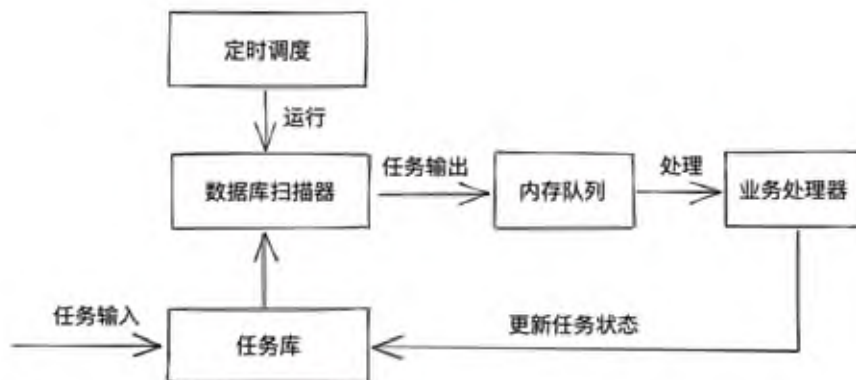
因此本文提出一种低延迟的超时中心实现方式，首先介绍传统的超时中心的实现方案，以及传统方案中的缺点，然后介绍低延迟的方案，说明如何解决传统方案中的延迟问题。

传统高延迟方案

整体框架

传统的超时中心整体框架如下所示，任务输入后存储在超时任务库中，定时器触发运行数据库扫描器，数据库扫描器从超时任务库中扫描已经到达超时时间的任务，已经到达超时时间的任务存储在机器的内存队列中，等待交给业务处理器进行处理，业务处理器处理完成后更新任务状态。

在大数据时代，超时任务数量肯定是很大的，传统的超时中心通过分库分表支持存储海量的超时任务，定时器触发也需要做相应的改变，需要充分利用集群的能力，下面分别从超时任务库和定时器触发两方面详细介绍。



任务库计划

任务库数据模型如下所示，采用分库分表存储，一般可设计为8个库1024个表，具体可以根据业务需求调整。biz_id为分表键，job_id为全局唯一的任务ID，status为超时任务的状态，action_time为任务的执行时间，attribute存储额外的数据。

只有当action_time小于当前时间且status为待处理时，任务才能被扫描器加载到内存队列。任务被处理完成后，任务的状态被更新成已处理。

1	job_id	bigint unsigned	超时任务的ID，全局唯一
2	gmt_create	datetime	创建时间
3	gmt_modified	datetime	修改时间
4	biz_id	bigint unsigned	业务id，一般为关联的主订单或子订单id
5	biz_type	bigint unsigned	业务类型
6	status	tinyint	超时任务状态（0待处理，2已处理，3取消）
7	action_time	datetime	超时任务执行时间
8	attribute	varchar	额外数据

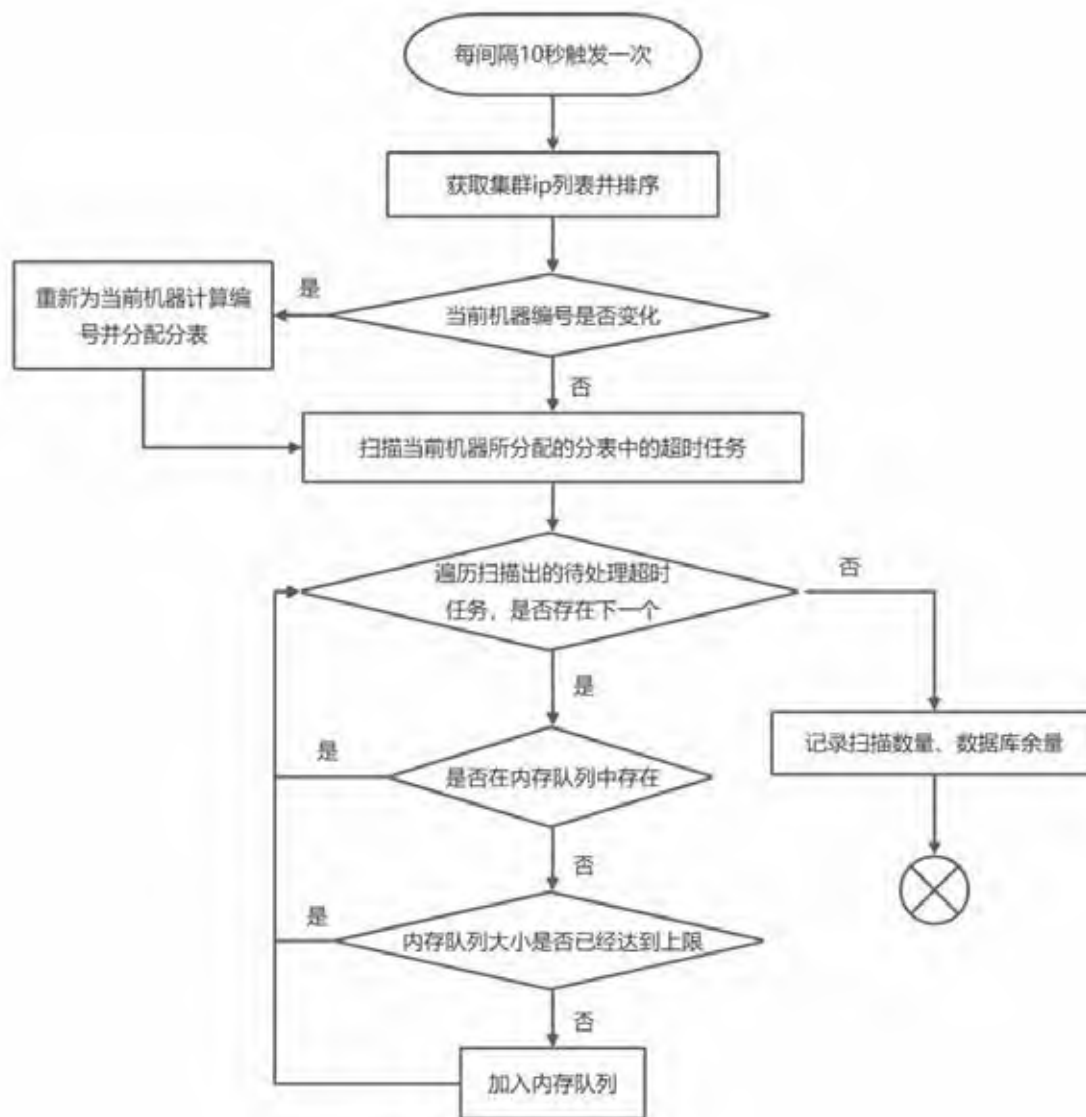
定时调度设计

定时调度流程图如下所示，定时器每间隔10秒触发一次调度，从集群configserver中获取集群ip列表并为当前机器编号，然后给所有ip分配表。

分配表时需要考虑好几件事：一张表只属于一台机器，不会出现重复扫描；机器上线下线需要重新分配表。

当前机器从所分配的表中扫描出所有状态为待处理的超时任务，遍历扫描出的待处理超时任务。

对于每个超时任务，当内存队列不存在该任务且内存队列未滿时，超时任务才加入内存队列，否则循环检查等待。



缺点

- 需要定时器定时调度，定时器调度间隔时间加长了超时任务处理的延迟时间；
- 数据库扫描器为避免重复扫描数据，一张表只能属于一台机器，任务库分表的数量就是任务处理的并发度，并发度受限制；
- 当单表数据量庞大时，即使从单张表中扫描所有待处理的超时任务也需要花费很长的时间；
- 本方案总体处理步骤为：先扫描出所有超时任务，再对单个超时任务进行处理；超时任务处理延迟时间需要加上超时任务扫描时间；

- 本方案处理超时任务的最小延迟为定时器的定时间隔时间，在任务数量庞大的情况下，本方案可能存在较大延迟。

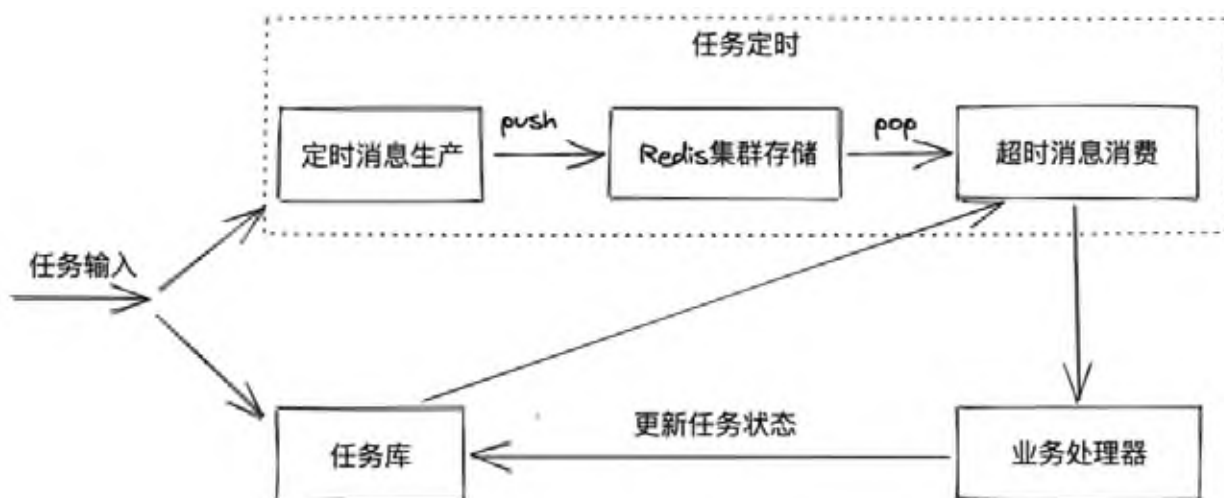
低延迟方案

任务输入后分为两个步骤。

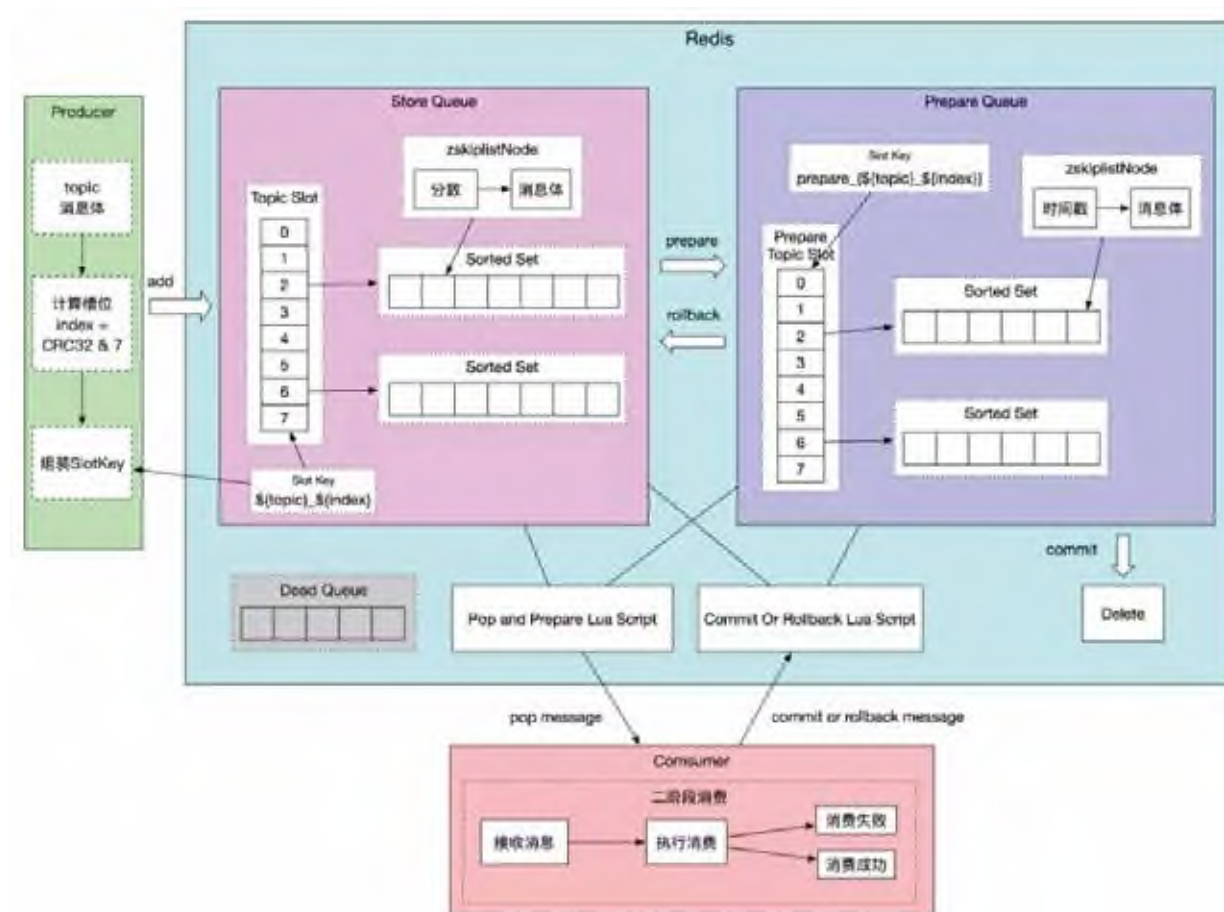
第一个步骤是将任务存储到任务库，本方案的任务库模型设计和上面方案中的任务库模型设计一样；

第二步是任务定时，将任务的jobId和actionTime以一定方式设置到Redis集群中，当定时任务的超时时间到了之后，从Redis集群pop超时任务的jobId，根据jobId从任务库中查询详细的任务信息交给业务处理器进行处理，最后更新任务库中任务的状态。

本方案与上述方案最大的不同点就是超时任务的获取部分，上述方案采用定时调度扫描任务库，本方案采用基于Redis的任务定时系统，接下来将具体讲解任务定时的设计。



Redis存储设计



Topic的设计

Topic的定义有三部分组成，topic表示主题名称，slotAmount表示消息存储划分的槽数量，topicType表示消息的类型。

主题名称是一个Topic的唯一标示，相同主题名称Topic的slotAmount和topicType一定是一样的。

消息存储采用Redis的Sorted Set结构，为了支持大量消息的堆积，需要把消息分散存储到很多个槽中，slotAmount表示该Topic消息存储共使用的槽数量，槽数量一定需要是2的n次幂。

在消息存储的时候，采用对指定数据或者消息体哈希求余得到槽位置。

StoreQueue的设计

上图中topic划分了8个槽位，编号0-7。

计算消息体对应的CRC32值，CRC32值对槽数量进行取模得到槽序号，SlotKey设计为#{topic}_#{index}（也即Redis的键），其中#{ }表示占位符。

StoreQueue结构采用Redis的Sorted Set，Redis的Sorted Set中的数据按照分数排序，实现定时消息的关键就在于如何利用分数、如何添加消息到Sorted Set、如何从Sorted Set中弹出消息。

定时消息将时间戳作为分数，消费时每次弹出分数小于当前时间戳的一个消息。

PrepareQueue的设计

为了保障每条消息至少消费一次，消费者不是直接pop有序集合中的元素，而是将元素从StoreQueue移动到PrepareQueue并返回消息给消费者，等消费成功后再从PrepareQueue删除，或者消费失败后从PrepareQueue重新移动到StoreQueue，这便是根据二阶段提交的思想实现的二阶段消费。

在后面将会详细介绍二阶段消费的实现思路，这里重点介绍下PrepareQueue的存储设计。

StoreQueue中每一个Slot对应PrepareQueue中的Slot，PrepareQueue的SlotKey设计为prepare_#{topic}#{index}。

PrepareQueue采用Sorted Set作为存储，消息移动到PrepareQueue时刻对应的（秒级时间戳*1000+重试次数）作为分数，字符串存储的是消息体内容。

这里分数的设计与重试次数的设计密切相关，所以在重试次数设计章节详细介绍。

PrepareQueue的SlotKey设计中需要注意的一点，由于消息从StoreQueue移动到PrepareQueue是通过Lua脚本操作的，因此需要保证Lua脚本操作的Slot在同一个Redis节点上，如何保证PrepareQueue的SlotKey和对应的StoreQueue的SlotKey被hash到同一个Redis槽中呢。

Redis的hash tag功能可以指定SlotKey中只有某一部分参与计算hash，这一部分采用{}包括，因此PrepareQueue的SlotKey中采用{}包括了StoreQueue的SlotKey。

DeadQueue的设计

消息重试消费16次后，消息将进入DeadQueue。

DeadQueue的SlotKey设计为prepare#{topic}#{index}，这里同样采用hash tag功能保证DeadQueue的SlotKey与对应StoreQueue的SlotKey存储在同一个Redis节点。

定时消息生产

生产者的任务就是将消息添加到StoreQueue中。

首先，需要计算出消息添加到Redis的SlotKey，如果发送方指定了消息的slotBasis（否则采用content代替），则计算slotBasis的CRC32值，CRC32值对槽数量进行取模得到槽序号，SlotKey设计为#{topic}_#{index}，其中#{表示占位符。

发送定时消息时需要设置actionTime，actionTime必须大于当前时间，表示消费时间戳，当前时间大于该消费时间戳的时候，消息才会被消费。

因此在存储该类型消息的时候，采用actionTime作为分数，采用命令zadd添加到Redis。

超时消息消费

每台机器将启动多个Woker进行超时消息消费，Woker即表示线程，定时消息被存储到Redis的多个Slot中，因此需要zookeeper维护集群中Woker与slot的关系，一个Slot只分配给一个Woker进行消费，一个Woker可以消费多个Slot。

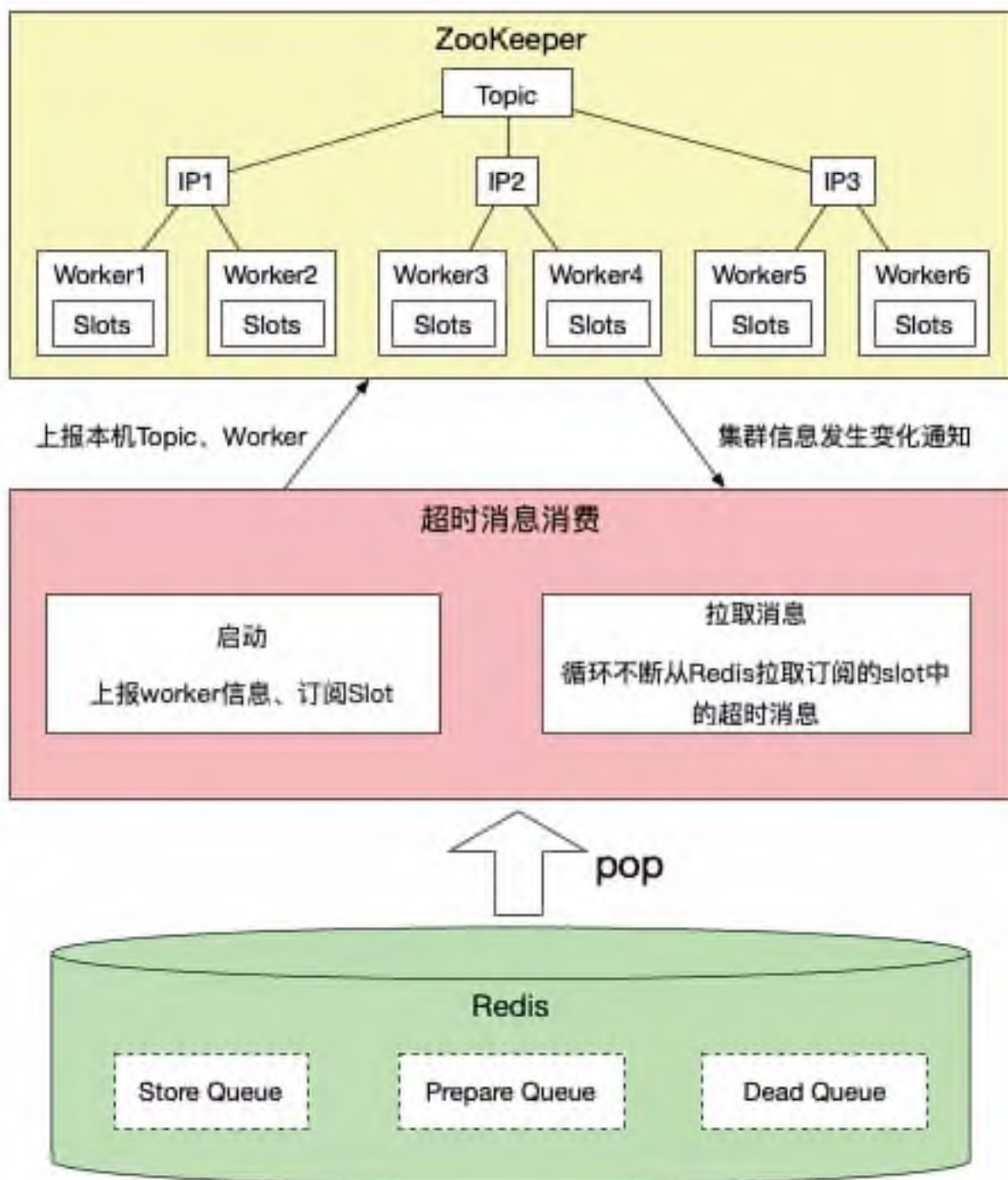
Woker与Slot的关系在每台机器启动与停止时重新分配，超时消息消费集群监听了zookeeper节点的变化。

Woker与Slot关系确定后，Woker则循环不断地从Redis拉取订阅的Slot中的超时消息。

在StoreQueue存储设计中说明了定时消息存储时采用Sorted Set结构，采用定时时间actionTime作为分数，因此定时消息按照时间大小存储在Sorted Set中。

因此在拉取超时消息进行只需采用Redis命令ZRANGEBYSCORE弹出分数小于当前时间戳的一条消息。

为了保证系统的可用性，还需要考虑保证定时消息至少被消费一次以及消费的重试次数，下面将具体介绍如何保证至少消费一次和消费重试次数控制。



至少消费一次

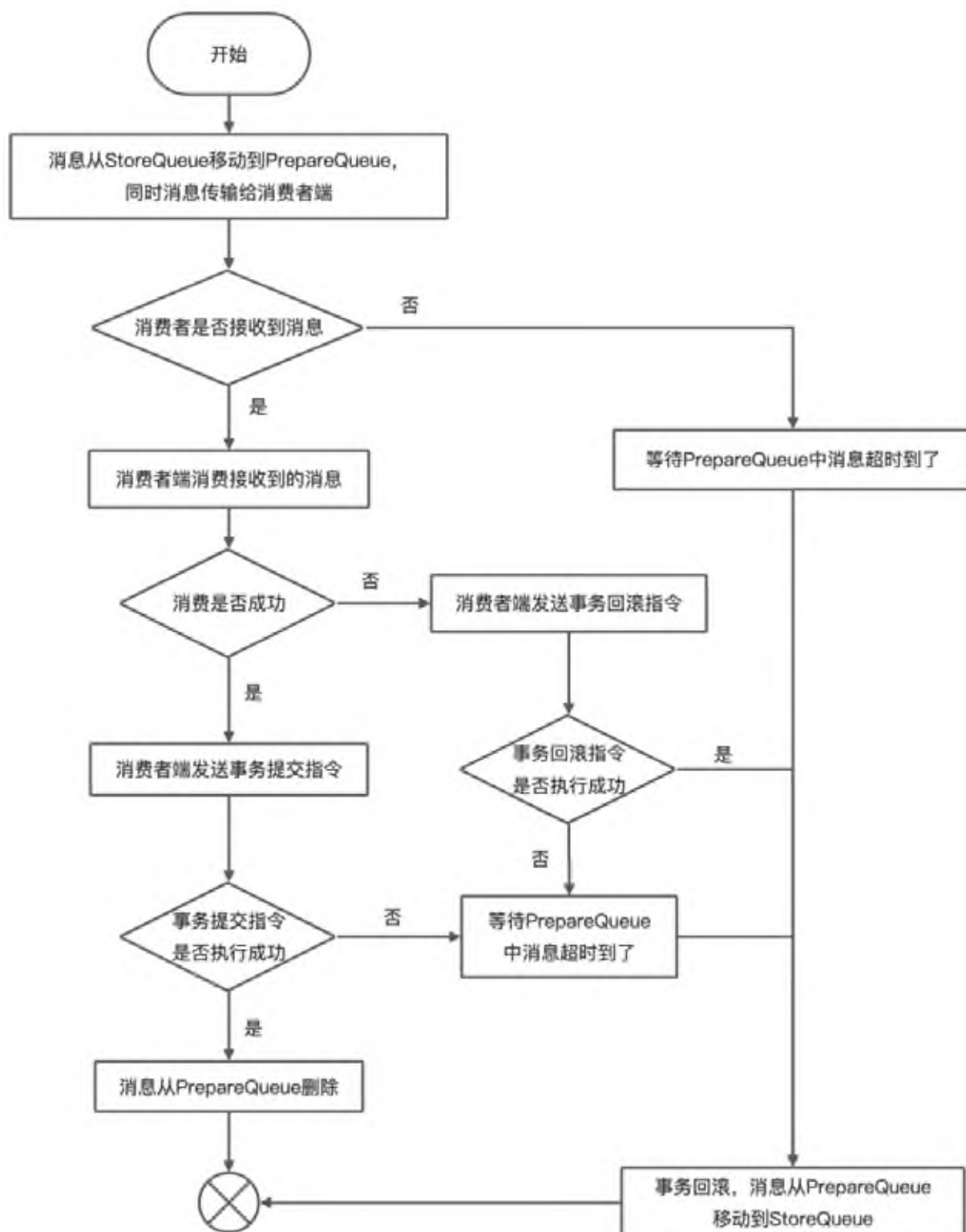
至少消费一次的问题比较类似银行转账问题，A向B账户转账100元，如何保障A账户扣减100同时B账户增加100，因此我们可以想到二阶段提交的思想。

第一个准备阶段，A、B分别进行资源冻结并持久化undo和redo日志，A、B分别告诉协调者已经准备好；第二个提交阶段，协调者告诉A、B进行提交，A、B分别提交事务。

本方案基于二阶段提交的思想来实现至少消费一次。

Redis存储设计中PrepareQueue的作用就是用来冻结资源并记录事务日志，消费者端即是参与者也是协调者。

第一个准备阶段，消费者端通过执行Lua脚本从StoreQueue中Pop消息并存储到PrepareQueue，同时消息传输到消费者端，消费者端消费该消息；第二个提交阶段，消费者端根据消费结果是否成功协调消息队列服务是提交还是回滚，如果消费成功则提交事务，该消息从PrepareQueue中删除，如果消费失败则回滚事务，消费者端将该消息从PrepareQueue移动到StoreQueue，如果因为各种异常导致PrepareQueue中消息滞留超



消费重试次数控制

采用二阶段消费方式，需要将消息在StoreQueue和PrepareQueue之间移动，如何实现重试次数控制呢，其关键在StoreQueue和PrepareQueue的分数设计。

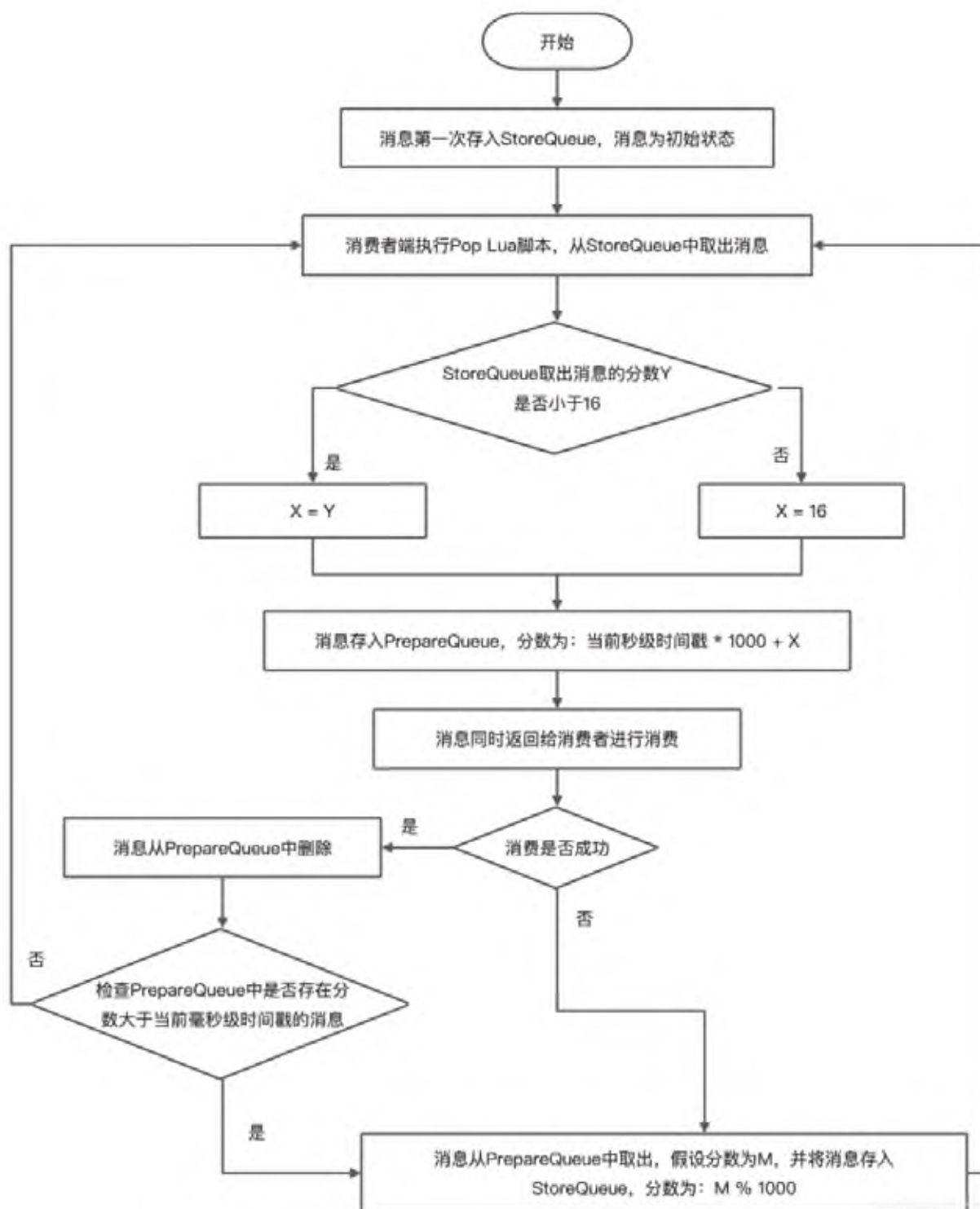
PrepareQueue的分数需要与时间相关，正常情况下，消费者不管消费失败还是消费成功，都会从Prepare-Queue删除消息，当消费者系统发生异常或者宕机的时候，消息就无法从PrepareQueue中删除，我们也不知道消费者是否消费成功，为保障消息至少被消费一次，我们需要做到超时回滚，因此分数需要与消费时间相关。

当PrepareQueue中的消息发生超时的时候，将消息从PrepareQueue移动到StoreQueue。

因此PrepareQueue的分数设计为：秒级时间戳*1000+重试次数。

定时消息首次存储到StoreQueue中的分数表示消费时间戳，如果消息消费失败，消息从PrepareQueue回滚到StoreQueue，定时消息存储时的分数都表示剩余重试次数，剩余重试次数从16次不断降低最后为0，消息进入死信队列。

消息在StoreQueue和PrepareQueue之间移动流程如下：



优点

- 消费低延迟：采用基于Redis的定时方案直接从Redis中pop超时任务，避免扫描任务库，大大减少了延迟时间。
- 可控并发度：并发度取决于消息存储的Slot数量以及集群Worker数量，这两个数量都可以根据业务需要进行调控，传统方案中并发度为分库分表的数量。
- 高性能：Redis单机的QPS可以达到10w，Redis集群的QPS可以达到更高的水平，本方案没有复杂查询，消费过程中从Redis拉取超时消息的时间复杂度为 $O(1)$ 。
- 高可用：至少消费一次保障了定时消息一定被消费，重试次数控制保证消费不被阻塞。

• Java类加载器 — classloader 的原理及应用

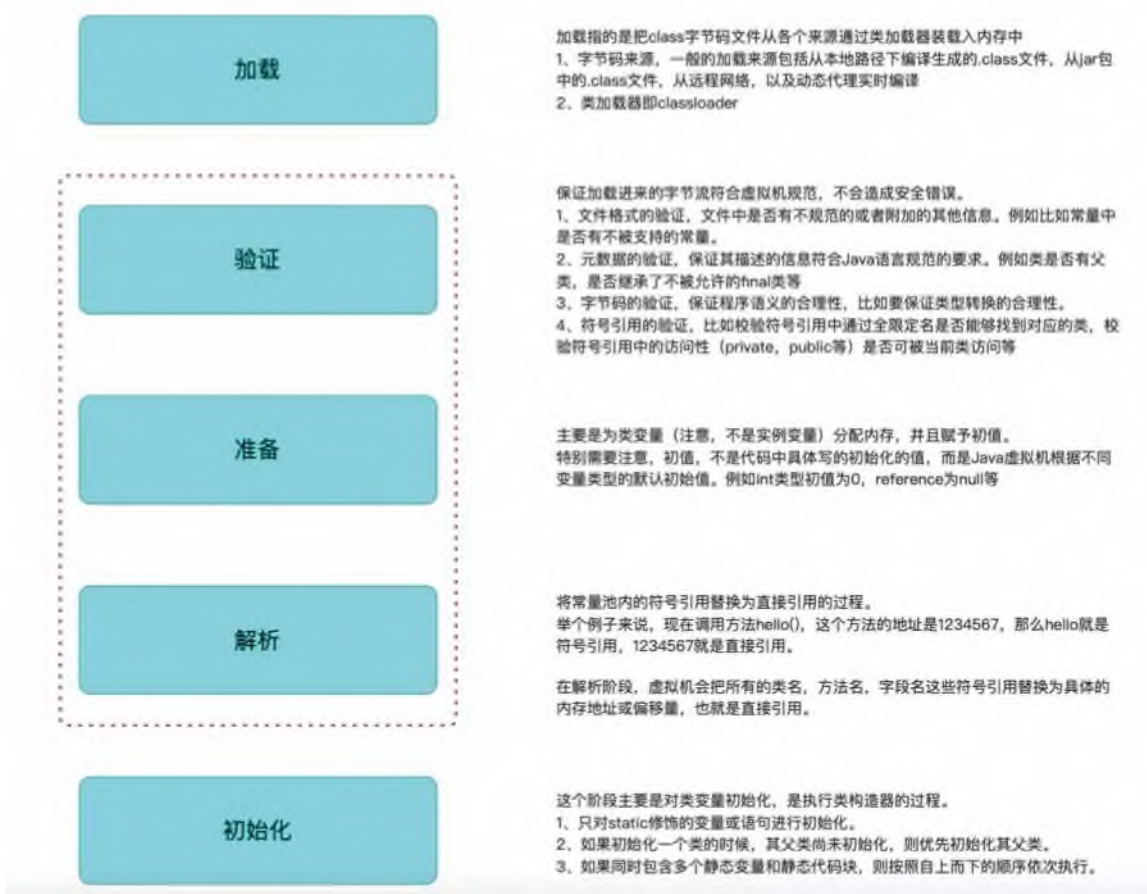
- 作者：金雅博（行泽）
- 出品：淘系技术

什么是classloader

classloader顾名思义，即是类加载。虚拟机把描述类的数据从class字节码文件加载到内存，并对数据进行检验、转换解析和初始化，最终形成可以被虚拟机直接使用的Java类型，这就是虚拟机的类加载机制。了解java的类加载机制，可以快速解决运行时的各种加载问题并快速定位其背后的本质原因，也是解决疑难杂症的利器。因此学好类加载原理也至关重要。

classloader的加载过程

类从被加载到虚拟机内存到被卸载，整个完整的生命周期包括：类加载、验证、准备、解析、初始化和使用和卸载七个阶段。其中验证，准备，解析三个部分统称为连接。接下来我们可以详细了解下类加载的各个过程。

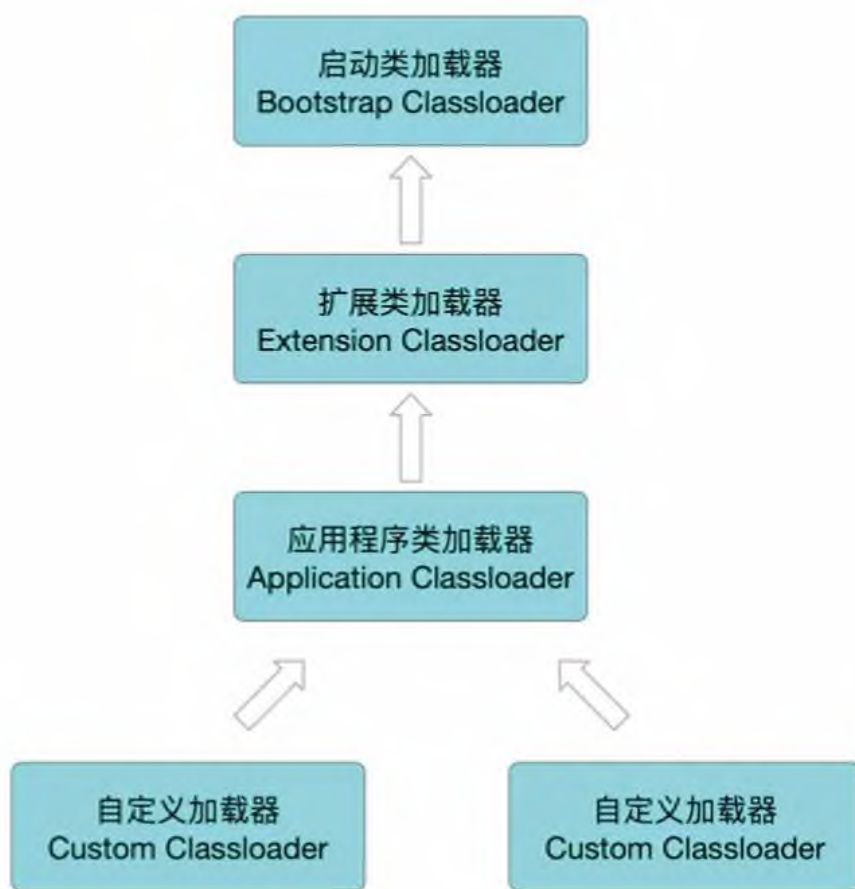


classloader的整个加载过程还是非常复杂的，具体的细节可以参考《深入理解java虚拟机》进行深入了解。为了方便记忆，我们可以使用一句话来表达其加载的整个过程，“家宴准备了西式菜”，即家(加载)宴(验证)准备(准备)了西(解析)式(初始化)菜。保证你以后能够很快的想起来。

虽然classloader的加载过程有复杂的5步，但事实上除了加载之外的四步，其它都是由JVM虚拟机控制的，我们除了适应它的规范进行开发外，能够干预的空间并不多。而加载则是我们控制classloader实现特殊目的最重要的手段了。也是接下来我们介绍的重点了。

classloader双亲委托机制

classloader的双亲委托机制是指多个类加载器之间存在父子关系的时候，某个class类具体由哪个加载器进行加载的问题。其具体的过程表现为：当一个类加载的过程中，它首先不会去加载，而是委托给自己的父类去加载，父类又委托给自己的父类。因此所有的类加载都会委托给顶层的父类，即Bootstrap Classloader进行加载，然后父类自己无法完成这个加载请求，子加载器才会尝试自己去加载。使用双亲委派模型，Java类随着它的加载器一起具备了一种带有优先级的层次关系，通过这种层次模型，可以避免类的重复加载，也可以避免核心类被不同的类加载器加载到内存中造成冲突和混乱，从而保证了Java核心库的安全。



整个java虚拟机的类加载层次关系如上图所示，启动类加载器(Bootstrap Classloader)负责将<JAVA_HOME>/lib目录下并且被虚拟机识别的类库加载到虚拟机内存中。我们常用基础库，例如java.util.**，java.io.**，java.lang.**等等都是由根加载器加载。

扩展类加载器(Extention Classloader)负责加载JVM扩展类，比如swing系列、内置的js引擎、xml解析器等，这些类库以javax开头，它们的jar包位于<JAVA_HOME>/lib/ext目录中。

应用程序加载器(Application Classloader)也叫系统类加载器，它负责加载用户路径(ClassPath)上所指定的类库。我们自己编写的代码以及使用的第三方的jar包都是由它来加载的。

自定义加载器(Custom Classloader)通常是我们为了某些特殊目的实现的自定义加载器，后面我们得会详细介绍到它的作用以及使用场景。

双亲委托机制看起来比较复杂，但是其本身的核心代码逻辑却是非常的清晰简单，我们着重抽取了类加载的双亲委托的核心代码如下，不过二十行左右。

```
protected Class<?> loadClass(String name, boolean resolve) throws ClassNotFoundException {
    synchronized (this.getClassLoadingLock(name)) {
        //缓存中是否已经存在
        Class<?> c = this.findLoadedClass(name);
        if (c == null) {
            long t0 = System.nanoTime();
            try {
                //如果parent classloader存在，则委托给父类进行加载
                if (this.parent != null) {
                    c = this.parent.loadClass(name, false);
                } else {
                    //根Bootstrap Classloader加载 ExtensionClassloader的parent为null
                    c = this.findBootstrapClassOrNull(name);
                }
            } catch (ClassNotFoundException var10) {
            }
            if (c == null) {
                long t1 = System.nanoTime();
                //自己加载
                c = this.findClass(name);
                PerfCounter.getParentDelegationTime().addTime(t1 - t0);
                PerfCounter.getFindClassTime().addElapsedTimeFrom(t1);
                PerfCounter.getFindClasses().increment();
            }
        }
        //是否进行解析
        if (resolve) {
            this.resolveClass(c);
        }
        return c;
    }
}
```

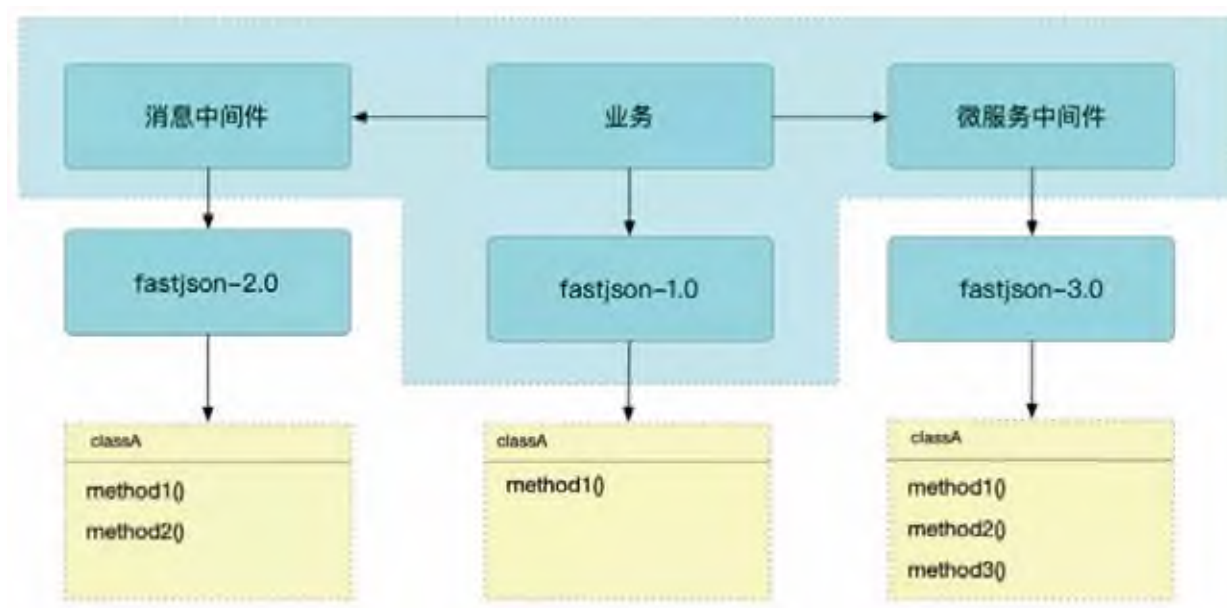
classloader的应用场景

类加载器是java语言的一项创新，也是java语言流行的重要原因这一。通过灵活定义classloader的加载机制，我们可以完成很多事情，例如解决类冲突问题，实现热加载以及热部署，甚至可以实现jar包的加密保护。接下来，我们会针对这些特殊场景进行逐一介绍。

依赖冲突

做过多人协同开发的大型项目的同学可能深有感触。基于maven的pom进制可以方便的进行依赖管理，但是由于maven依赖的传递性，会导致我们的依赖错综复杂，这样就会导致引入类冲突的问题。最典型的就是NoSuchMethodError错误。

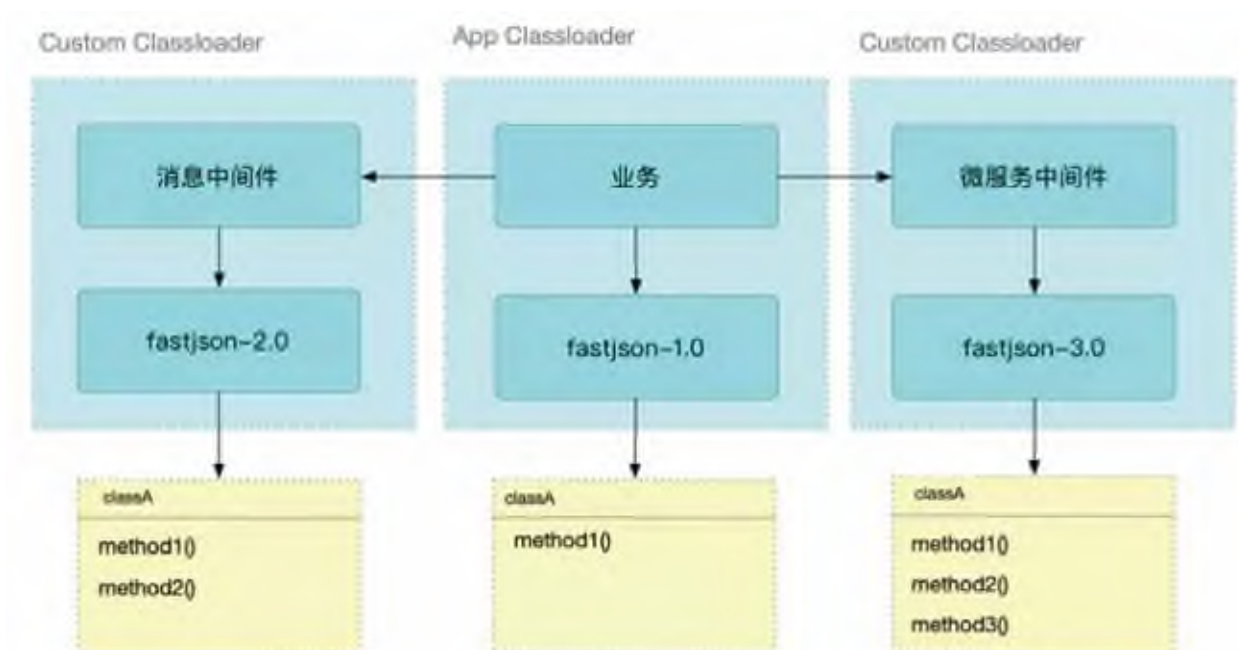
在阿里平时的项目开发中是否也会遇到类似的问题吗，答案是肯定的。例如阿里内部也很多成熟的中间件，由不同的中间件团队来负责。那么当一个项目引入不同的中间件的时候，该如何避免依赖冲突的问题呢？首先我们用一个非常简单的场景来描述为什么会出现类冲突的问题。



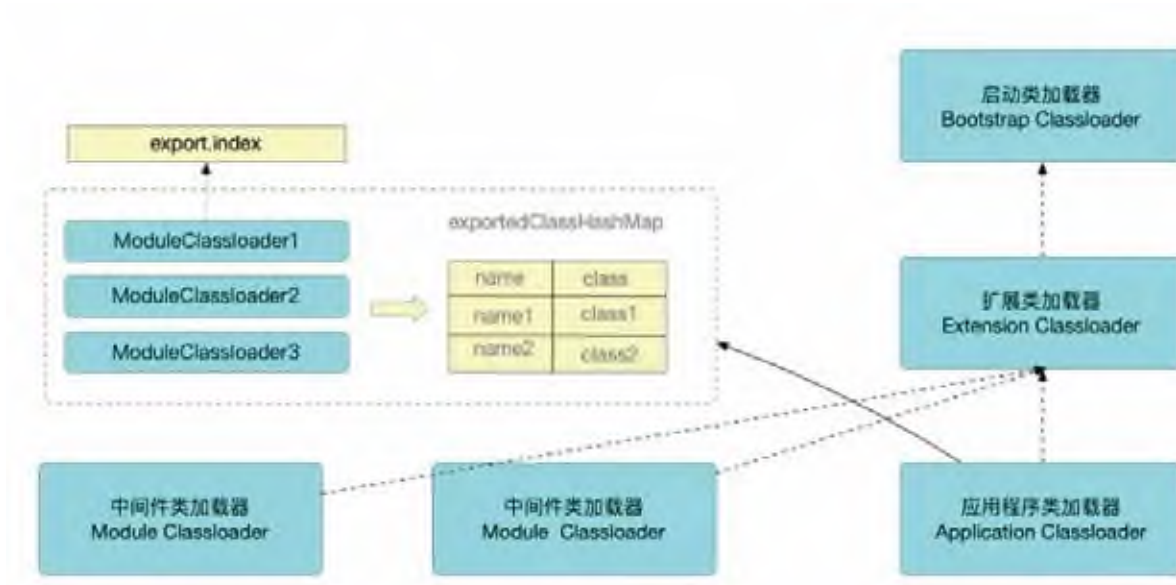
某个业务引用了消息中间件(例如metaq)和微服务中间件(例如dubbo)，这两个中间件也同时引用了fastjson-2.0和fastjson-3.0版本，而业务自己本身也引用了fastjson-1.0版本。这三个版本表现不同之处在于classA类中方法数目不相同，我们根据maven依赖处理的机制，引用路径最短的fastjson-1.0会真正作为应用最终的依赖，其它两个版本的fastjson则会被忽略，那么中间件在调用method2()方法的时候，则会抛出方法找不到异常。或许你会说，将所有依赖fastjson的版本都升级到3.0不是就能解解决问题吗？确实这样能够解决问题，但是在实际操作中不太现实，首先，中间件团队和业务团队之间并不是一个团队，并不能做到高效协同，

其次是中间件的稳定性是需要保障的，不可能因为包冲突问题，就升级版本，更何况一个中间件依赖的包可能有上百个，如果纯粹依赖包升级来解决，不仅稳定性难以保障，排包耗费的时间恐怕就让人窒息了。

那如何解决包冲突的问题呢？答案就是pandora(潘多拉)，通过自定义类加载器，为每个中间件自定义一个加载器，这些加载器之间的关系是平行的，彼此没有依赖关系。这样每个中间件的classloader就可以加载各自版本的fastjson。因为一个类的全限定名以及加载该类的加载器两者共同形成了这个类在JVM中的惟一标识，这也是阿里pandora实现依赖隔离的基础。



可能到这里，你又会有新的疑惑，根据双亲委托模型，App Classloader分别继承了Custom Classloader.那么业务包中的fastjson的class在加载的时候，会先委托到Custom ClassLoader。这样不就会导致自身依赖的fastjson版本被忽略吗？确实如此，所以潘多拉又是如何做的呢？



首先每个中间件对应的ModuleClassLoader在加载中间对应的class文件的同时，根据中间件配置的export.index负责将要需要透出的class(主要是中间件api接口的相关类)索引到exportedClassHashMap中，然后应用程序的类加载器会持有这个exportedClassHashMap，因此应用程序代码在loadClass的时候，会优先判断exportedClassHashMap是否存在当前类，如果存在，则直接返回，如果不存在，则再使用传统的双亲委托机制来进行类加载。这样中间件ModuleClassLoader不仅实现了中间件的加载，也实现了中间件关键服务类的透出。

我们可以大概看下应用程序类加载的过程：

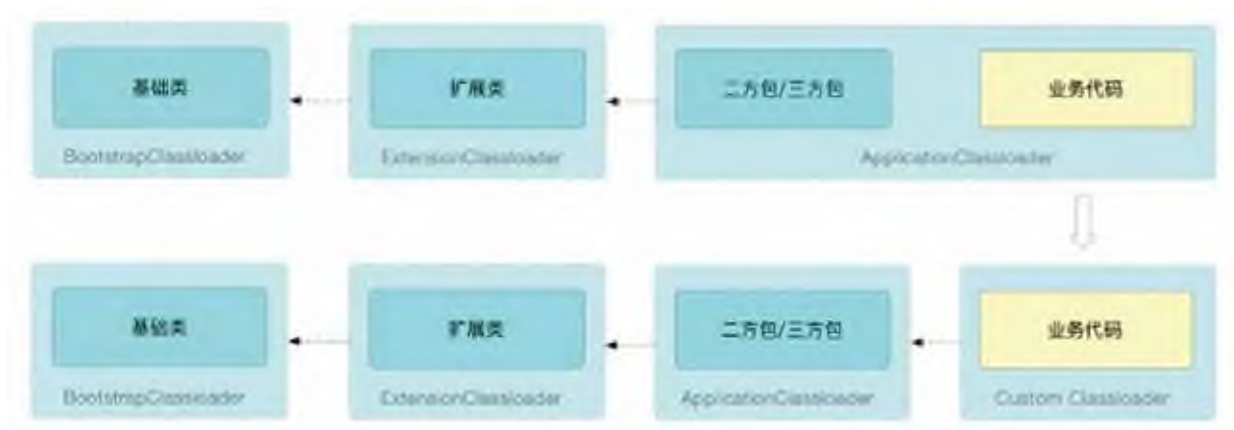
```
protected Class<?> loadClass(String name, boolean resolve)
    throws ClassNotFoundException {
    //导出类中是否存在，如果存在则直接返回
    if (classCache != null && classCache.containsKey(name)) {
        return classCache.get(name);
    }
    //双亲委托机制加载
    synchronized (this.getClassLoadingLock(name)) {
        Class<?> clazz = this.findLoadedClass(name);
        if (clazz == null) {
            try {
                if (this.parent != null) {
                    clazz = this.parent.loadClass(name, false);
                } else {
                    clazz = this.findBootstrapClassOrNull(name);
                }
            } catch (ClassNotFoundException var10) {
            }
            if (clazz == null) {
                clazz = this.findClass(name);
            }
        }
        if (resolve) {
            this.resolveClass(clazz);
        }
        return clazz;
    }
}
```

热加载

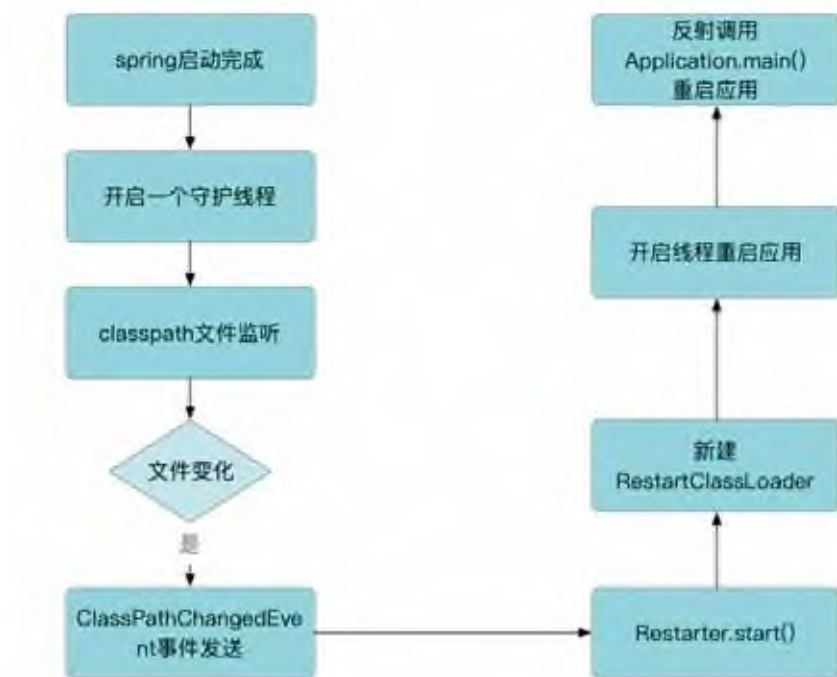
在开发项目的时候，我们需要频繁的重启应用进行程序调试，但是java项目的启动少则几十秒，多则几分钟。如此慢的启动速度极大地影响了程序开发的效率，那是否可以快速的进行启动，进而能够快速的进行开发验证呢？答案也是肯定的，通过classloader我们可以完成对变更内容的加载，然后快速的启动。

常用的热加载方案有好几个，接下来我们介绍下spring官方推荐的热加载方案，即spring boot devtools。

首先我们需要思考下，为什么重新启动一个应用会比较慢，那是因为在启动应用的时候，JVM虚拟机需要将所有的应用程序重新装载到整个虚拟机。可想而知，一个复杂的应用程序所包含的jar包可能有上百兆，每次微小的改动都是全量加载，那自然是很慢了。那么我们是否可以做到，当我们修改了某个文件后，在JVM中替换到这个文件相关的部分而不全量的重新加载呢？而spring boot devtools正是基于这个思路进行处理的。



如上图所示，通常一个项目的代码由以上四部分组成，即基础类、扩展类、二方包/三方包、以及我们自己编写的业务代码组成。上面的一排是我们通常的类加载结构，其中业务代码和二方包/三方包是由应用加载器加载的。而实际开发和调试的过程中，主要变化的是业务代码，并且业务代码相对二方包/三方包的内容来说会更少一些。因此我们可以将业务代码单独通过一个自定义的加载器Custom Classloader来进行加载，当监控发现业务代码发生改变后，我们重新加载启动，老的业务代码的相关类则由虚拟机的垃圾回收机制来自动回收。其工程流程大概如下。有兴趣的同学可以去看下源码，会更加清楚。



RestartClassLoader为自定义的类加载器，其核心是loadClass的加载方式，我们发现其通过修改了双亲委托机制，默认优先从自己加载，如果自己没有加载到，从parent进行加载。这样保证了业务代码可以优先被RestartClassLoader加载。进而通过重新加载RestartClassLoader即可完成应用代码部分的重新加载。

```
public Class<?> loadClass(String name, boolean resolve) throws ClassNotFoundException {
    String path = name.replace('.', '/').concat(".class");
    ClassLoaderFile file = this.updatedFiles.getFile(path);
    if (file != null && file.getKind() == Kind.DELETED) {
        throw new ClassNotFoundException(name);
    }
    synchronized (getClassLoadingLock(name)) {
        Class<?> loadedClass = findLoadedClass(name);
        if (loadedClass == null) {
            try {
                //优先从自己加载(编译生成的target/classes目录)
                loadedClass = findClass(name);
            }
            catch (ClassNotFoundException ex) {
                //如果没有加载到，则从父类来加载
                loadedClass = Class.forName(name, false, getParent());
            }
        }
        if (resolve) {
            resolveClass(loadedClass);
        }
        return loadedClass;
    }
}
```

热部署

热部署本质其实与热加载并没有太大的区别，通常我们说热加载是指在开发环境中进行的classloader加载，而热部署则更多是指在线上环境使用classloader的加载机制完成业务的部署。所以这二者使用的技术并没有本质的区别。那热部署除了与热加载具有发布更快之外，还有更多的更大的优势就是具有更细的发布粒度。我们可以想像以下的一个业务场景。



假设某个营销投放平台涉及到4个业务方的开发，需要对会场业务进行投放。而这四个业务方的代码全部都在一个应用里面。因此某个业务方有代码变更则需要对整个应用进行发布，同时其它业务方也需要跟着回归。因此每个微小的发动，则需要走整个应用的全量发布。这种方式带来的稳定性风险估且不说，整个发布迭代的效率也可想而知了。这在整个互联网里，时间和效率就是金钱的理念下，显然是无法接受的。

那么我们完全可以通过类加载机制，将每个业务方通过一个classloader来加载。基于类的隔离机制，可以保障各个业务方的代码不会相互影响，同时也可以做到各个业务方进行独立的发布。其实在移动客户端，每个应用模块也可以基于类加载，实现插件化发布。本质上也是一个原理。

在阿里内部像阿拉丁投放平台，以及crossbow容器化平台，本质都是使用classloader的热加载技术，实现业务细粒度的开发部署以及多应用的合并部署。

加密保护

众所周知，基于java开发编译产生的jar包是由.class字节码组成，由于字节码的文件格式是有明确规范的。因此对于字节码进行反编译，就很容易知道其源码实现了。因此大致会存在如下两个方面的诉求。例如在服务端，我们向别人提供三方包实现的时候，不希望别人知道核心代码实现，我们可以考虑对jar包进行加密，在客户端则会比较普遍，那就是我们打包好的apk的安装包，不希望被人家反编译而被人家翻个底朝天，我们也可以对apk进行加密。

jar包加密的本质，还是对字节码文件进行操作。但是JVM虚拟机加载class的规范是统一的，因此我们在最终加载class文件的时候，还是需要满足其class文件的格式规范，否则虚拟机是不能正常加载的。因此我们可以在打包的时候对class进行正向的加密操作，然后，在加载class文件之前通过自定义classloader先进行反向的解密操作，然后再按照标准的class文件标准进行加载，这样就完成了class文件正常的加载。因此这个加密的jar包只有能够实现解密方法的classloader才能正常加载。

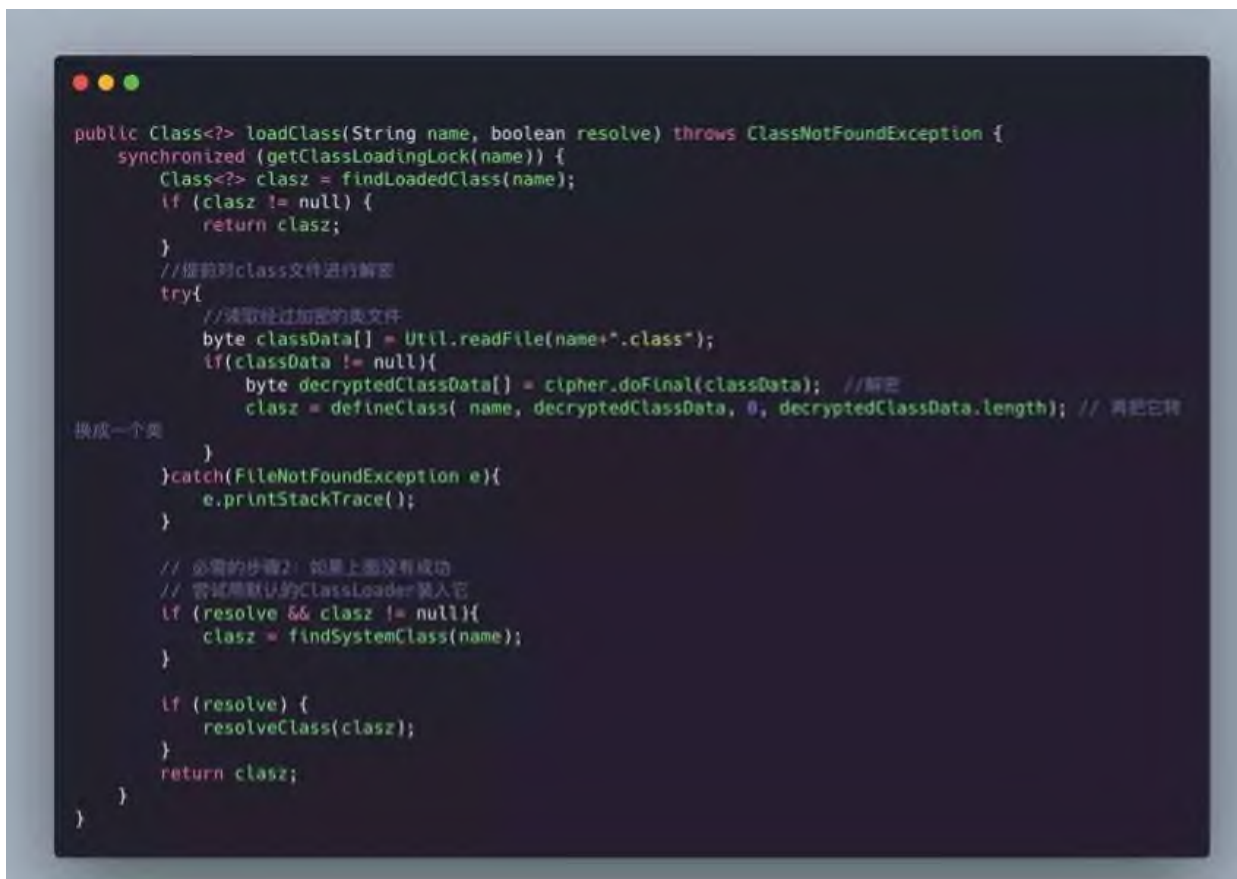
打包



加载



我们可以贴一下简单的实现方案：



```
public Class<?> loadClass(String name, boolean resolve) throws ClassNotFoundException {
    synchronized (getClassLoadingLock(name)) {
        Class<?> classz = findLoadedClass(name);
        if (classz != null) {
            return classz;
        }
        // 提前对class文件进行解密
        try {
            // 读取经过加密的类文件
            byte classData[] = Util.readFile(name + ".class");
            if (classData != null) {
                byte decryptedClassData[] = cipher.doFinal(classData); // 解密
                classz = defineClass(name, decryptedClassData, 0, decryptedClassData.length); // 再把它转
                换成一个类
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        // 必需的步骤2: 如果上面没有成功
        // 尝试用默认的ClassLoader装入它
        if (resolve && classz != null) {
            classz = findSystemClass(name);
        }

        if (resolve) {
            resolveClass(classz);
        }
        return classz;
    }
}
```

这样整个jar包的安全性就有一定程度的提高，至于更高安全的保障则取决于加密算法的安全性了以及如何保障加密算法的密钥不被泄露的问题了。这有种套娃的感觉，所谓安全基本都是相对的。并且这些方法也不是绝对的，例如可以通过对classloader进行插码，对解密后的class文件进行存储；另外大多数JVM本身并不安全，还可以修改JVM，从ClassLoader之外获取解密后的代码并保存到磁盘，从而绕过上述加密所做的一切工作，当然这些操作的成本就比单纯的class反编译就高很多了。所以说安全保障只要做到使对方破解的成本高于收益即是安全，所以一定程度的安全性，足以减少很多低成本的攻击了。

案例简介

本文对classloader的加载过程和加载原理进行了介绍，并结合类加载机制的特征，介绍了其相应的使用场景。由于篇幅限制，并没有对每种场景的具体实现细节进行介绍，而只是阐述了其基本实现思路。或许大家觉得classloader的应用有些复杂，但事实上只要大家对class从哪里加载，搞清楚loadClass的机制，就已经成功了一大半。正所谓万变不离其宗，抓住了本质，其它问题也就迎刃而解了。

• 如何在业务逻辑当中优雅引入重试机制

- 作者：殷浩
- 出品：淘系技术

为什么要引入重试机制

我们首先看看正常的业务系统交互流程，就像下面图中所示一样，我们自己开发的系统通过HTTP接口或者通过RPC去访问其他业务系统，其他系统在没出现任何问题的情况下会返回给我们需要的数据，状态为success。

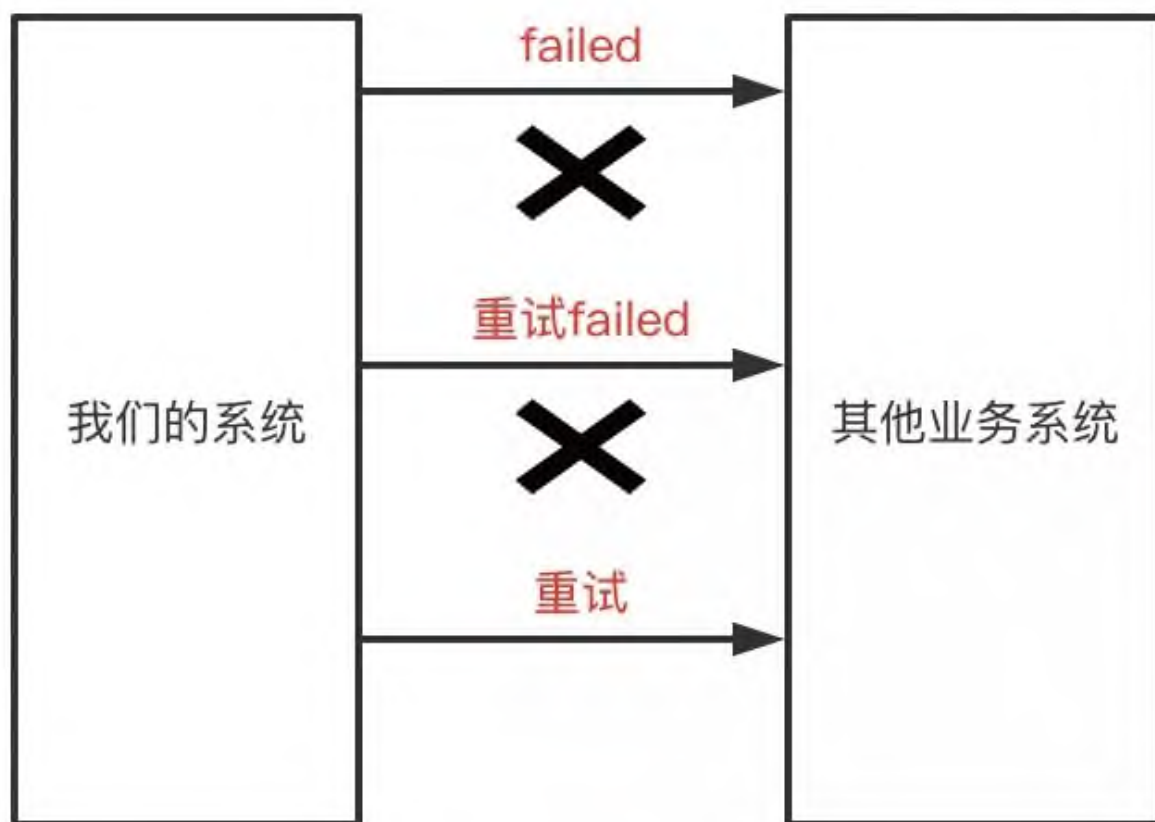


但大家在日常的开发工作当中应该碰到过不少这样的问题：自己应用因为业务需求需要调其他关联应用的接口或二方包，而其他应用的接口稳定性不敢过分恭维，老是出一些莫名奇妙的幺蛾子，比如由于接口暂时升级维护导致的短暂不可用，又或者网络抖动因素导致的单次接口请求失败。



诸如此类的麻烦问题会因为业务强依赖致使我们自己维护的系统也跟着陷入一种不稳定的状态(当然这个强依赖是没有办法的事情，毕竟业务之间需要解耦独立开发维护)。

所以也就是说重试的使用场景大多是因为我们的系统依赖了其他的业务，或者是由于我们自己的业务需要通过网络请求去获取数据这样的场景。既然一次请求结果的状态非常不可控、不稳定，那么一个非常自然的想法就是多试几次，就能很好的避开网络抖动或其他关联应用暂时down机维护带来的系统不可用问题。



当然，这里也有几个引入重试机制以后需要考虑的问题。

- 我们应该重试几次？
- 每次重试的间隔设置为多少合适？
- 如果所有重试机会都用完了还是不成功怎么办？下面我们就这几个问题展开分析一下。

重试几次合适

通常来说我们单次重试所面临的情况就如上面我们分析的一样，有很大的不可确定性，那到底多少次是比较合理的次数呢？这个就要“具体业务具体分析”了，但一般来说3次重试就差不多可以满足大多数业务需求了，当然，这是需要结合后面要说的重试间隔一起讨论的。为什么说3次就基本够了呢，因为如果被请求系统实在处于长时间不可用状态。我们重试多次是没有什么意义的。

重试间隔设置为多少合适

如果重试间隔设置得太小，可能被调用系统还没来得及恢复过来我们就又发起调用，得到的结果肯定还是Fail；如果设置的太大，我们自己的系统就会牺牲掉不少数据时效性。所以，重试间隔也要根据被调用的系统平均恢复时间去正确估量，通常而言这个平均恢复时间很难统计到，所以一般的经验值是3至5分钟。

重试机会用完以后依旧Fail怎么办

这种情况也是需要认真考虑的，因为不排除被调用系统真的起不来的情况，这时候就需要采取一定的补偿措施了。首先要做的就是在我们自己的系统增加错误报警机制，这样我们才能即时感知到应用发生了不可自恢复的调用异常。其次就是在我们的代码逻辑中加入触发手动重试的开关，这样在发生异常情况以后我们就可以方便的修改触发开关然后手动重试。

在这里还有一个非常重要的问题需要考虑，那就是接口调用的幂等性问题，如果接口不是幂等的，那我们手动重试的时候就很容易发生数据错乱相关的问题。

Spring重试工具包

Spring为我们提供了原生的重试类库，我们可以方便地引入到工程当中，利用它提供的重试注解，没有太多的业务逻辑侵入性。如下，我们先引入依赖包。

```
1 <dependency>
2   <groupId>org.springframework.retry</groupId>
3   <artifactId>spring-retry</artifactId>
4 </dependency>
5 <dependency>
6   <groupId>org.aspectj</groupId>
7   <artifactId>aspectjweaver</artifactId>
8 </dependency>
```

然后在启动类或者配置类上添加@EnableRetry注解，并在需要重试的方法上添加@Retryable注解。

```
1 @Retryable
2 public String hello(){
3     long times = helloTimes.incrementAndGet();
4     log.info("hello times:{}, times);
5     if (times % 4 != 0){
6         log.warn("发生异常，time: {}", LocalTime.now());
```

```
7         throw new HelloRetryException("发生Hello异常");
8     }
9     return "hello " + nameService.getName();
10 }
```

更详细的用法和属性大家参阅Spring Retry的文档就好了，解释的非常清楚。这里要说的一点是，Spring的重试机制也还是存在一定的不足，只支持对异常进行捕获，而无法对返回值进行校验。

Guava Retry

相比Spring Retry，Guava Retry具有更强的灵活性，可以根据返回值校验来判断是否需要重试。我们依然需要先引入它的依赖包。

```
1 <dependency>
2   <groupId>com.github.rholder</groupId>
3   <artifactId>guava-retrying</artifactId>
4   <version>2.0.0</version>
5 </dependency>
```

在用的时候也很简单，先创建一个Retryer实例，然后使用这个实例对需要重试的方法进行调用，可以通过很多方法来设置重试机制，比如使用retryIfException来对所有异常进行重试，使用retryIfExceptionOfType方法来设置对指定异常进行重试，使用retryIfResult来对不符合预期的返回结果进行重试，使用retryIfRuntimeException方法来对所有RuntimeException进行重试。

```
1 @Test
2 public void guavaRetry() {
3     Retryer<String> retryer = RetryerBuilder.<String>newBuilder()
4         .retryIfExceptionOfType(HelloRetryException.class)
5         .retryIfResult(StringUtils.isEmpty)
6         .withWaitStrategy(WaitStrategies.fixedWait(3, TimeUnit.SECONDS))
7         .withStopStrategy(StopStrategies.stopAfterAttempt(3))
8         .build();
9
10    try {
11        retryer.call(() -> helloService.hello());
12    } catch (Exception e){
13        e.printStackTrace();
14    }
15 }
```

相比Spring，Guava Retry提供了几个核心特性。

- 可以设置任务单次执行的时间限制，如果超时则抛出异常。
- 可以设置重试监听器，用来执行额外的处理工作。
- 可以设置任务阻塞策略，即可以设置当前重试完成，下次重试开始前的这段时间做什么事情。
- 可以通过停止重试策略和等待策略结合使用来设置更加灵活的策略，比如指数等待时长并最多10次调用，随机等待时长并永不停止等等。

小结

上面针对我们为什么要引入重试机制，引入重试机制需要思考的几个核心问题，以及为重试机制提供良好支持的工具类库都分别作了简单介绍，相信大家在今后的开发工作中遇到类似场景也能驾轻就熟地使用思考了。

我们日常工作中有很多“大”的业务场景需要我们集中精力去突破、去思考，但也有很多类似的“小”点需要我们去打穿、吃透，大家共勉。

• 怎么提高自己的系统设计和架构理论水平？

- 作者：勇剑
- 出品：淘系技术

常规业务系统设计关键-领域模型

业务系统设计的关键是在于如何定义系统的模型以及模型之间的关系，其中主要是领域模型的定义，当我们在模型确定之后，模型之间的关系也会随之明确。

模型设计可以参考领域模型的经典书籍《Domain-Driven Design》一书，通过这个基本可以对领域定义、防腐层、贫血模型等概念有一个较为清晰的认识了。

单个应用内的领域模型系统也需要注意领域分层，作为开发大家是不是见过、重构过很多Controller-Service-DAO 样式的代码分层设计？往往在做重构的时候会令人吐血。

设计较好的领域设计这里给一个分层建议：

- **接口层 Interface**：主要负责与外部系统进行交互&通信，比如一些 dubbo服务、Restful API、RMI等，这一层主要包括 Facade、DTO还有一些Assembler。
- **应用层 Application**：这一层包含的主要组件就是 Service 服务，但是要特别注意，这一层的Service不是简单的DAO层的包装，在领域驱动设计的架构里面，Service层只是一层很“薄”的一层，它内部并不实现任何逻辑，只是负责协调和转发、委派业务动作给更下层的领域层。
- **领域层 Domain**：Domain 层是领域模型系统的核心，负责维护面向对象的领域模型，几乎全部的业务逻辑都会在这一层实现。内部主要包含Entity（实体）、ValueObject（值对象）、Domain Event（领域事件）和 Repository（仓储）等多种重要的领域组件。
- **基础设施层 Infrastructure**：它主要为 Interfaces、Application 和 Domain 三层提供支撑。所有与具体平台、框架相关的实现会在 Infrastructure 中提供，避免三层特别是 Domain 层掺杂进这些实现，从而“污染”领域模型。Infrastructure 中最常见的一类设施是对象持久化的具体实现。

高并发系统设计

在面试中是不是经常被问到一个问题：如果你系统的流量增加 N 倍你要怎么重新设计你的系统？这个高并发的的问题可以从各个层面去解，比如

代码层面：

- 锁优化（采用无锁数据结构），主要是 concurrent 包下面的关于 AQS 锁的一些内容
- 数据库缓存设计（降低数据库并发争抢压力），这里又会有缓存、DB 数据不一致的问题，在实际使用中，高并发系统和数据一致性系统采用的策略会截然相反。
- 数据更新时采用合并更新，可以在应用层去做更新合并，同一个 Container 在同一时间只会有一个 DB 更新请求。
- 其他的比如基于 BloomFilter 的空间换时间、通过异步化降低处理时间、通过多线程并发执行等等。

数据库层面：

- 根据不同的存储诉求来进行不同的存储选型，从早期的 RDBMS，再到 NoSql（KV 存储、文档数据库、全文索引引擎等等），再到最新的 NewSql（TiDB、Google spanner/F1 DB）等等。
- 表数据结构的设计，字段类型选择与区别。
- 索引设计，需要关注聚簇索引原理与覆盖索引消除排序等，至于最左匹配原则都是烂大街的常识了，高级一点索引消除排序的一些机制等等，B+树与B树的区别。
- 最后的常规手段：分库分表、读写分离、数据分片、热点数据拆分等等，高并发往往会做数据分桶，这里面往深了去说又有很多，比如分桶如何初始化、路由规则、最后阶段怎么把数据合并等等，比较经典的方式就是把桶分成一个主桶+N个分桶。

架构设计层面：

- 分布式系统为服务化
- 无状态化支持水平弹性扩缩容
- 业务逻辑层面 failfast 快速失败
- 调用链路热点数据前置
- 多级缓存设计
- 提前容量规划等等

Application层

对于可用性要求非常高的系统，一般我们都说几个9的可用率，比如 99.999% 等。

面对高可用系统设计也可以从各个方面来进行分析

代码层面： 需要关注分布式事务问题，CAP理论是面试的常规套路

软件层面： 应用支持无状态化，部署的多个模块完全对等，请求在任意模块处理结果完全一致 => 模块不存储上下文信息，只根据请求携带的参数进行处理。目的是为了快速伸缩，服务冗余。常见的比如session问题等。

负载均衡问题： 软件部署多份之后，如何保证系统负载？如何选择调用机器？也就是负载均衡问题

- 狭义上的负载均衡按照类型可以分为这几种：
- 硬件负载：比如F5等
- 软件负载：比如 LVS、Nginx、HaProxy、DNS等。
- 当然，还有代码算法上的负载均衡，比如Random、RoundRobin、ConsistentHash、加权轮训等等算法
- 广义上的负载均衡可以理解为负载均衡的能力，比如一个负载均衡系统需要如下4个能力：
- 故障机器自动发现
- 故障服务自动摘除（服务熔断）
- 请求自动重试
- 服务恢复自动发现

幂等设计问题

上面提负载均衡的时候，广义负载均衡需要完成自动重试机制，那么在业务上，我们就必须保证幂等设计。这里可以从2个层面来进行考虑：

- 请求层面：由于请求会重试所以必须做幂等，需要保证请求重复执行和执行一次的结果完全相同。请求层面的幂等设计需要在数据修改的层做幂等，也就是数据访问层读请求天然幂等，写请求需要做幂等。读请求一般是天然幂等的，无论查询多少次返回的结果都是一致。这其中的本质实际上是分布式事务问题，这里下面再详细介绍。
- 业务层面：不幂等会造成诸如奖励多发、重复下单等非常严重的问题。业务层面的幂等本质上是分布式锁的问题，后面会介绍。如何保证不重复下单？这里比如token机制等等。如何保证商品不超卖？比如乐观锁等。MQ消费方如何保证幂等等都是面试的常见问题。

分布式锁

业务层面的幂等设计本质上是分布式锁问题，什么是分布式锁？分布式环境下锁的全局唯一资源，使请求串行化，实际表现互斥锁，解决业务层幂等问题。常见的解决方式是基于 Redis 缓存的 setnx 方法，但作为技术人员应该清楚这其中还存在单点问题、基于超时时间无法续租问题、异步主从同步问题等等，更深一点，CAP理论，一个AP系统本质上无法实现一个AP需求，即使是 RedLock 也不行。那我们如何去设计一个分布式锁呢？强一致性、服务本身要高可用是最基本的需求，其他的比如支持自动续期，自动释放机制，高度抽象接入简单，可视化、可管理等。基于存储层的可靠的解决方案比如

- zookeeper: CP/ZAB/N+1可用：基于临时节点实现和Watch机制。
- ETCD: CP or AP/Raft/N+1可用：基于 restful API；KV存储，强一致性，高可用，数据可靠：持久化；Client TTL 模式，需要心跳CAS 唯一凭证 uuid。

服务的熔断

微服务化之后，系统分布式部署，系统之间通过 RPC 通讯，整个系统发生故障的概率随着系统规模的增长而增长，一个小的故障经过链路传导放大，有可能造成更大的故障。希望在调用服务的时，在一些非关键路径服务发生服务质量下降的情况下，选择尽可能地屏蔽所造成的影响。大部分熔断返回默认值 null，也可以定制，RPCClient 原生支持最好，业务方少改代码（熔断放的地方），进入熔断时，打印熔断日志，同时返回 Exception（业务方定制熔断方法），需要有服务治理平台，可以看到服务的状态、是否降级、是否熔断、可以实时下发阈值配置等。

服务降级

服务整体负载超出预设的上限，或者即将到来的流量预计将会超过阈值，为了保证重要或者基本的服务能够正常运行，拒绝部分请求或者将一些不重要的不紧急的服务或任务进行服务的延迟使用或暂停使用

主要的手段如下：

- 服务层降级，主要手段：
 - 拒绝部分请求（限流），比如缓存请求队列，拒绝部分等待时间长的请求；根据Head，来拒绝非核心请求；还有其他通用算法上的限流比如令牌桶、漏桶算法等等。
 - 关闭部分服务：比如双11大促0点会关闭逆向退款服务等等。
 - 分级降级：比如自治式服务降级，从网关到业务到DB根据拦截、业务规则逐渐降低下游请求量，体现上是自上到下的处理能力逐渐下降。
- 数据层降级
 - 比如流量大的时候，更新请求只缓存到MQ，读请求读缓存，等流量小的时候，进行补齐操作(一般数据访问层如果做了降级，就没必要在数据层再做了)
- 柔性可用策略
 - 比如一些指定最大流量的限流工具，又或是根据CPU负载的限流工具等，需要保证自动打开，不依赖于人工。

发布方式引发的可用性问题

发布方式也是影响高可用的一个点，哈哈，以前还经历过一些线上直接停机发布的案例（银行内部系统），不过作为高大上的互联网，主要会采用这几种发布方式：

灰度发布、蓝绿发布、金丝雀发布等等。

数据一致性系统设计

一般一些金融、账务系统对这一块要求会非常严格，下面主要介绍下这里面涉及到的事务一致性、一致性算法等内容。

事务一致性问题

在 DB 层面，一般通过 **刚性事务** 来实现数据一致性，主要通过 预写日志(WAL) 的方式来实现，WAL(write ahead logging)预写日志的方式。就是所有对数据文件的修改，必须先写日志，这样，即使在写数据的时候崩溃了，也能通过日志文件恢复，传统的数据库事务就是基于这一个机制（REDO 已提交事务的数据也求改 UNDO 未提交事务的回滚）。

除了这个方式之外，还有一个就是通过 **影子数据块** 来进行数据备份，提前记录被修改的数据块的修改前的状态，备份起来，如果需要回滚，直接用这个备份的数据块进行覆盖就好了。其他的就是基于二阶段提交的 **XA模型** 了。

但是目前互联网系统，已经广泛采用分布式部署模式了，传统的刚性事务无法实现，所以 **柔性事务** 成了目前主流的分布式事务解决防范，主要的模式有下面几种：

TCC 模式/或者叫2阶段模式：在 try 阶段预扣除资源（但是不锁定资源，提升可用性），在Confirm 或者 Cancel 阶段进行数据提交或者回滚。一般需要引入协调者，或者叫事务管理器。

SAGA模式：业务流程中每个参与者都提交本地事务，当出现某一个参与者失败则补偿前面已经成功的参与者，支持向前或者向后补偿。

MQ的事务消息：就是先发 halfMsg，在处理完之后，再发送 commit 或者 rollback Msg，然后 MQ 会定期询问 producer，halfMsg 能不能 commit 或者 rollback，最终实现事务的最终一致性。实际上是把补偿的动作委托给了 RocketMQ。

分段事物（异步确保）：基于可靠消息+本地事务消息表 + 消息队列重试机制。目前这也是一些大厂的主流方案，内部一般称为**分段事物**。

柔性事务基本都是基于最终一致性去实现，所以肯定会有 **补偿** 动作在里面，在达到最终一致性之前，对用户一般展示 **软状态**。

需要注意的一点是，并不是所有的系统都适合引入数据一致性框架，比如用户可以随时修改自己发起的请求的情况，例如，商家设置后台系统，商户会随时修改数据，这里如果涉及到一致性的话，引入一致性框架会导致补偿动作达到最终一致性之前，资源锁会阻塞用户后续的请求。导致体验较差。这种情况下就需要通过其他手段来保障数据一致性了，比如数据对账等操作。

一致性算法

从早期的 Paxos 算法，再到后面衍生的 zab 协议（参考：A simple totally ordered broadcast protocol），提供了当下可靠的分布式锁的解决方案。再到后来的 Raft 算法（In Search of an Understandable Consensus Algorithm），也都是分布式系统设计里面需要了解的一些知识要点。

最后

这里简单介绍了不同系统设计的时候会面临的一些难点，基本里面每一个点，都是前人在解决各种疑难问题的道路上不断探索，最终才得出的这些业界解决方案，呈现在大家眼前，作为一个技术人员，学会这些技术点只是时间问题，但这种发现问题、直面问题、再到解决问题的能力 and 精神才是我们最值得学习的地方，也是做为一个系统设计人员或者说是架构师的必要能力。

最后，祝进步～！

• 公司项目并发量都特小，自己如何实际接触高并发项目？

- 作者：昭明
- 出品：淘系技术

想提升高并发系统的设计和开发能力，有2个方面：

- 一个是系统的学习相关理论；
- 一个是找一个目标系统，不断想办法去提升他的性能。

前者是后者的理论基础。

如果想从事一个高并发系统开发的岗位，要学习的相关技术其实是很多的，这些技术核心就是解决高并发情况下如何保持系统的高可用和低延迟。

以Java工程师为例，互联网程序员面试中经常会考察的内容包括：

（1）架构设计：

高可用与稳定性、事务一致性、多副本一致性、CAP理论。

（2）相关技术：

多线程（JUC/AQS/线程池）、RPC调用及框架（如Thrift）、NIO及NIO框架（如Netty）、高并发框架（如Disruptor）、微服务框架（SpringBoot）、微服务治理（Spring Cloud）、数据库相关技术（如：索引优化、分库分表、读写分离）、分布式缓存（如redis）、消息中间件系统（如RabbitMQ）、容器技术（如docker）。

（3）工具：

系统性能查看（top、uptime、vmstat、iostat）、压测工具（如ab、locust、Jmeter、go）、线程分析（如jps、jstack）等。

当然，一开始，我们不可能逐一把这些技能全部掌握，我们可以从一个实际项目入手，不断的把这些技术用上去，发现哪些知识不足，再去补充相关的知识。

“如何设计一个好的秒杀系统”，一定是互联网大厂面试中最常问的一个问题。所以从设计一个秒杀系统开始实践，是个不错的选择。

秒杀系统的特点：

（1）瞬时并发量大

秒杀时会有大量用户在同一时间进行抢购，瞬时并发访问量突增 10 倍，甚至 100 倍以上都有。

（2）库存量少

一般秒杀活动商品量很少，这就导致了只有极少量用户能成功购买到。

（3）业务简单

流程比较简单，一般都是下订单、扣库存、支付订单。

设计秒杀系统的关键点：

（1）限流

由于活动库存量一般都是很少，对应的只有少部分用户才能秒杀成功。所以我们需要限制大部分用户流量，只准少量用户流量进入后端服务器。

（2）削峰

秒杀开始的那一瞬间，会有大量用户冲击进来，所以在开始时候会有一个瞬间流量峰值。如何把瞬间的流量峰值变得更平缓，是能否成功设计好秒杀系统的关键因素。实现流量削峰填谷，一般的采用缓存和 MQ 中间件来解决。

（3）异步

秒杀其实可以当做高并发系统来处理，在这个时候，可以考虑从业务上做兼容，将同步的业务，设计成异步处理的任务，提高网站的整体可用性。

（4）缓存

秒杀系统的瓶颈主要体现在下订单、扣减库存流程中。在这些流程中主要用到 OLTP 的数据库，类似 MySQL、Oracle。由于数据库底层采用 B+ 树的储存结构，对应我们随机写入与读取的效率，相对较低。如果我们把部分业务逻辑迁移到内存的缓存或者 Redis 中，会极大的提高并发效率。

从0到1搭建一个秒杀系统，也并不容易，涉及到很多前端、后端、中间件的技术。这个跟其实是所有公司的工作常态，大部分时间也是在搭架子，真正做技术优化的时间并不多，经常是在业务量突增或者大促活动来临时，集中搞一波性能优化。

所以，如果没有实际的高并发项目可做，自己弄个秒杀系统自娱自乐也是不错的。

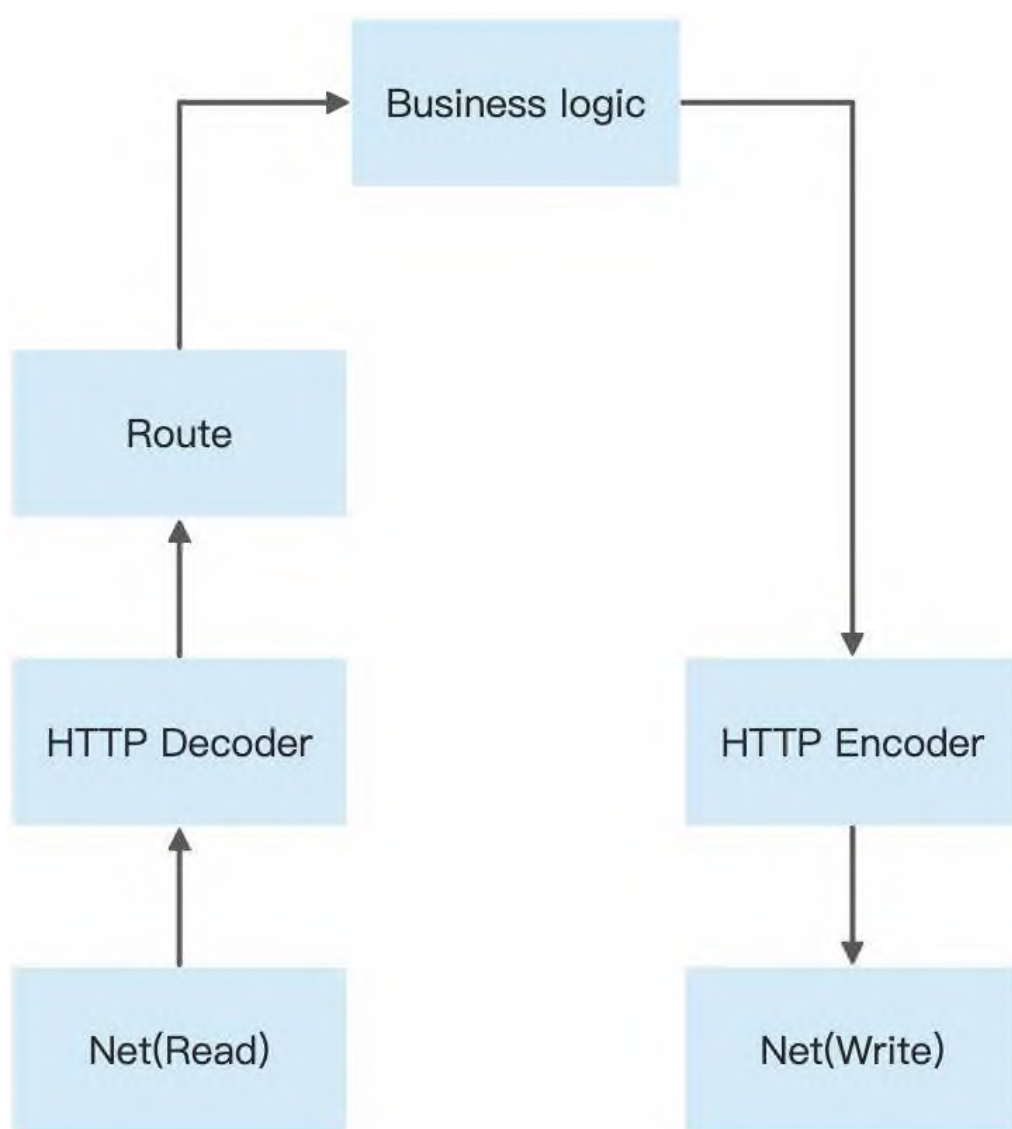
搭建系统 -> 压测 -> 发现问题 -> 学习知识 -> 优化系统，通过这样的循环，相信你一定既能体验到学习的乐趣，同时实力也大幅提升。

• 为什么Go的web框架速度还不如Java?

- 作者：风弈
- 出品：淘系技术

http server 概述

首先描述一下一个简单的 web server 的请求处理过程：



Net 层读取数据包后经过 HTTP Decoder 解析协议，再由 Route 找到对应的 Handler 回调，处理业务逻辑后设置相应 Response 的状态码等，然后由 HTTP Encoder 编码相应的 Response，最后由 Net 写出数据。

而 Net 之下的一层由内核控制，虽然也有很多优化策略，但这里主要比较 web 框架本身，那么暂时不考虑 Net 之下的优化。

看了下 techempower 提供的压测框架源码，各类框架基本上都是基于 epoll 的处理，那么各类框架的性能差距主要体现在上述这些模块的性能了。

关于各类压测的简述

我们再看 techempower 的各项性能排名，有 JSON serialization, Single query, Multiple queries, Cached queries, Fortunes, Data updates 和 Plaintext 这几大类的排名。

其中 JSON serialization 是对固定的 Json 结构编码并返回 (message: hello word), Single query 是单次 DB 查询，Multiple queries 是多次 DB 查询，Cached queries 是从内存数据库中获取多个对象值并以 json 返回，Fortunes 是页面渲染后返回，Data updates 是对 DB 的写入，Plaintext 是最简单的返回固定字符串。

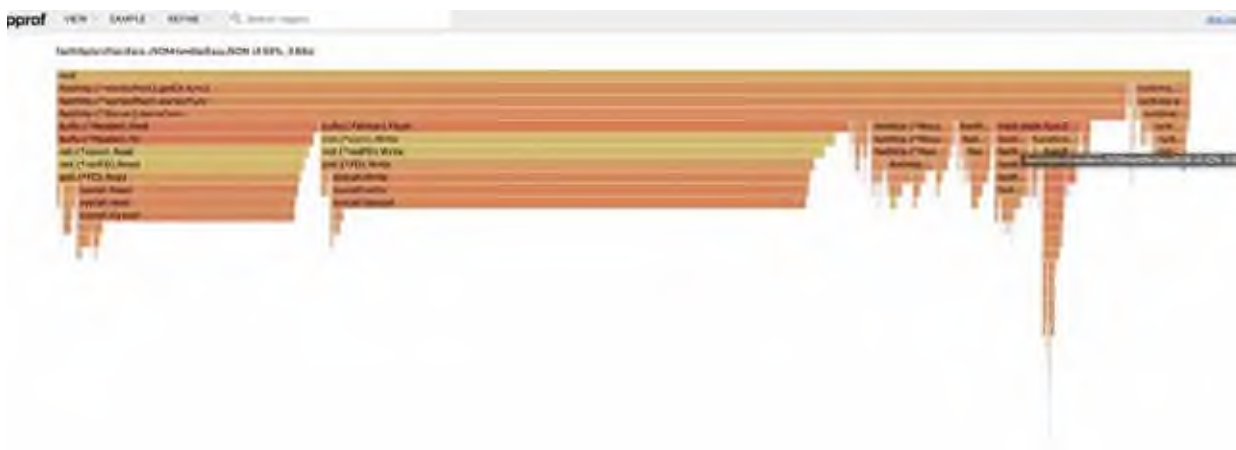
这里的 json 编码，DB 操作，页面渲染和固定字符串返回就是相应的业务逻辑，当业务逻辑越重(耗时越大)时，则相应的业务逻辑逐渐就成为了瓶颈，例如 DB 操作其实主要是在测试相应 DB 库和 DB 本身处理逻辑的性能，而框架本身的基础功能消耗随着业务逻辑的繁重将越来越忽略不计(Round 19 中物理机下 Plaintext 下的 QPS 在七百万级，而 Data updates 在万级别，相差百倍以上)，所以这边主要分析 Json serialization 和 Plaintext 两种相对能比较体现出框架本身 http 性能的排名。

在 Round 19 Json serialization 中 Java 性能最高的框架是 firenio-http-lite (QPS: 1,587,639)，而 Go 最高的是 fasthttp-easyjson-prefork(QPS: 1,336,333)，按照这里面的数据是 Java 性能高。



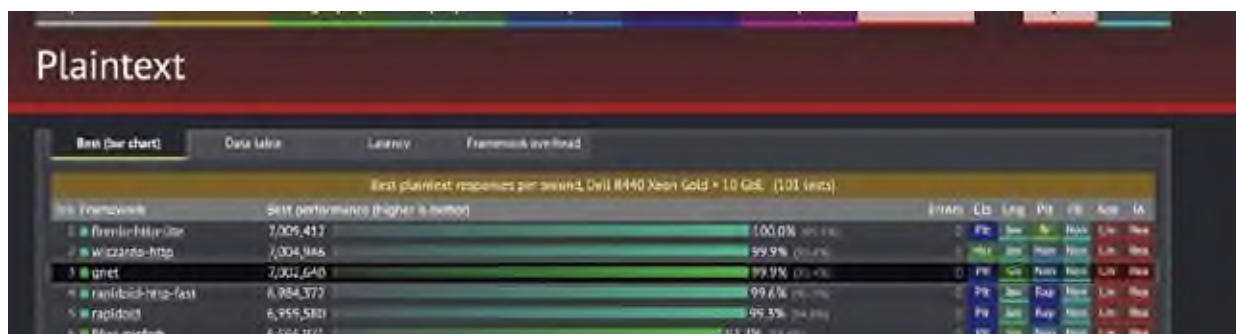
Framework	QPS	Latency (ms)	Throughput (MB/s)	Score
firenio-http-lite	1,587,639	0.00063	1.587639	100
fasthttp-easyjson-prefork	1,336,333	0.00075	1.336333	84

从 fasthttp-easyjson-prefork 的 pprof 看除了 read 和 write 外，json (相当于 Business logic) 占了 4.5%，fasthttp 自身(HTTP Decoder, HTTP Encoder, Router)占了 15%，仅看 Json serialization 似乎会有一种 Java 比 Go 性能高的感觉。

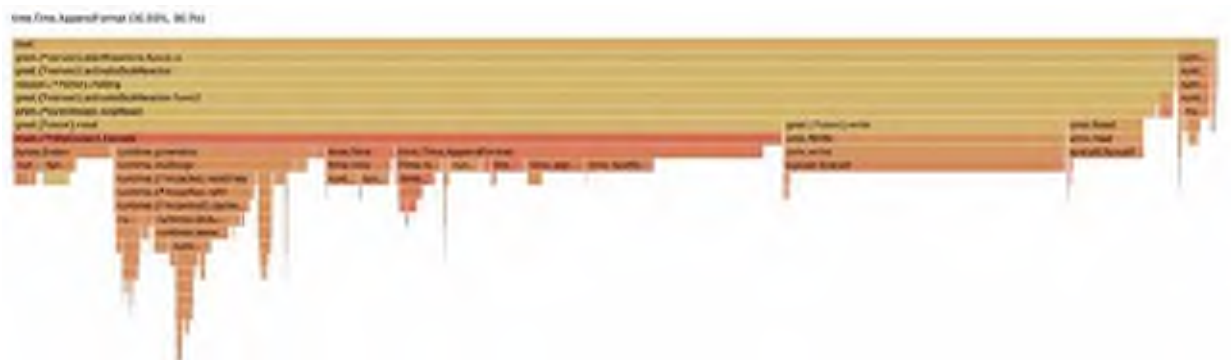


那我们继续把业务逻辑简化，看一下 Plaintext 的排名，Plaintext 模式其实是在使用 HTTP pipeline 模式下压测的，在 Round 19 中 Java 和 Go 已经几乎一样的 QPS 了，在 Round 19 之后的一次测试中 gnet 已经排在所有语言的第二，但是前几个框架QPS其实差别很微小。

这时候其实主要瓶颈都在 net 层，而 go 官方的 net 库包含了处理 goroutine 相关的逻辑，像 gonet 之类的直接操作 epoll 的会少一些这方面的消耗，Java 的 nio 也是直接操作的 epoll 。



拿了 gnet 的测试源码跑了下压测，看到 pprof 如下，其实这里 gnet 还有更进一步的性能优化空间：time.-Time.AppendFormat 占用 30% CPU。



可以使用如下提前 Format，允许减少获取当前时间精度的情况下大幅减少这部分的消耗。

```
var timetick atomic.Value

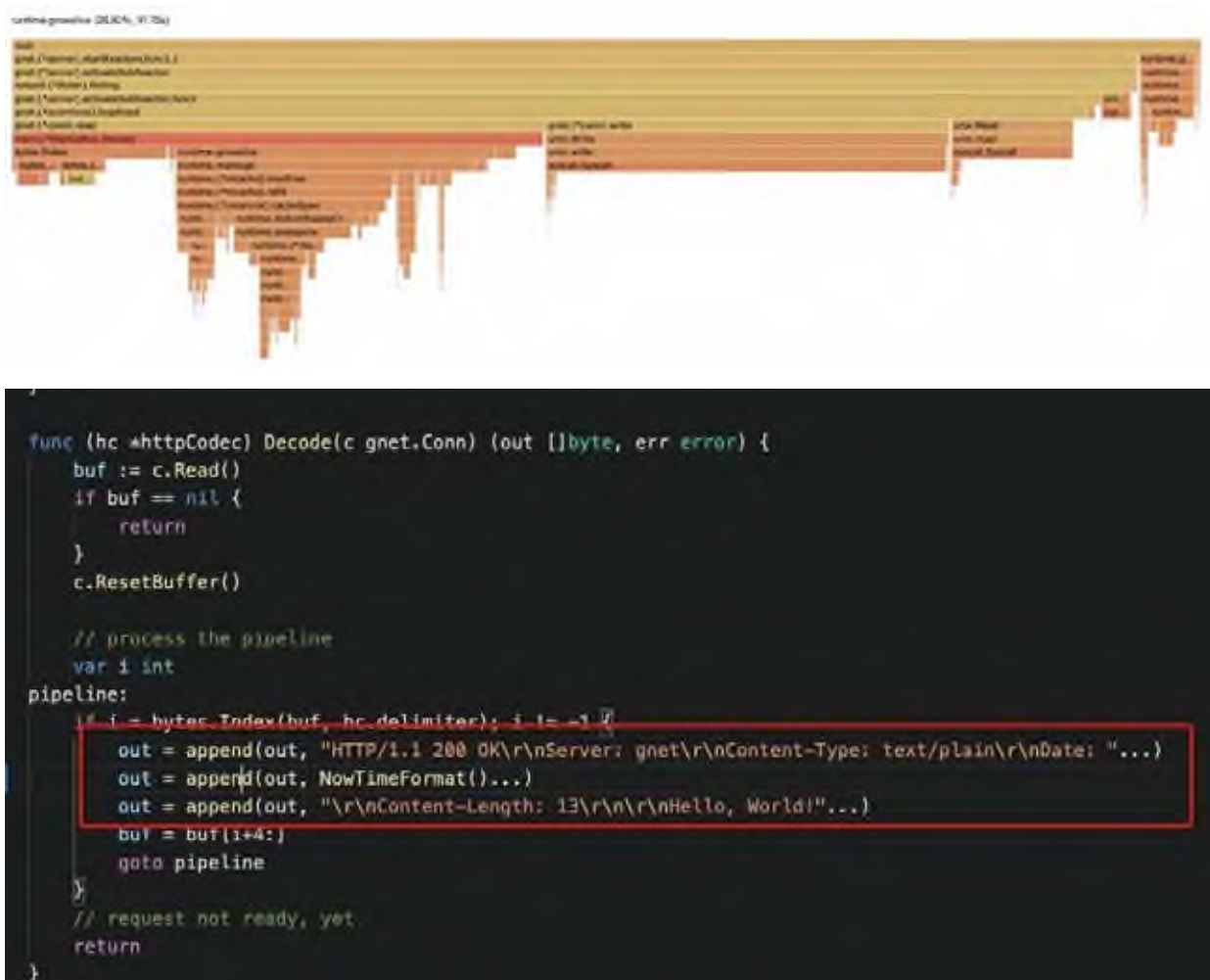
func NowTimeFormat() []byte {
    return timetick.Load().([]byte)
}

func tickloop() {
    timetick.Store(nowFormat())
    for range time.Tick(time.Second) {
        timetick.Store(nowFormat())
    }
}

func nowFormat() []byte {
    return []byte(time.Now().Format("Mon, 02 Jan 2006 15:04:05 GMT"))
}

func init() {
    timetick.Store(nowFormat())
    go tickloop()
}
```

这样优化后接下来的瓶颈在于 runtime 的内存分配，是由于这个压测代码中还存在下面的部分没有复用内存：



其实 gnet 本身的消耗已经做到非常小了，而 c++ 的 ulib 也是类似这样使用的非常简单的 HTTP 编解码操作来压测。

分析

对于这里面测试的框架，影响因素主要如下：

2、zero copy 和内存复用：内部处理字节的 0 拷贝（go 官方 http 库为了减少开发者的出错概率，没有使用 zero copy，否则开发者可能在无意中引用了已经放回 buff 池内的数据造成没有意识到的并发问题等等），而内存复用，大部分框架或多或少都已经做了。

3、prefork: 注意到 go 框架中有使用了 prefork 进程的方式(比如 fasthttp-prefork), 这是 fork 出多个子进程, 共享同一个 listen fd, 且每个进程使用单核但并发(1 个 P)处理的逻辑可以避免 go runtime 内部的锁竞争和 goroutine 调度的消耗(但是 go runtime 中为了并发和 goroutine 调度而存在的相关“无用”代码的消耗还是会有一些)

4、语言本身的性能差异对于第一点, 其实简化了各种编解码和路由之后, 虽然提高了性能, 但是往往会降低框架的易用性, 对于一般的业务而言, 不会出现如此高的QPS, 同时选择框架的时候往往还需要考虑易用性和可扩展性等, 同时还需要考虑到公司内部原有中间件或者 SDK 所使用的框架集成复杂度。

对于第二点, 如果是作为一个网络代理而言, 没有业务方的开发, 往往可以使用真正的完全 zero copy, 但是作为业务开发框架提供出去的话是需要考虑一定的业务出错概率, 往往牺牲一部分性能是划算的。

第三点 prefork, java netty 等是直接对于线程操作, 可以更加定制化的优化性能, 而 go 的 goroutine 需要的是一个通用协程, 目的是降低编写并发程序的难度, 在这个层次上难免性能比不上一个优化的非常出色的 Java 基于线程操作的框架; 但是直接操作线程的话需要合理控制好线程数, 这是个比较头疼的调优问题(特别是对于新手来说), 而 goroutine 则可以不太关心池子的大小, 使得代码更加优雅和简洁, 这对于工程质量保障其实是一个提升。另外这里存在 prefork 是由于 go 没法直接操作线程, 而 fasthttp 提供了 prefork 的能力, 使用多进程方式来对标 Java 的多线程来进一步提高性能。

第四点, 语言本身来说 Java 还是更加的成熟, 包括 JVM 的 Jit 能力也使得在热代码中和 Go 编译型语言的差异不大, 何况 Go 本身的编译器还不是特别成熟, 比如逃逸分析等方面的问题, Go 本身的内存模型和 GC 的成熟度也比不上 Java。还有很重要的一点, Go 的框架成熟度和 Java 也不在一个级别, 但相信这些都会随着时间逐步成熟。

总之, 对于这个框架压测数据意义在于了解性能天花板, 判断继续优化的空间和ROI (投入产出比)。具体选择框架还是要根据使用场景, 性能, 易用性, 可扩展性, 稳定性以及公司内部生态等作出选择, 语言和性能分别只是其中一个因素。

各种框架的应用场景不同导致其优化侧重点不同, 如 spring web 为了易用性, 可扩展性和稳定性而牺牲了性能, 但它同样拥有庞大的社区和用户。再比如 Service Mesh Sidecar 场景下 Go 的天然并发编程上的优势, 以及小内存占用, 快速启动, 编译型语言等特点使得比 Java 更加适合。

(附: 其实我使用上述代码和 dockerfile 构建, 并且使用同样的压测脚本, 在阿里云4核独享机器测试下 go fasthttp-easyjson-prefork 框架 Json serialization 的性能要高于 Java wizzardo-http 和 firenio-http-lite 30% 以上且延迟更低的, 这可能和内核有关)。

算法篇

• SIGIR2021 | 超越I2I和向量内积，淘宝新一代召回范式：PDN模型

- 作者：济贤、仙基、肖荣
- 出品：淘系技术

摘要

一般来说业务的推荐系统的常用的召回算法有两个范式，相似度索引范式(如I2I)，EBR范式（如Deep-Match）。I2I范式缺点在于对共现少的pair难以泛化，难以建模U2I部分，从而模型缺乏准确和个性化。EBR范式虽建模了U2I部分，将用户的兴趣整合成了一个向量。但却无法建模用户每一个行为和打分item之间的关系（类似于Target Attention），从而召回即缺乏多样性。为了融合两者的优点，尽可能的减少两者的缺点，我们提出了一种新的范式Path based Deep Network（PDN）。PDN模型用TriggerNet建模U2I的部分，SimNet建模I2I的部分，进而端到端的建模U2I2I。目前PDN模型已经在手淘首页内容信息流场景上全量，成为线上最主要的召回源，带来了20%左右的点击个数、GMV、多样性的提升。同时PDN也被SIGIR2021高分录取。

背景

推荐技术在淘宝中的应用是十分重要和普遍的，目的在于建立一个桥梁使得用户可以直达他们感兴趣的物品以提高用户的体验及效益转换。一般的推荐系统主要包含召回，粗排，精排和重排四大环节。由于召回环节处在整条推荐链路的最底层，决定了推荐效果的瓶颈及上限，因此本次工作主要针对有好货场景的该环节进行优化。召回环节的主要任务是高效地从整个商品池中筛选出一小部分（一般来说是千~十万级）用户可能感兴趣的物品供其他环节进行筛选和排序。工业界的召回链路大致包含两类算法：相关索引召回范式，向量化召回范式（Embedding Based Retrieval，EBR）。

目前工业界，相关索引召回以Item2Item范式为主。具体做法是：Step1、离线阶段，基于一些物品相似度衡量指标（如皮尔逊相关系数）去构建倒排索引表；Step2、服务阶段，利用用户的历史行为序列直接查表进行检索。

Item2Item范式的优势在于：

- 1、可以保证用户兴趣的相关性；
- 2、行为丰富的用户召回也是多样的；
- 3、可以捕捉用户的实时兴趣。

但是存在以下四点问题：

- 1、往往I2I的索引是基于一种共现的统计，可能出现冷门商品排不上，新品排不了的问题；
- 2、如何即考虑I2I的共现信息，又考虑Item两端的Side Info；
- 3、如何将这种索引的建立和多样的业务目标关联；
- 4、如何考虑多个Trigger指向相同的一个Item的联合概率。

向量化召回模型（EBR）可以利用Side Info，也试图去建模用户多个行为的联合概率，因此近年受到更多的关注。简单来说，该算法分别得到用户表示和商品表示后，在服务的时候利用近邻搜索实现召回。当然，这类算法也存在不足，主要有两点，一个是这类算法仅用一个或若干个（类似于MIND）向量对用户进行表示，无法像i2i那样，逐商品细粒度的表示用户的多维兴趣；另一个是由于商品端和用户端是并行架构，难以引入目标商品与交互过商品的共现信息。

总体来说，由于受到现有召回模型框架的约束，双塔模型采用了用户信息和商品Profile信息，却无法显式地利用商品共现信息。I2I索引主要利用采用了商品共现信息，但是忽略了用户和商品Profile信息，且无法考虑行为序列对目标商品的综合影响。同时，由于相似度计算方法有所不同，线上往往有多种I2I索引同时工作，我们希望找到一种方法能统一这种I2I相似度，并且尽可能的解决上述提到的四点问题。

摘要

如图1所示，我们将推荐问题归纳为基于二度图的链式预测问题：

$$\hat{y}_{ui} = f(\mathbf{z}_u, \mathbf{x}_i, \{\mathbf{x}_j\}, \{\mathbf{a}_{uj}\}, \{\mathbf{c}_{ji}\}) \text{ with } j \in N(u)$$

其中， $N(u)$ 表示用户交互过的商品集合。现有的大多数工作，包括i2i的协同过滤方法和双塔模型都能看作上述公式的特例。例如，基于回归的商品协同滤波方法可以被定义为：

$$\hat{y}_{ui} = f(\{\mathbf{a}_{uj}\}, \{\mathbf{c}_{ji}\}) = \sum_{j \in N(u)} f_r(\mathbf{a}_{uj}) \mathbf{c}_{ji}$$

其中，定义为预测用户对交互商品感兴趣程度的函数， $\mathbf{c}_{ji} \in \mathbb{R}^1$ 表示交互商品与目标商品的相关性程度。因此，该方法可以看作对n条二跳路径的求和，每条路径的权重为 $f_r(\mathbf{a}_{uj}) \mathbf{c}_{ji}$ 。此外基于向量召回的方法，例如MF，可以被定义为：

$$\hat{y}_{ui} = f(\mathbf{z}_u, \mathbf{x}_i, \{\mathbf{x}_j\}) = \mathbf{q}_i (\mathbf{p}_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} \mathbf{p}_j)^T$$

其中， q_i 、 p_u 、 p_j 分别表示目标商品，用户信息和交互商品的特征向量。MF可以看作是对二度图的 $n+1$ 条路径进行求和，具体来说， q_i 、 p_u 表示直接路径的权重， $1/\sqrt{|N(u)|} \cdot q_i p_j$ 表示二跳路径的权重。同样的，MF的深度化版本YoutubeDNN可以被定义为：

$$\hat{y}_{ui} = f(z_u, x_i, \{x_j\}) = q_i \left(MLP(p_u, \frac{1}{|N(u)|} \sum_{j \in N(u)} p_j) \right)^T$$

已有召回方法受到召回效率、模型结构的限制，难以使用到图中的所有信息。例如，I2I范式缺少了用户信息和商品信息，EBR范式没有显式地建模商品共现信息。因此，我们提出了一种新型框架path-based deep network (PDN)，来合理使用所有信息以实现低时延的个性化用户多峰兴趣召回。

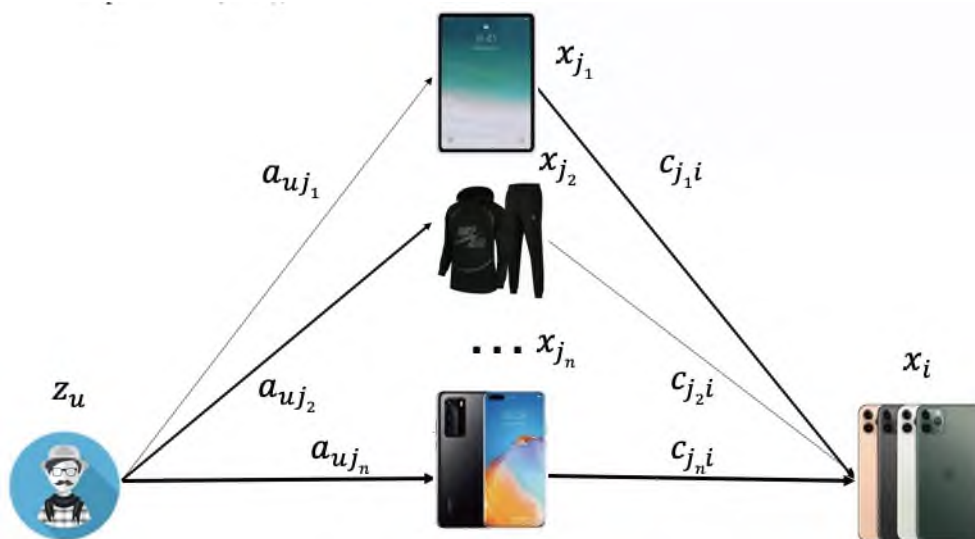


图1: 将用户对目标商品的喜爱程度解耦成二度图

其中第一跳表示用户对交互商品的喜爱程度，第二跳表示交互商品与目标商品的相似程度。其中， z_u 表示用户 u 的用户信息（id，性别等）， $\{x_{j_k}\}_{k=1}^n$ 表示用户交互过的 n 个商品的商品信息（id，类目等）， x_i 表示目标商品的商品信息， $\{a_{uj_k}\}_{k=1}^n$ 表示用户对第 k 个交互商品的行为信息（停留时长，购买次数等）， $\{c_{j_k i}\}_{k=1}^n$ 表示第 k 个交互商品和目标商品的相关性信息（共现次数等），边的粗细表示该条边的权重大小。

方法

为了保证召回的时候能结合个性化的用户细粒度多峰兴趣，我们基于如图1所示的二度图构建了新一代召回框架。该框架克服了之前框架无法使用所有信息的劣势，并且融入了i2i和双塔模型各自的优势实现统一的优化。

其中，图1包含 n （历史行为序列长度）条二跳路径及1条直接路径（user&position bias）。二跳路径中的第一跳表示为用户对交互商品的感兴趣程度，第二跳表示为交互商品与目标商品的相似度，因此，与双塔模型不同的是，我们细粒度的独立建模了用户的多峰兴趣（每个交互商品建立一个兴趣表示路径），解决了单一向量难以表达多维兴趣的问题。直接路径表示了用户对目标商品的直观喜爱程度，例如女生可能对服饰更感兴趣，男生可能

对电子产品更感兴趣。

具体来说，对于n条二跳路径，我们的框架

(1) 基于用户信息，行为信息和交互商品信息，采用一个TriggerNet建模用户对每一个交互过商品的喜爱程度，最终得到一个变长的用户表示向量（维度为 $1 \times n$ ），其中，第k个维度表示用户对第k个交互商品的喜爱程度；

(2) 基于交互商品与目标商品的信息，相关性信息，采用Similarity Net建模交互商品与目标商品的相似度，最终得到一个变长的目标商品表示向量，其中，第k维表示第k个交互商品和目标商品的相似度。最后综合n+1条路径的权重，预测最后对目标商品的喜爱程度。

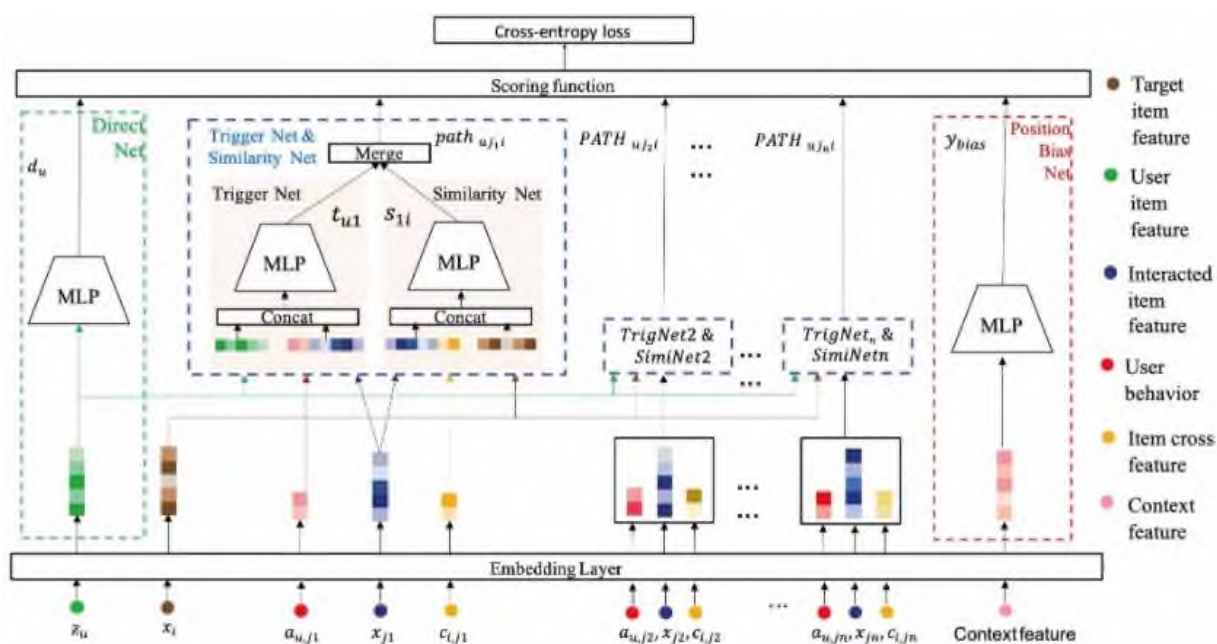


图2 PDN整体框架

PDN整体概述

图2展示了我们所提出的召回框架PDN，主要包含Embedding Layer，Trigger Net (TrigNet)，Similarity Net (SimNet)，Direct & Bias Net 四个模块。PDN的前向过程可以概括为：

$$\hat{y}_{ui} = AGG\left(f_d(z_u, pos), \{PATH_{uji}\}\right) \text{ with } j \in N(u)$$

$$PATH_{uji} = MEG\left(TrigNet(z_u, a_{uj}, x_j), SimNet(x_j, c_{ji}, x_i)\right)$$

其中, f_d 表示直接路径权重的计算函数, $PATH_{ij}$ 表示基于交互商品j的二跳路径权重, AGG表示融合n+1条路径权重, 预测用户与目标商品相关性的评分函数。MEG表示融合每条二跳图权重的函数。

为了保证PDN满足召回环节的时延要求, 我们将MEG定义为两个向量的点积或相加, f_d 定义为点积, 因此, PDN可以被形式化的定义为:

$$\hat{y}_{ui} = pbias + ubias + \sum_{j \in N(u)} MEG(TrigNet(\mathbf{z}_u, \mathbf{a}_{uj}, \mathbf{x}_j), SimNet(\mathbf{x}_j, \mathbf{c}_{ji}, \mathbf{x}_i))$$

下面, 我们将详细介绍PDN中的各个模块。

Embedding Layer

如图1所示, PDN主要使用四类特征, 包括用户信息 $\mathbf{z}_u$, 商品信息 $\mathbf{x}$, 行为信息 $\{\mathbf{a}_{ujk}\}_{k=1}^n$ 以及商品相关性信息 $\{\mathbf{c}_{jki}\}_{k=1}^n$ 。PDN通过Embedding Layer将其转化为编码向量:

$$\mathbf{E}(\mathbf{z}_u) \in \mathcal{R}^{1 \times d_u}, \mathbf{E}(\mathbf{x}) \in \mathcal{R}^{1 \times d_i}, \mathbf{E}(\mathbf{a}_{uj}) \in \mathcal{R}^{1 \times d_a}, \mathbf{E}(\mathbf{c}_{ji}) \in \mathcal{R}^{1 \times d_c},$$

其中 d_u, d_i, d_a, d_c 表示各类特征的维度。

Trigger Net & Similarity Net

经过编码层后, PDN计算用户与目标商品间的每条二跳路径。对于第一跳, PDN利用TrigNet计算用户对每个交互商品的喜爱程度来补获用户的多峰兴趣。具体来说, 给定用户u及他的交互商品j, 计算方式如下所示:

$$t_{uj} = TrigNet(\mathbf{z}_u, \mathbf{a}_{uj}, \mathbf{x}_j) = MLP(CAT(\mathbf{E}(\mathbf{z}_u), \mathbf{E}(\mathbf{a}_{uj}), \mathbf{E}(\mathbf{x}_j)))$$

其中

$$CAT(\mathbf{E}(\mathbf{z}_u), \mathbf{E}(\mathbf{a}_{uj}), \mathbf{E}(\mathbf{x}_j)) \in \mathcal{R}^{1 \times (d_u + d_a + d_i)}$$

表示concatenation操作, t_{uj} 表示u对j的喜爱程度。当用户有n个交互商品时, $T_u = [t_{u1}, t_{u2}, \dots, t_{un}]$ 可以被看作一个变长的用户表示。双塔模型往往以一个或多个定长的向量对用户进行表示, 这被认为是捕获多兴趣的瓶颈, 因为多兴趣信息被没有约束地混合到几个向量里, 导致召回的不准确。相较于这类方法, T_u 能更加细粒度地刻画用户的多峰兴趣, 且更具可解释性, 因为向量中的每一个维度显式传递了用户的感兴趣程度。

对于第二跳, SimNet基于商品信息和共现信息计算交互商品与目标商品的相似度:

$$s_{ji} = SimNet(\mathbf{x}_j, \mathbf{c}_{ji}, \mathbf{x}_i) = MLP(CAT(\mathbf{E}(\mathbf{x}_j), \mathbf{E}(\mathbf{c}_{ji}), \mathbf{E}(\mathbf{x}_i)))$$

其中, s_{ji} 表示商品j和i的相似度, $S_i = [s_{1i}, s_{2i}, \dots, s_{ni}]$ 可以被看作目标商品的变长向量表示。值得强调的是, SimNet显式的学习了商品间的相似度, 因此, 它可以实现线上的独立部署来代替原有的i2i策略。得到 t_{uj} 和 s_{ji} 后, PDN计算得到每条二跳路径的相关性权重:

$$PATH_{uji} = MEG(t_{uj}, s_{ji}) = \ln(1 + e^{t_{uj}} e^{s_{ji}}) = \text{softplus}(t_{uj} + s_{ji})$$

Direct & Bias Net

位置偏差等选择性偏差被证明是推荐系统中的重要影响因素。例如, 一个用户更倾向于去点击靠近顶部的商品, 即使它不是最相关的商品。为了消除该类偏差, 我们基于会导致选择性偏差的特征(位置信息等)训练了一个浅层塔。如图2所示, 训练期间, Bias Net的输出 y_{bias} 被添加到主模型的输出中。而在服务的时候, Bias Net被移除以保证无偏差的打分。Direct Net是类似的, 主要建模user bias, 我们将这两部独立出来主要是为了让TrigNet、SimNet学习出来的东西是和用户、position无关的。

损失函数

用户是否会点击该商品可以被看作是二分类任务。因此, PDN融合了n+1条路径的权重以及偏差得分得到用户与商品的相关性得分, 并将其转化为点击概率:

$$\hat{y}_{u,i} = \sum_{j=1}^n \ln(1 + e^{t_{uj}} e^{s_{ji}}) + \text{softplus}(y_{bias})$$

$$p_{u,i} = 1 - \exp(-\hat{y}_{u,i})$$

由于softplus的引入, 导致 $\hat{y}_{u,i} \in [0, +\infty]$, 因此, 我们利用 $1 - \exp()$ 将预测值投影到0到1之间。我们采取交叉熵损失训练该模型:

$$l_{u,i} = -(y_{u,i} \log(p_{u,i}) + (1 - y_{u,i}) \log(1 - p_{u,i}))$$

其中, $y_{u,i}$ 为样本标签。

约束学习

为了确保模型收敛到更优的区域, 我们精心设计了二跳路径上的约束形式。正如上面所提到的, 在TrigNet和SimNet的最后一层, 我们利用 $\exp()$ 代替其他激活函数来约束输出为正, 即 $e^{s_{ji}}$ 和 $e^{t_{uj}}$ 。

如果允许负权重的输出, 导致PDN在更宽泛的参数空间中搜索局部最优值, 这很容易导致过度拟合。由于我们在真实使用的时候, SimNet是用于生成Index, 而TrigNet是用于Trigger Selection。这种过拟合的后果可不是效果差一些, 而很可能导致学习出来的索引不可用。

如果允许负权重的输出，导致PDN在更宽泛的参数空间中搜索局部最优值，这很容易导致过度拟合。由于我们在真实使用的时候，SimNet是用于生成Index，而TrigNet是用于Trigger Selection。这种过拟合的后果可不是效果差一些，而很可能导致学习出来的索引不可用。

我们通过允许相关权重为负的例子，来说明不带约束的学习可能出现的问题。

第一个例子，如图3左所示，某个用户点击过Ipad和华为P40pro，有一个负样本的Iphone出现，假设商品相似度都学对了，但Trigger这个部分过拟合了。这个负样本的出现可能是表示用户在这个类目上的兴趣已经消费完了，我们希望通过这个负例让模型捕捉到这个信息，因此是希望模型能学出两个较小的Trigger Weight。但如图中一正一负的情况，也是一种次优解，此时Loss比较小，优化器可能落入这个陷阱出不来。

第二个例子，如图3右所示。某个用户点击过耐克套装和特仑苏，有一个负样本的Iphone出现，假设Trigger Weight学对了，但相似度这个部分过拟合了。此时 $0.8 * -0.8 + 0.5 * 0.5 = -0.11$ ，产生的Loss非常低。却学出了特仑苏和Iphone之间的相似度。如果约束为正时，优化器就会这两个相似度尽可能往0压，从而避免一正一负这种过拟合的情况。



图3 当二跳路径权重为负时的bad case

线上使用

基于路径的检索

为了满足召回环节的时延要求，如图4所示，我们构造了一种新的基于贪心策略的召回链路：基于路径的检索（path retrieval）。具体来说，我们将路径检索解耦为两部分：

- （1）利用TrigNet检索出用户最感兴趣的top-m个交互商品；
- （2）利用SimNet构建的商品相似度索引分别对top-m中的每个交互商品实现i2i检索。

对于TrigNet，我们将其部署为实时的线上服务用于对每个交互商品打分；对于SimNet，我们基于其所计算的商品相似度离线构造倒排索引表。线上召回环节的步骤可以总结如下：

- 索引生成：基于SimNet，我们为商品池中的每一个商品选取k个最相关的商品构建索引，并存储相关性得分 s_{ji} 。详细生成方法见4.2。
- 交互商品提取：当用户进入场景，我们使用TrigNet为用户所有交互过的商品进行打分 t_{uj} ，并返回top-m个交互商品。
- Top-K检索：我们基于top-m个交互商品查询SimNet构建的索引表。并基于如下公式对 $m \times k$ 个候选商品进行得分并返回最后召回结果。

$$\hat{s}_{u,i} = \sum_{j=1}^m \text{softplus}(t_{uj} + s_{ji})$$

此时不再需要Position Bias和User Bias。整体召回框架如图5所示。

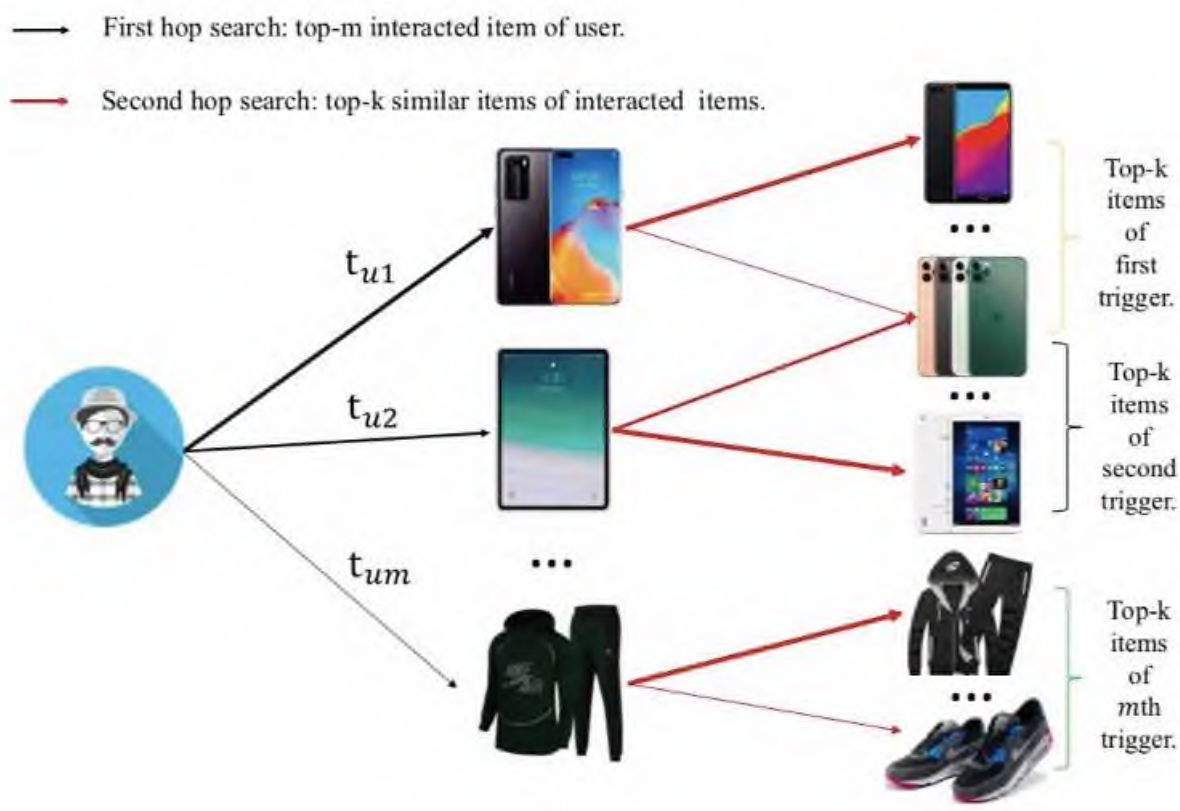


图4 基于路径的检索 (path retrieval)

索引生成

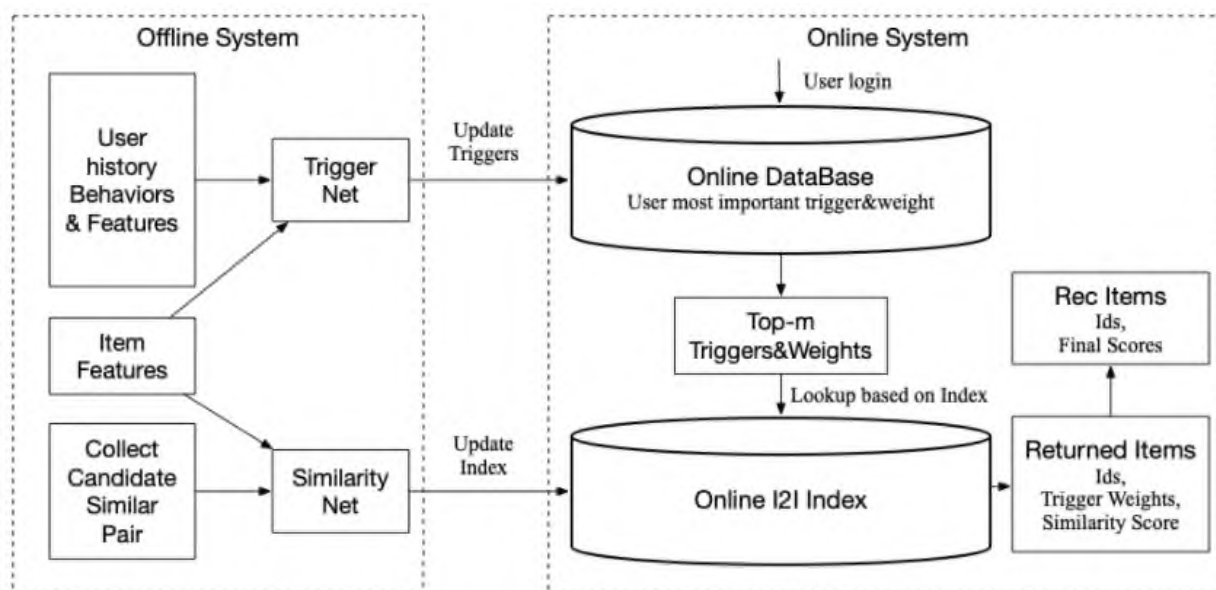
由于商品池很大，我们需要压缩相似度矩阵 $R^{N \times N} \rightarrow R^{N \times k}$ ，以保证离线的计算效率和存储资源。具体包含三个步骤。

步骤一，候选商品对枚举：我们主要基于两个策略生成候选pair对，一个是同一个Session中共现过的商品，另一个是基于商品的信息，例如同品牌/同店铺的商品。

步骤二，候选对排序：利用SimNet对每一个pair对进行打分。

步骤三，索引构建：对每个商品，基于simNet的得分按照某种规则进行排序截断，构造 $N \times k$ 的索引表。

由于SimNet的输入除了共现信息，还有两端的side info，我们可以解决新品召回的问题。具体的做法是，在步骤一的时候，多枚举出一些和新品在商品属性的角度很相似度商品对。



实验

离线验证

表1给出了I2I召回方法的离线验证，用的是线上曝光点击日志。离线召回方法为，利用用户3天内的所有行为作为Trigger，用不同的四种索引，每一个Trigger在索引中找TopN(3/8)个。

这种验证方法是所用Trigger都用上的，不同用户召回个数是不一样的。考虑到有些相似度倒排索引，在某些Trigger下，不一定满3个或者8个，特地加上了Precision这个指标。其中RankI2I为GBDT模型，使用了Swing I2I的作为特征，也使用了item的profile。

PDN模型的SimNet侧使用的特征和RankI2I大体一致，训练目标也一致，均是有好货一跳页的CTR。PDNV13和PDNV43的区别在于，V43采用了带约束学习。

表1 基于有好货曝光日志的离线Hitrate对比

	Swing I2I	RankI2I	PDNV13	PDNV43
TOP3 Hit-Rate	7.56%	14.55%	12.08%	22.99%
TOP3 Precision	0.08%	0.15%	0.12%	0.23%
TOP8 Hit-Rate	13.09%	20.18%	22.12%	34.68%
TOP8 Precision	0.06%	0.09%	0.09%	0.14%

线上效果

表2是线上AB实验的效果，Baseline是有多路召回组成，分为索引召回部分和向量召回部分。索引召回部分同时用SwingI2I、RankI2I、DeepI2I等，向量召回部分有单塔/双塔(Deep Match)/Node2Vec等等。线上实验是将索引召回全部替换为PDNV43（统一了索引，即Trigger保持不变，替换之前所有种I2I索引）。图6是，PDNV43上线之后的线上占比，它将单/双塔召回的比例压缩为6%。几乎吃掉了全部的算法召回部分的份额。

表2 有好货在线效果对比

Base:RankI2I+Swing	人均点击数	pctr	uctr	dpr	人均停时长	GMV	多样性
PDNV43	+17.55%	+15.07%	+1.50%	+7.81%	+7.56%	+21.25%	+19.12%

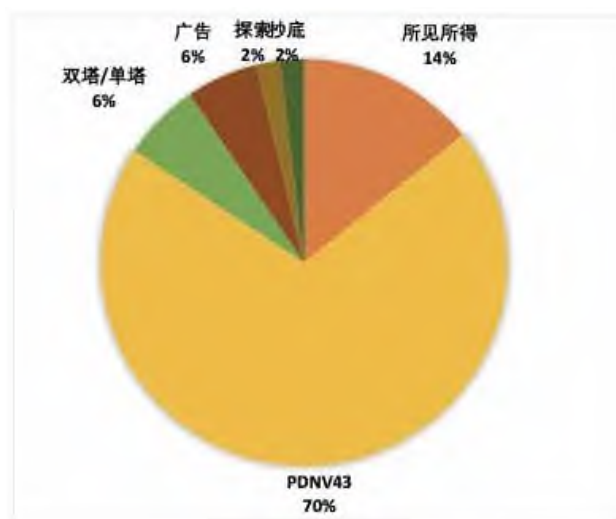


图6 有好货在线召回分路占比

用户Trigger数影响分析

I2I范式的召回，受制于Trigger数的影响。我们特地的对比了多个方法。按照用户的Trigger 数，将用户分为4段：小于等于15的，15~30个的，30~45个的，大于45个的。我们采用两种方法进行验证：一种如表3，是TopN的Hitrate；一种如图7，是TopN用户兴趣的类目覆盖度(Diversity)。

在不同分段上PDN比Swing I2I以及双塔均有较大的提升。值得关注的是，当Trigger数比较少的时候，双塔的Hitrate也低。说明在用户行为不丰富的时候，双塔召回也难以预测用户的兴趣。相对于SwingI2I而言，在Trigger数小于等于45时，hitrate的绝对增幅稳定在20%这样。相对于双塔，hitrate的绝对增幅稳定在15%这样。从多样性的角度，PDNV43相比于双塔，一直维持15~20%绝对值增幅。图7中的BST[3]，也是一种双塔向量召回，只是用户序列那侧用Transformer建模。

表3 基于有好货曝光日志的离线Hitrate对比（按照Trigger进行用户分段）

	Hirate@300 with various trigger number			
	0<n<=15	15<n<=30	30<n<=45	n>45
SWINGI2I	6.59%	11.07%	13.09%	14.13%
双塔（EBR）	12.30%	17.51%	19.06%	20.04%
PDNV43	26.35%	32.61%	32.92%	29.46%

PDNV43对比其他方法的hitrate提升值(+后面为绝对提升 x后面为相对提升)				
PDNV43 Vs. Swing	+20% (× 300%)	+22% (× 195%)	+20%(× 152%)	+15 (× 109%)
PDNV43 Vs. 双塔	+14% (× 114%)	+15% (× 86%)	+14%(× 73%)	+9%(× 47%)

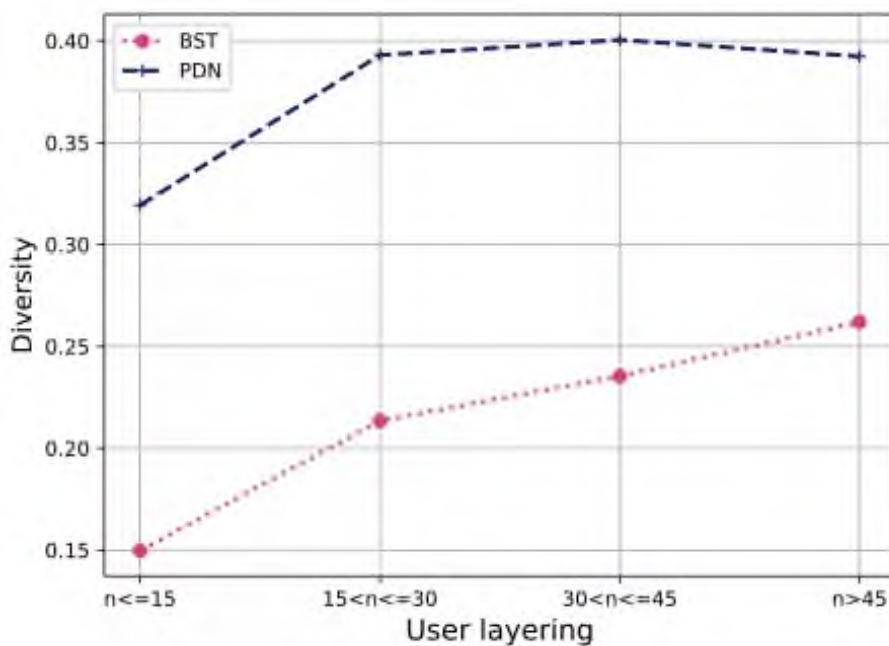


图7 用户按照Trigger数分层与多样性

公开数据数据集对比

同时也在公开数据集上做了实验。在EBR召回范式中，我们选择了DSSM、Youtube DNN、BST三种模型。这三种模型的区别可以理解为：无用户序列、用户序列mean pooling、用户序列过Transformer。在I2I召回范式中，我们选择了传统的Item-CF、和SLIM。为了突出PDN模型本身具备一定的排序能力，我们同时选择DIN模型作为对比之一。DIN模型在算Hitrate的时候，将对所有的候选集都打一遍，算TopN。

表4

Group	Method	Movielens		Pinterest		Amazon book	
		Hitrate@10	NDCG@10	Hitrate@10	NDCG@10	Hitrate@10	NDCG@10
Embedding Based Retrieval	DSSM[1]	0.4699	0.2603	0.273	0.1394	0.6433	0.4163
	Youtube DNN[2]	0.6187	0.3579	0.7585	0.4351	0.6539	0.437
	BST[3]	0.6316	0.3603	0.8176	0.4955	0.6923	0.4403
I2I Retrieval	PCF[4]	0.4033	0.2132	0.2800	0.1470	0.0547	0.0203
	SLIM[5]	0.4400	0.2337	0.6067	0.3323	0.1621	0.0679
Ranking model	DIN[6]	0.6454	0.3767	0.8185	0.5051	0.6973	0.4698
Incorporated	PDN w/o bias net	0.6323	0.3401	0.8015	0.4814	0.6869	0.452
	PDN	0.677	0.4071	0.8283	0.5358	0.7019	0.4735

讨论与展望

模型结构也可以理解为两个长度为item corp size（记作N）的稀疏向量进行内积。具体的，对于某个有k次行为的用户来说，其user representation是一个维度为N的向量，但是这个向量上只有k维有值，其他全部为0。对于某一个有m个相似商品的某商品来说，其item representation也是一个维度为N的向量，但其中只有m维有值，其他都为0。又由于 $k \ll N$ ， $m \ll N$ ，所以这两个向量是稀疏向量。章节3.5中的第一个公式，可以看做两个稀疏向量的内积公式，只不过通过一个ln函数将数值降了下来。相比于将user&item representation固定为64维或者128维来说，整个模型的Capacity是很高的。

PDN可以直接当作排序模型来用。这个方向上我们已经在直播信息流的排序模型中进行尝试，目前已经拿到了离线验证的效果。粗排模型的基础上+PDN，相对于粗排模型预测AUC涨+0.6%，已经接近于精排模型的预测AUC。

	训练AUC	预测AUC
精排模型：Target Attention + mean pooling	72.2	72.8
粗排模型：User侧Mean pooling + 双塔内积	71.1	72.0
粗排模型的基础上+PDN	72.0	72.63

• SIGKDD2021 | 淘宝搜索向量化召回实践

- 作者：福豆
- 出品：淘系技术

摘要

近年来，随着深度学习在图像、自然语言处理等领域的蓬勃发展，越来越多的工业界搜索和推荐系统将大规模深度学习应用到真实的业务中。

一般来说业务的搜索系统由多阶段（召回-粗排-精排-重排等）组成，而召回决定了搜索性能的上限。

区别于网页搜索，电商搜索的召回除需要满足用户基本的搜索相关性需求外，还需要尽可能展现给用户对特点品牌、风格、价格等的个性化商品。然而传统的基于倒排索引的召回机制难以将用户个性化行为引入，向量化的深度召回模型则由于其天然的embedded特性逐渐成为重要的个性化召回来源。

但在我们长期实战中发现，虽然个性化向量召回对搜索成交提效有明显帮助，但其召回的商品相关性可控性差，损害了用户搜索体验，也成为其继续提效的瓶颈。因此，我们提出了一种同时兼顾个性化和搜索相关性的向量召回模型，将用户个性化行为和搜索词高效地建模到网络中，使最终学习到的用户向量表达既能满足个性化需求也能保证良好的搜索相关性。

我们的模型替换原有的向量召回模型在淘宝搜索上线，线上A/B测试显示用户搜索体验提升的同时，搜索成交也得到显著提升。同时我们的工作也被SIGKDD Applied Data Science Track录用。

背景

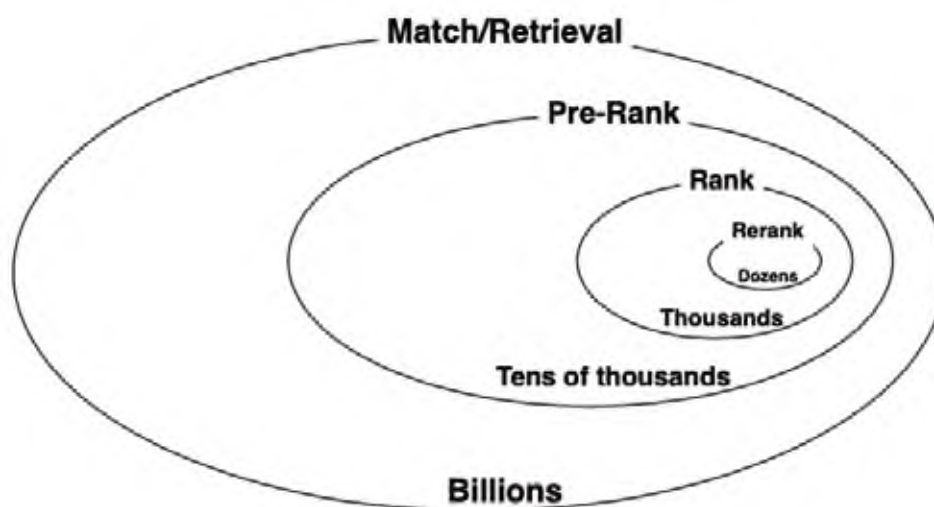
电商平台的商品搜索、推荐服务，从海量候选中高效且准确地为用户查找、过滤和展示商品，已然成为人们生活中重要的购物方式之一。

在电商搜索领域，除了满足推荐系统通常考虑的用户个性化需求外，搜索系统还必须满足用户查询的最基本相关性需求。

例如，当用户的搜索请求为“羽绒服”时，搜索系统既要满足用户个体对品牌、颜色和风格等偏好，还要保证所展示商品的品类都是羽绒服，而不能出现棉服。

从技术角度看，工业界搜索、推荐系统通常采用了多阶段（包括召回、排序等）方法，如下图所示，是淘宝搜索系统的宏观视角，召回处于系统的最下层，首先负责从海量商品池中初步检索出符合用户意图的候选商品，进而送到排序阶段进行精细化的个性化排序，最终展示少量符合用户意图的商品。由于其处于搜索系统的上游流程，通常决定着搜索系统的性能和服务质量的上限。

近年来，随着深度学习在各领域的迅速发展，向量化召回技术EBR（Embedding-Based Retrieval）在工业界搜索（百度、京东、Facebook搜索等）、推荐（淘宝猜你喜欢推荐等）、广告（阿里妈妈）实际应用中同样发挥了重要作用，为用户提供了区别于传统召回视角的商品集候选。然而，在我们落地实战过程中发现，向量化召回虽然通过引入用户历史行为建模和模糊匹配为现有的淘宝搜索召回模块取得了明显的线上收益。但由于向量内积检索相比于传统的倒排索引检索是一种非精确匹配（inexact match）的方式，在搜索相关性控制方面不足，给后链路引入了较多badcase，即使有相关性排序模块，依然对用户搜索带来了负面体验。此外，这些无效召回降低了召回率，无法满足排序模块的需求，降低了系统利用率。因此，只有在商品的检索召回阶段同时建模用户查询的语义相关性以及其个性化偏好，才能为排序阶段提供更好的商品候选集合，最终为用户提供优良的个性化商品搜索服务和在线购物体验。



问题分析

基于项目背景的分析，我们问题的定义可以数学化表示为：

$$O = \max \sum_{i=1}^n GMV(query_i, user_i, item_i)$$
$$s. t. \frac{1}{n} \sum_{i=1}^n relevance(query_i, item_i) \geq requirement$$

我们在使用用户个性化优化模型效率GMV（核心KPI）的同时，需要使得模型召回的商品与搜索query的相关性满足指定阈值（用户体验尽可能好）。建模用户个性化行为比较直接，在搜索、推荐系统中方法也比较常见，我们的线上向量召回base模型也已经将其引入，核心问题则是如何保证效率的同时，提升模型召回结果的相关性，使得相关性和个性化相辅相成。

我们在提出解决方法之前，通过相关工作调研，先分析了个性化向量召回搜索相关性低的原因，主要是以下两点：

exact match能力的缺乏：模型使用低维压缩的embedding进行向量内积检索，因此不能灵活地在词级别上进行文本相关性学习，可解释性和相关性可控性都不够好。

高质量训练样本的缺乏：模型通常使用点击曝光日志来训练，但“点击不等于相关”和“selection bias”的现象普遍存在，导致缺乏适合相关性学习的训练数据成为相关性低的原因之一。

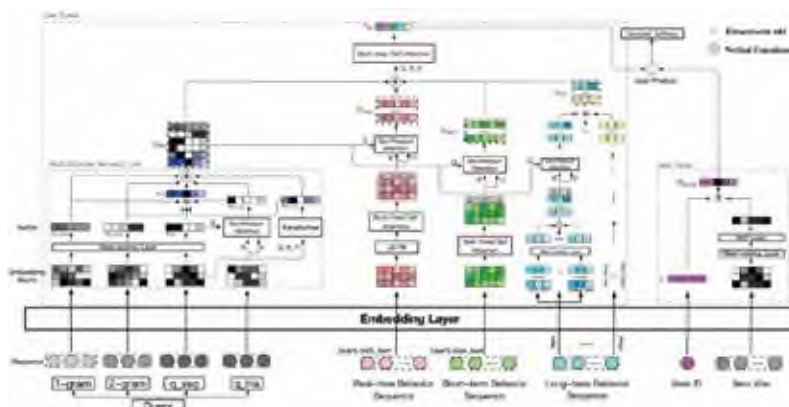
学术界在exact match方向上的解决方法通常是query和item表达做复杂交叉建模，由于工业界召回需要使用向量内积K近邻检索，因此在工业级召回问题上不能适用。我们便将主要精力放在了样本方向上。此外，与文本搜索不同，电商搜索的用户个性化发挥了重要作用，我们通过设计平衡个性化和语义相关性的模型结构，在保证模型相关性的同时，优化模型的GMV效率。

模型设计

我们提出一种新颖的平衡融合搜索语义相关性和个性化建模的深度向量召回模型（MGDSPR），输入用户在淘宝的历史行为和当前搜索query，模型建模得到用户搜索向量表示，与商品向量通过内积进行打分，从而获得检索结果的topK，计算打分公式数学化表示为：

$$z = \mathcal{F}(\phi(q_u, \mathcal{R}^u, \mathcal{S}^u, \mathcal{L}^u), \psi(i))$$

q_u 表示用户搜索词， $\mathcal{R}^u, \mathcal{S}^u, \mathcal{L}^u$ 分别表示用户的实时、近期和长期行为。 $\mathcal{F}(\cdot)$ 表示内积函数， $\phi(\cdot)$ 表示query和行为联合建模函数， $\psi(\cdot)$ 表示商品向量建模函数。建模函数具体结构如图所示。



相关性建模

我们对向量召回模型相关性提升的主要收益来源于样本的改进，根据实战经验总结了多阶段下不同样本的优势。

样本类型	个性化	相关性	语义GAP
曝光点击/未点击样本	✓		
进精排未曝光样本	✓		
进海选未进精排样本	✓		
batch内负样本		✓	✓
全库随机负样本		✓	✓
难负样本		✓	
难正样本	✓		

我们实践过程中尝试了上表中不同样本的利用，比较成功、能够显著提升模型相关性效果的方法主要包括全库随机负样本及其噪音平滑，以及难负样本生成。

训练样本和损失函数

把用户点击或购买的商品视为正样本，并按商品曝光频率从全集合采样商品作为负样本集合，我们发现使用这种全局随机负样本相比于只使用曝光点击和未点击样本，可以大幅提升模型的搜索相关性。为了使模型训练的样本空间与在线服务的样本空间一致，我们使用softmax来表示对某一商品的打分偏好在全局商品集合的概率分布，进而使用交叉熵作为损失函数：

$$\hat{y}_{(i^+|q_u)} = \frac{\exp(\mathcal{F}(H_{qu}, i^+))}{\sum_{i' \in I} \exp(\mathcal{F}(H_{qu}, i'))}$$
$$L(\nabla) = - \sum_{i \in I} y_i \log(\hat{y}_i)$$

由于基于我们的商品集合在千万级别，上述公式中分母巨大，因此将喜好分输入到基于采样的归一化指数（sampled softmax）函数，以近似在全集合上的商品分数分布。

噪音平滑

我们往sampled softmax函数中引入温度参数以平滑弱监督训练数据中存在的相关性差的噪声影响。通过平滑

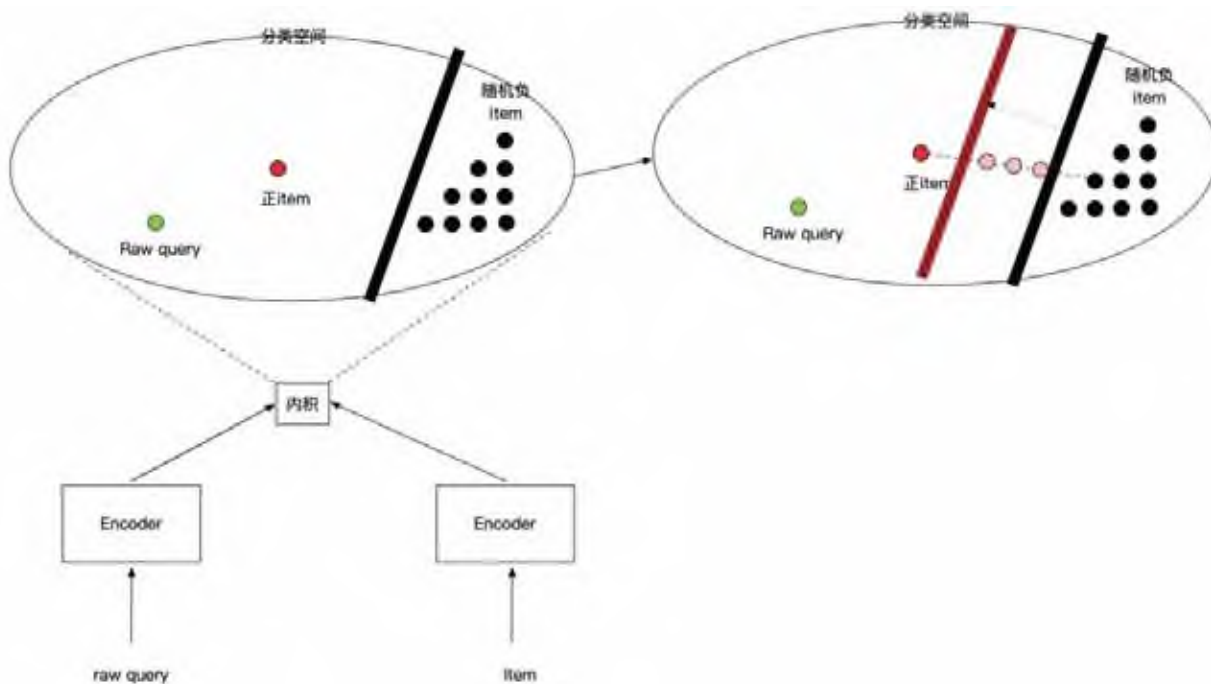
操作可以降低模型对某些item置信度，提升多样性，减少由拟合点击样本带来的相关性差的影响，大幅提升模型召回结果的相关性，具体公式如下。

$$\hat{y}_{(i^+|q_u)} = \frac{\exp(\mathcal{F}(H_{qu}, i^+)/\tau)}{\sum_{i' \in I} \exp(\mathcal{F}(H_{qu}, i')/\tau)}$$

值得注意的是，该思路不仅适用于softmax函数，同时还适用于pairwise函数（通过调整pairwise的margin值，也能提升样本的相关性，并且由我们的实验现象看，随着margin值的增大，相关性评测分数下降）。

难负样本生成

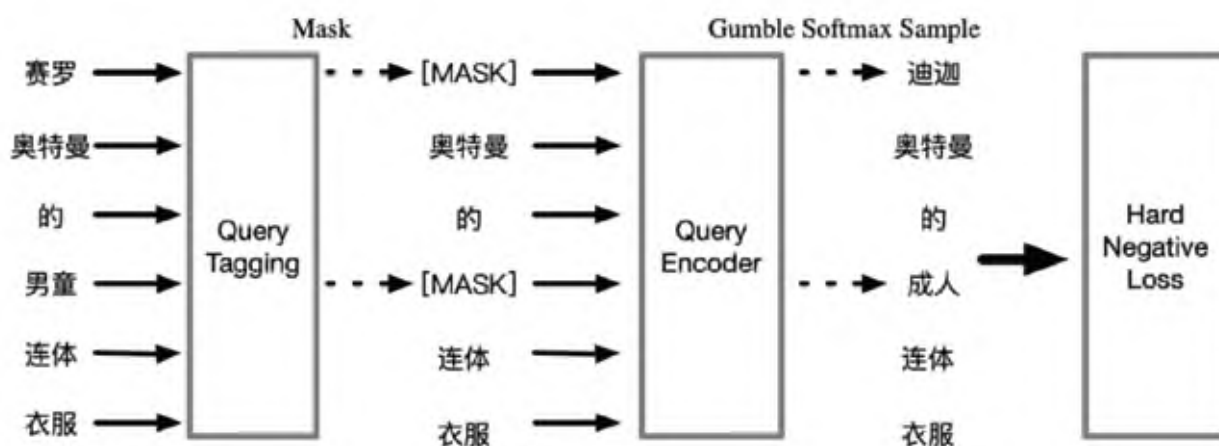
我们尝试了两种难负样本生成的方式，一种是采用在线插值方法，从正样本和负样本中生成相关性难负样本并加入到负样本集合中，接着分别计算用户向量与正样本和负样本集合的内积分数。



首先计算用户query向量与随机采样得到的负样本向量内积的分数，选取top N个距离query向量比较近的负样本放入集合中。通过均匀分布中采样，将正样本向量和选出的负样本向量做插值运算生成难负样本，越接近于1，生成的样本越难。

$$I_{hard} \in \{i^- | rank(score(H_{qu}, i^-)) < Top_N\}$$
$$I_{mix} = \alpha i^+ + (1 - \alpha) I_{hard}$$

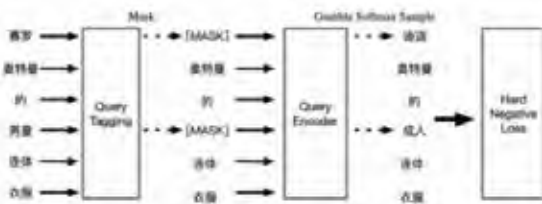
另一种是生成式难负样本，对一个item title，使用querytagging对词粒度打标后，我们将相关性信号重要的词（品类、型号、品牌、风格、人名、书名、电影名等）mask掉，比如“赛罗 奥特曼 的 男童 连体 衣服”转换成“[MASK] 奥特曼 的 [MASK] 连体 衣服”，然后用一个文本生成器输入这个转换后的title，预测被mask掉的词，生成一个新的title，可能为“迪迦 奥特曼 的 成人 衣服”，将新的title作为相关性难负样本放到我们上面提到的改进loss里。这样我们可以使模型从样本角度学习到相关性的exact match和term importance关键词信号。



有了难负样本之后，我们将其加入softmax函数的分母中，继而参与到最后的loss计算中：

$$\hat{y}_{(i^+|q_u)} = \frac{\exp(\mathcal{F}(q_u, i^+)/\tau)}{\sum_{i' \in I \cup I_{mix}} \exp(\mathcal{F}(q_u, i')/\tau)}$$

小结

	baseline模型	本项目工作	简单说明
训练样本	曝光点击/ 未点击样本	曝光点击 全库随机负样本	随机采样代替近似在全库中的生成概率
损失函数	point-wise cross- entropy	sampled-softmax cross-entropy	计算概率公式 $\text{loss} = \frac{1}{ D } \sum_{(x,y) \in D} -\log \frac{\exp(\text{score}(x,y) - \log \eta(y x))}{\exp(\text{score}(x,y) - \log \eta(y x)) + \sum_{x' \neq x} \exp(\text{score}(x,y') - \log \eta(y' x))}$
难负样本	无	在线插值生成难负样本 (可解释性差, 用正样本和随机负样本插值向量)	$I_{hard} \in \{i^- \text{rank}(\text{score}(q_u, i^-)) < Top_N\}$ $I_{mix} = \alpha i^+ + (1 - \alpha) I_{hard}$
		在线生成器生成难负样本 (可解释性强, MASK掉关键词, 生成相似但不相关的难样本)	
噪音处理	无	$f_\theta(q, d) = \frac{f_\theta(q, d)}{\tau}$	对输入softmax函数的logits 用温度T>1平滑噪音处理

语义与个性化建模

多粒度语义单元

从不同层面挖掘用户搜索查询词的语义信息，形成多粒度语义查询矩阵表示。具体地，使用用户搜索词的 uni-gram、bi-gram、分词序列和历史相关搜索词序列组成特征矩阵，使用平均值池化或者transformer来建模向量表达。

比如，给定搜索词“红色连衣裙”，分词序列为

$$q_u = \{w_1, \dots, w_n\} = \{\text{“红色”}, \text{“连衣裙”}\}$$

$$w_1 = \{c_1, \dots, c_m\} = \{\text{“红”}, \text{“色”}\}$$

$$w_2 = \{\text{“连”}, \text{“衣”}, \text{“裙”}\}$$

历史相关搜索词

$$q_{his} = \{q_1^u, \dots, q_k^u\} = \{\text{“绿色”}, \text{“半身裙”}, \text{“黄色”}, \text{“长裙”}\}$$

语义建模向量表达为：

$$\begin{aligned} q_{1\_gram} &= \text{mean\_pooling}(c_1, \dots, c_m) \\ q_{2\_gram} &= \text{mean\_pooling}(c_1 c_2, \dots, c_{m-1} c_m) \\ q_{seg} &= \text{mean\_pooling}(w_1, \dots, w_n) \\ q_{seg\_seq} &= \text{mean\_pooling}(\text{Trm}(w_1, \dots, w_n)) \\ q_{his\_seq} &= \text{softmax}(q_{seg} \cdot (q_{his})^T) q_{his} \\ q_{mix} &= q_{1\_gram} + q_{2\_gram} + q_{seg} + q_{seg\_seq} + q_{his\_seq} \\ Q_{mgs} &= \text{concat}(q_{1\_gram}, q_{2\_gram}, q_{seg}, q_{seg\_seq}, q_{his\_seq}, q_{mix}) \end{aligned}$$

用户个性化异构行为表达

区别于网页搜索，在我们多年的实践中发现，考虑用户在淘宝存在多周期、多类型的历史行为，对于无论是推荐还是搜索、排序还是召回，都有显著的效果增益。在我们模型中，我们主要收集了用户实时行为 $\mathcal{R}^u$ 、近期行为 $\mathcal{S}^u$ （最近100个商品点击）、长期行为 $\mathcal{L}^u$ （一个月内的行为），长期行为还包括用户在商品、类目、店铺、品牌维度的点击、加购、购买、收藏行为序列。使用常用的行为序列建模方法，比如LSTM、transformer等得到相应的行为序列表达向量 $R_{real}, S_{short}, L_{long}$ 。此外与推荐不同的是，用户历史行为存在于当前搜索query完全不相关的情况，为此我们给每一个行为序列向量加了一个0向量来减轻噪音的影响，比如：

$$\mathcal{R}^u \rightarrow \mathcal{R}_{zero\_att}^u = \{0, h_1^u, \dots, h_t^u, \dots, h_T^u\}$$

多粒度语义下用户个性化表示矩阵

计算多粒度语义矩阵与用户历史行为序列之间的重要性加权重，以聚合行为序列中与当前查询词相关的个性化用户向量表示，得到 $H_{real}, H_{short}, H_{long}$ 。比如：

$$H_{real} = \text{softmax}(Q_{mgs} \cdot R_{zero\_att}^T) \cdot R_{zero\_att}^T$$

动态融合语义与个性化行为表达

个性化行为如何更合适地与用户搜索语义意图结合是学术界和工业界的热点话题。这里我们直接参考了现有电商商品搜索个性化相关工作的结论：个性化在越往头部query方向起到正向作用，在长尾方向作用降低甚至会对搜索结果有损。在模型设计上，也参考了这些研究工作的方法，使用自注意力机制融合多粒度语义查询矩阵和用户个性化表示矩阵的关系，得到兼顾用户查询语义相关性及其个性化偏好的用户向量表示。

$$H_{qu} = \text{Self\_Att}^{first}([CLS], Q_{mgs}, H_{real}, H_{short}, H_{long})$$

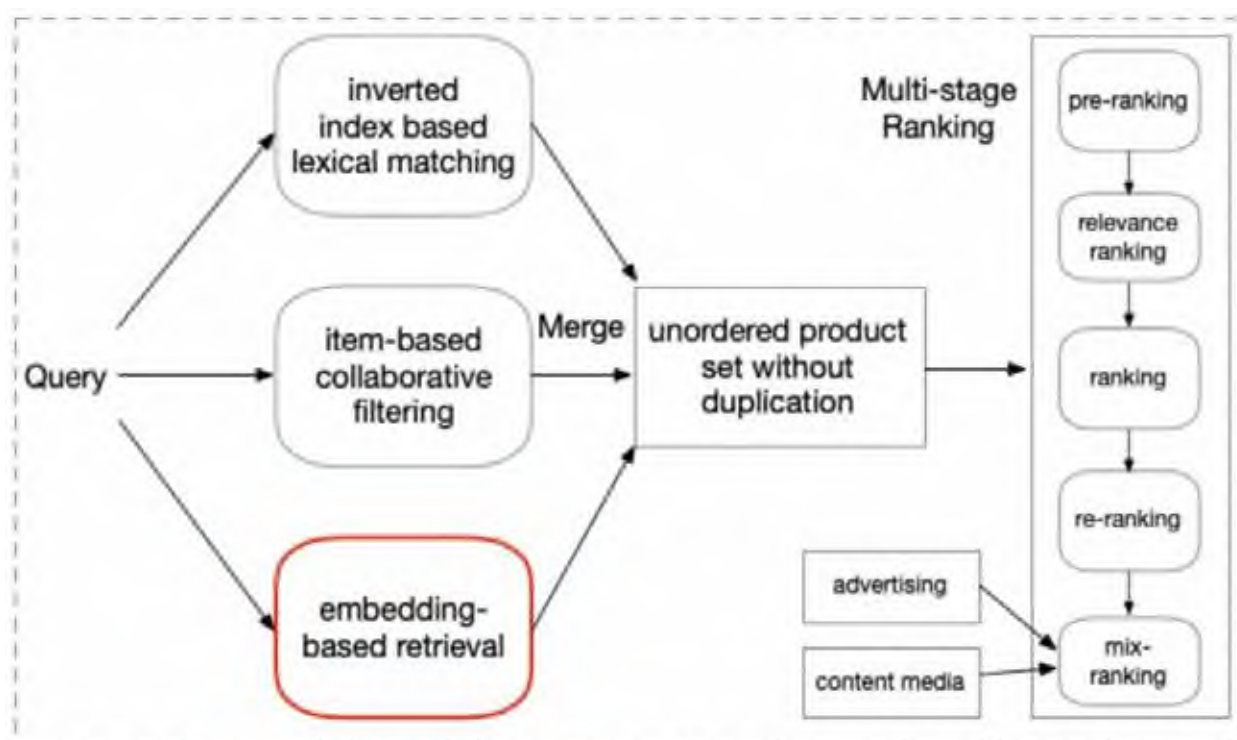
商品建模

商品向量表达 H_{item} 使用了商品ID和标题文本，通过简单的向量相加得到。

$$H_{item} = e + \tanh(W_t \cdot \frac{\sum_{i=1}^N w_i}{N})$$

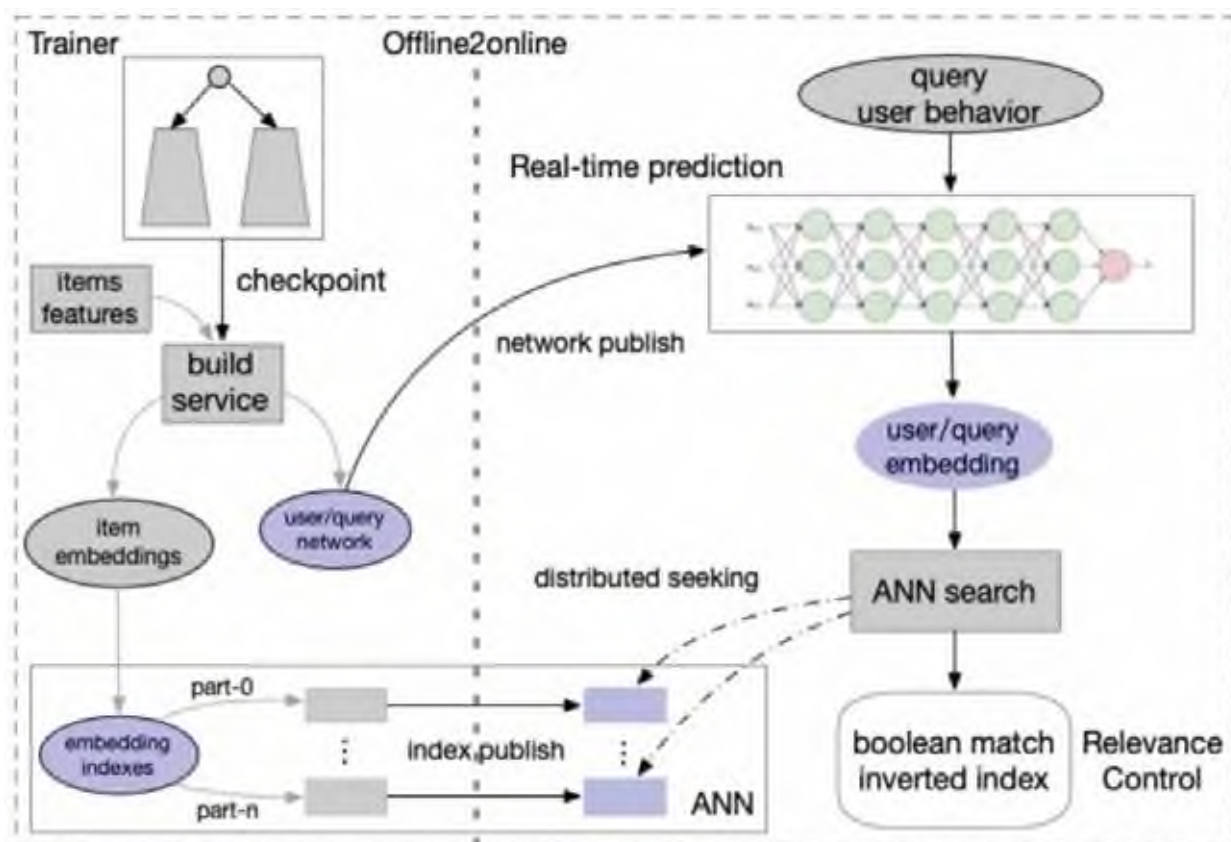
系统架构

线上服务链路，向量召回作为多通道召回的其中一路，加入到已有召回系统上，简略架构图如下图所示。



向量召回具体的线上大规模训练与部署我们采用了offline2online的架构，一次查询在所有商品候选中生效。除向量召回部署通用流程外，我们又增加了一个相关性控制模块，作为对模型相关性增强的补充。该模块结合了文本倒排索引，对向量召回的结果做过滤操作，过滤掉指定的搜索关键属性词。比如用户搜索“阿迪达斯运动鞋”，品牌“阿迪达斯”和品类“鞋”是核心属性词，对不包含这两个词的召回商品做过滤操作，从而高度保障用户搜索体验。

(ANN results) and (Brand: Adidas) and (Category: Shoes)



实验

离线评价

我们收集淘宝搜索的线上日志（点击、成交），来计算模型检索的topK的召回率和相关性。计算曝光点击、成交的商品集合 $T(|T| = N)$ 是否命中模型预测的top K结果来计算 i_i Recall@1000。鉴于召回结果集合的规模，我们使用线上相关性模型打分来替代人工评测，计算召回商品的相关性good率，以此来辅助指导相关性方向的建模工作。而线上A/B实验中，我们则使用了人工评测来衡量相关性体验。

$$Recall@K = \frac{\sum_{i=1}^K i_i \in T}{N}$$

$$P_{good} = \frac{\sum_{i=1}^K \mathbb{I}(i_i)}{K}$$

$\mathbb{I}(i_i)$ 表示商品是否属于当前query的相关商品集合。此外，为了增加评测难度，我们还加入了一份由搜索点击引流到场景外成交的相关商品日志来评测。一共采样第N+1天上上述类型的150多万条user query日志进行评测，使用前N天数据训练模型，模型离线评测训练使用与线上相同的全量数据进行。预测top K首先模拟线上全图化流程离线导出user和item向量，内积检索使用公司内部的Proxima建进行离线向量召回。Proxima离线的使用，我们尽可能与线上向量内积算法的配置保持一致，比如一致的建索引、查询索引参数，索引准确率，是否使用MIPS转换等。

由于淘宝搜索早已使用多路召回，而且每路召回的特点不同，每类query类型的特点也不同，我们又将Recall和相关性分数拆分成在query类型和曝光商品召回来源类型下观测。比如模型在长尾query下Recall提升，说明模型在往长尾方向优化；如果模型Recall在top query下的Recall提升，说明模型在往个性化方向迭代。通过这种方式来辅助我们判断模型优化走向。也正是由于多路召回存在，离线单模型Recall并不能直接反应线上效果，因为可能与已有召回重复。我们在线上做了计算逻辑等价于离线recall的指标——线上Recall，作为中间指标来参照，即计算线上所有向量召回（包括与其他路重叠）点击或成交在所有点击或成交中的占比。根据我们的上线经验，离线Recall增长，必然会使得线上向量召回单模型增强，线上Recall也会增长。由于线上链路对召回有很多其他限制，离线评测环境不能与线上完全保证一致，因此线下Recall数值上也不能与线上完全一致，只能提供给线上的一个upper bound参考。我们一直使用这个指标来判断模型线上是否正常服务，优化是否在线上起生效。线上成交是否增长，还是要看模型优化是否提供了正向的增益，但已有历史数据无法做到这点，目前来看还是需要上线测试。

线上实验

我们替换掉原有的向量召回模型上线，线上多天A/B实验显示，用户展示商品的相关性人工评测good率提升0.3%的条件下，搜索成交笔数和GMV都有显著提升。

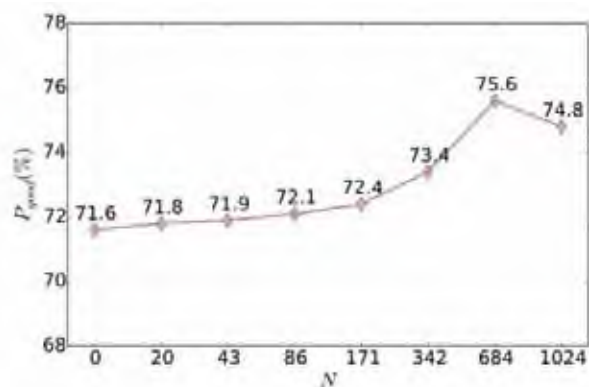
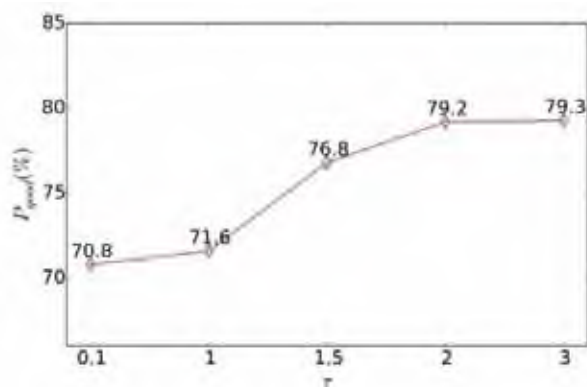
Launched Platform	GMV	#Transactions
Taobao Search on Mobile	+0.77%	+0.33%

消融实验

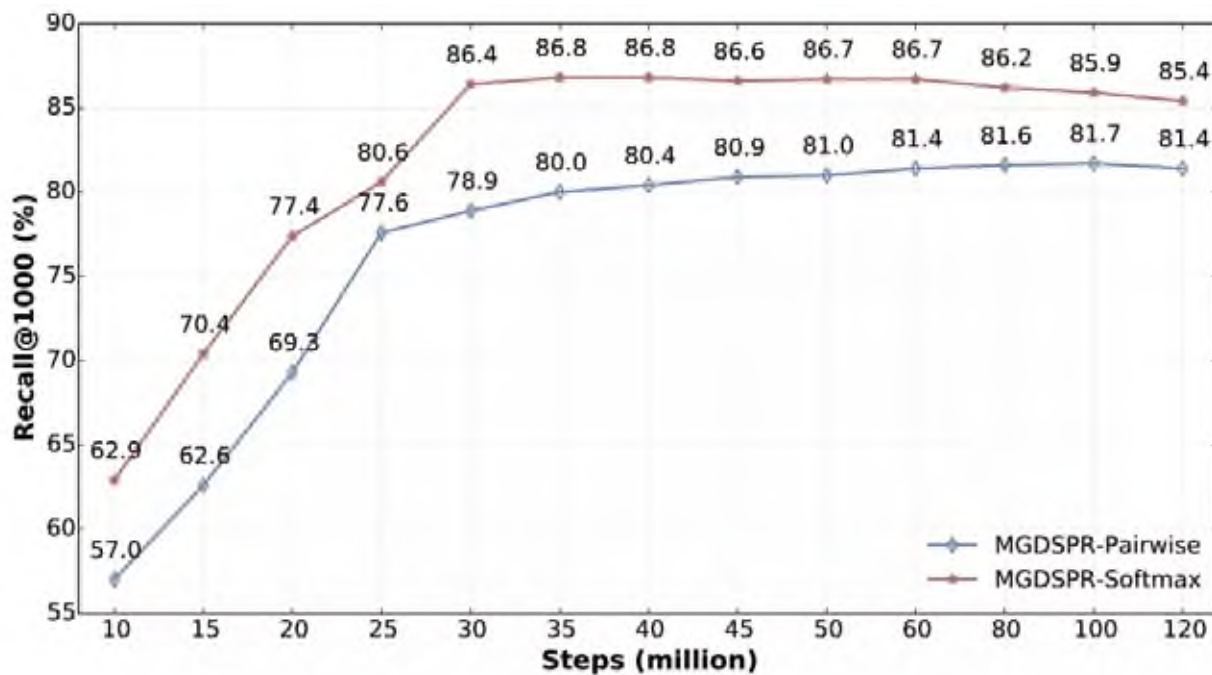
在这里，我们列了模型结构中提出的几个关键部分对召回模型的召回率、相关性的影响。

Methods	Recall@1000	P_{good}
MGDSPR	85.6%	71.2%
MGDSPR + mgs	86.0%	71.6%
MGDSPR + trm	86.4%	71.4%
MGDSPR + τ	85.5%	79.0%
MGDSPR + mgs + trm + τ	86.8%	79.2%
MGDSPR + I_{mix}	83.6%	75.6%
MGDSPR + all	84.7%	80.0%

MGDSPR是基础版本的模型，不含 mgs （多语义单元）、 trm （self-attention动态融合语义和个性化表达）、 τ （softmax温度平滑）和 I_{mix} （难负样本）。可以看出样本的温度平滑和难负样本对优化模型相关性至关重要，我们对具体的超参做了更为细致的实验，如下图所示，温度越高、难负样本的数量 N 一定区间内越大，模型召回结果的相关性就越好。我们在召回率与相关性之间tradeoff，选择综合最优模型上线。



我们还实验了使用softmax相比传统pairwise方式的损失函数的优势，模型召回率更高，收敛更快，加速线上学习迭代。



总结和展望

如何让搜索系统同时兼备用户搜索相关性和个性化的能力，是电商搜索的核心问题之一。往往在真实业务背景下，相关性模块为了保证用户体验，对展示的商品控制严了，就会降低搜索效率（如成交），解决该问题比较好的方法就是在召回源头截断掉bad case，同时不失检索商品的个性化能力。我们的工作在这一方向上迈出了一小步，但仍需深入探索。此外，我们还发现，因为线上召回模块往往是多通道召回，向量召回与其他召回来源的商品存在比较大的交集，解决向量召回与其他通道召回的多样性也是未来亟待解决的问题，需要更加明确向量召回提供的增益方向。

● 情感计算在ugc应用进展

- 作者：淘系内容算法团队
- 出品：淘系技术

观点、情感以及与之相关的许多概念，如评价、态度、感情和心情，与我们主观的感觉和感受密切相关。我们对于现实世界的感知和感受，包括我们做出的任何选择，也很大程度上受到他人对于当前世界的洞察和观点的影响。情感分析伴随着网络社交媒体（如评论、论坛、微博、新闻资讯网站）的兴起而快速发展。这些也被称为用户生成内容（ugc）的社交媒体数据，社交媒体中用户生成内容蕴含了大量的用户观点信息，要从中挖掘有用知识需要对情感计算与观点挖掘问题进行研究。自2000年来年初以来，情感计算已称为自然语言处理领域最活跃的研究问题之一。情感计算系统几乎在每一个企业、卫生机构、政府乃至整个社会都有广泛应用。（下拉滑到留言区，有惊喜）

情感计算任务

根据所处理文本的颗粒度，情感计算研究可以划分为三个级别：篇章级、句子级和属性级。下面进行简要介绍：

篇章级

篇章级情感分析的目标是判断整篇文档表达了褒义倾向还是贬义倾向。如一篇商品评论，我们需要判别出这篇评论对于目标商品总体上持褒义情感倾向还是贬义的情感倾向。这一任务的基本假设是认为一篇文档只对一个实体（如一个商品或者服务）进行了评价。这种假设与实际情况显然是不符的，在现实中，很多文档会同时对多个实体进行了评价或比较，因此需要对目标文档需要做更细粒度的情感分析。因缺乏使用价值，篇章级情感分析目前研究较少。

句子级

句子级情感分析任务是判别一个句子是否包含了褒义、贬义或者中性的情感。这里的中性情感往往意味着没有表达任何观点。

属性级

无论篇章级还是句子级情感分析，都无法确切知道用户到底喜欢或者不喜欢的是什么东西。前面两个级别的分析方法都无法获取观点评价的对象。例如我们仅仅知道“我喜欢iphone12”中包含褒义情感，这对于实际应用来说显然是不够的，我们需要知道用户对iphone12表达了褒义的情感。句子级情感分析假设：如果一个句子中含有褒义观点，则认为用户对于这个句子中所有提到的事物都表达了褒义的观点。

如：apple在这个贫困的国家销售收入表现良好，很难说这句话应该被分成褒义情感还是贬义情感，因为这里表扬了apple但是吐槽了经济。为了得到更细粒度的结果，我们需要细粒度的情感分析，属性级情感分析直接关注的是观点级观点的目标（观点评价对象target），在实际应用中，如评论：虽然服务很差，但我依然很喜欢这家餐厅，观点评价对象（餐厅和服务）通常是一个实体或者实体的属性。所以属性级情感分析的目标是挖掘与发现评论在实体及其属性的观点摘要。在工业界，几乎所有的情感计算系统都是在这个级别上进行分析。

除了按文本颗粒度对情感分析进行划分外，我们还可以根据观点类型的不同划分为常规型观点和比较型观点。

- 常规型观点通常针对一个目标实体或者其属性表达情感倾向。如茶非常好喝，评论者对茶表达了褒义的情感倾向。
- 比较型观点比较多个实体的共同属性，如这家店的服务好于上次去的那家，评论者将这家店和上次的店在服务（属性）上进行的比较。

情感分析的发展也是经历了从规则系统（通过构建情感词典，每个词都有对应的情感极性，根据正向和负向情感词的数量投票及其他规则得到情感倾向）、机器学习系统（svm、gbdt、随机森林等）、深度学习（textcnn、bi_lstm等）、到现在的预训练语言模型continue pretrain+finetune的标准范式sota方法。

淘宝属性级情感分析

在电商领域，淘宝天猫商品积累了海量的商品评论，人们在购买某一产品之前，往往会查看该产品的相关评论，获得其他购买过用户对商品各个维度的反馈，并通过与其他产品的对比来作最终的决策。由于商品评价量十分巨大，用户没有足够的时间和精力浏览全部的评论信息，导致最终的决策带有风险性。因此如果能对整体评价的无结构文本进行结构化分析，得到<商品的属性、情感词、情感极性>三元组并汇总成观点，就能够让用户对整体评价有个大体的印象，同时结合个性化推荐，有助于解决用户这方面的痛点。技术上涉及到属性级情感分析和评论级情感分析，个性化推荐。

消费者经常会在一个评论中针对不同的属性（又称方面）发表观点，而针对不同属性aspect对应的观点（情感词opinion word）情感极性可能不同。如“料子很好，码数合适，但是有色差”包含了三个属性“料子”、“码数”和“色差”，针对“料子”、“码数”表达了褒义的情感，而对“色差”表达了贬义的情感。属性级情感分析可以对上层情感分析任务提供基础，如观点摘要系统，例如用户需要分别查看对某个实体/商品X就某个方面Y的正面和负面评价，如下图淘宝的大家印象。



问题定义

电商情感分析业务的目标是提取出一些买家对商品的描述标签，这些标签尽量能够从各个维度、生动具体的描述商品的购买体验，所以需要兼顾生动性和多样性。生动性的话需要我们能够细致的找到买家在形容的商品特点，而不是一些广泛的概念，比如“材质结实”比“质量不错”更有信息量；多样性是指我们希望在优先的产品空间内，能够尽量多的展示商品各个维度的特征。

结合上述需求，我们将问题定义为Aspect Sentiment Triplet Extraction 的问题，需要抽取<属性词-情感词-情感极性>三元组。

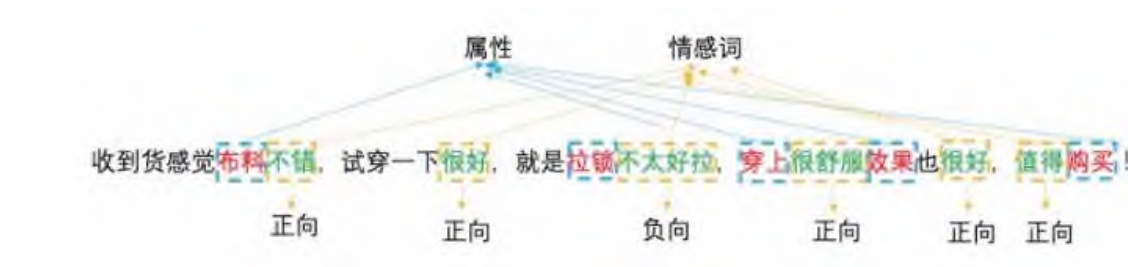
属性词是指

- (1)、商品的一个部件或者一部分，比如笔记本电脑的电池，
- (2)、商品本身的属性，比如价格、材质等，
- (3)、商品的组件或其中一部分的属性，比如电池的寿命。

情感词是用户对属性表达主观情感倾向词语，解释了情感倾向是正/负/中性的原因，比如：电池的寿命长，句子表达的是用户电池寿命的正向情感，情感词是“长”，解释了情感倾向的原因。

情感倾向是指情感词对应的情感极性，我们主要关注了正向、负向、中性三类。

示例如下图所示：



数据的特点及技术上的挑战

不同领域表达的差异性非常大

淘宝有百级别一级核心品类，万级别叶子类目，不同一级核心品类/叶子类目评论的商品属性和情感表达存在非常大的差异性如服饰的属性有面料、款式、版型、风格等属性，美妆的属性保湿、遮瑕、防晒等属性。同时相同的情感表达在不同领域情感极性也会不同，如冰箱声音大是负向，但音响声音大是正向。如何充分利用类目表达的差异性和共性的知识是很大的挑战。

正负向数据严重不均衡

在真实的淘宝评论里因为各种商业利益评论区的正向表达和负向表达也非常不均衡，不同类目达到10:1~20:1，给模型性能带来了非常大的挑战。

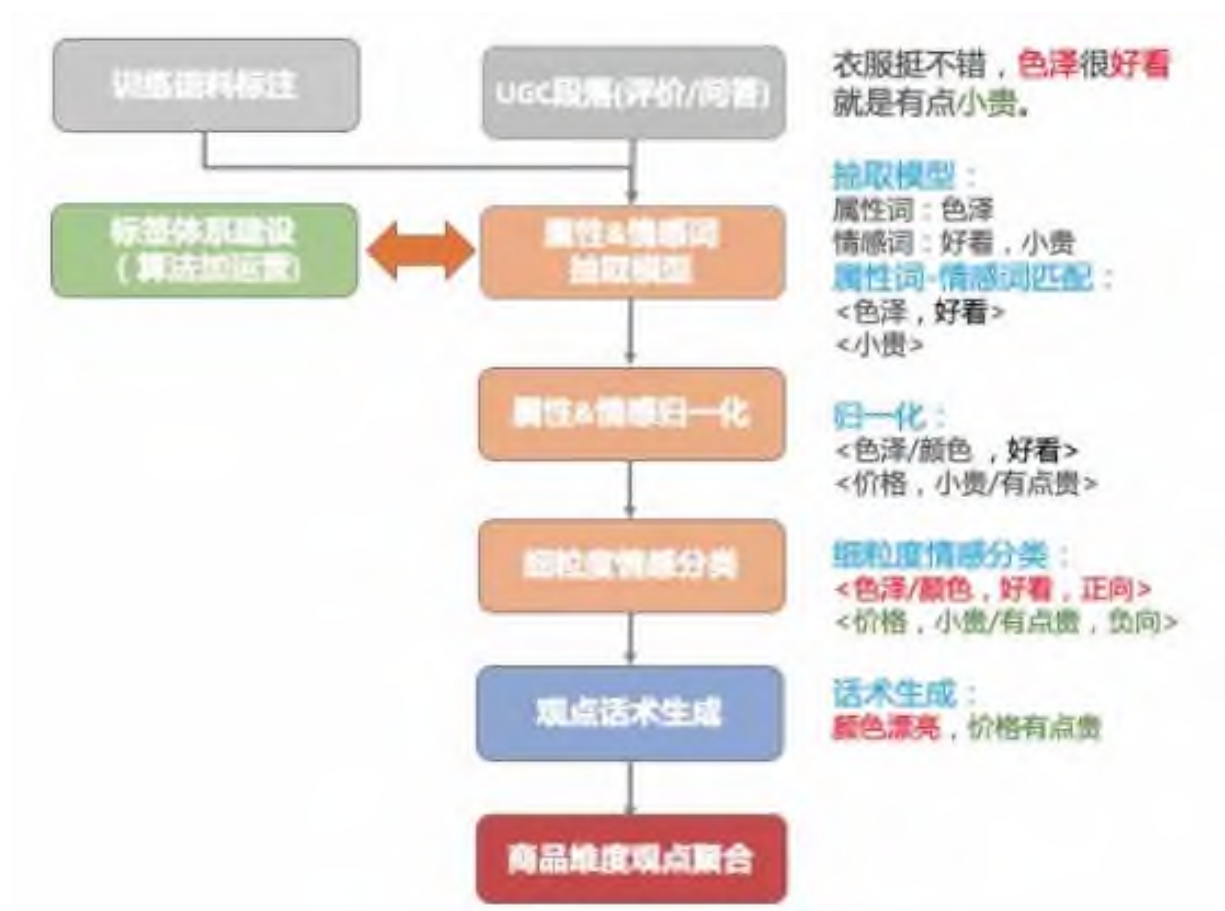
标注数据稀缺

标注数据稀缺，标注任务因为复杂而导致标注成本昂贵，给整个任务都带了很大挑战。

总之，如何解决好不同领域表达的差异性和共性，更高效的利用标注数据和未标注数据是我们的核心痛点，我们在具体的方案设计中也充分考虑这些难点和挑战。

pipeline系统

我们把属性级情感分析拆解为3个子任务，给定一个句子，一个是自动抽取句子中的所有属性和情感词，一个是属性情感词搭配关系，还有就是针对不同的属性判断其情感极性。通过这一套pipeline系统，印象词的CTR从0.9%提升到400%。

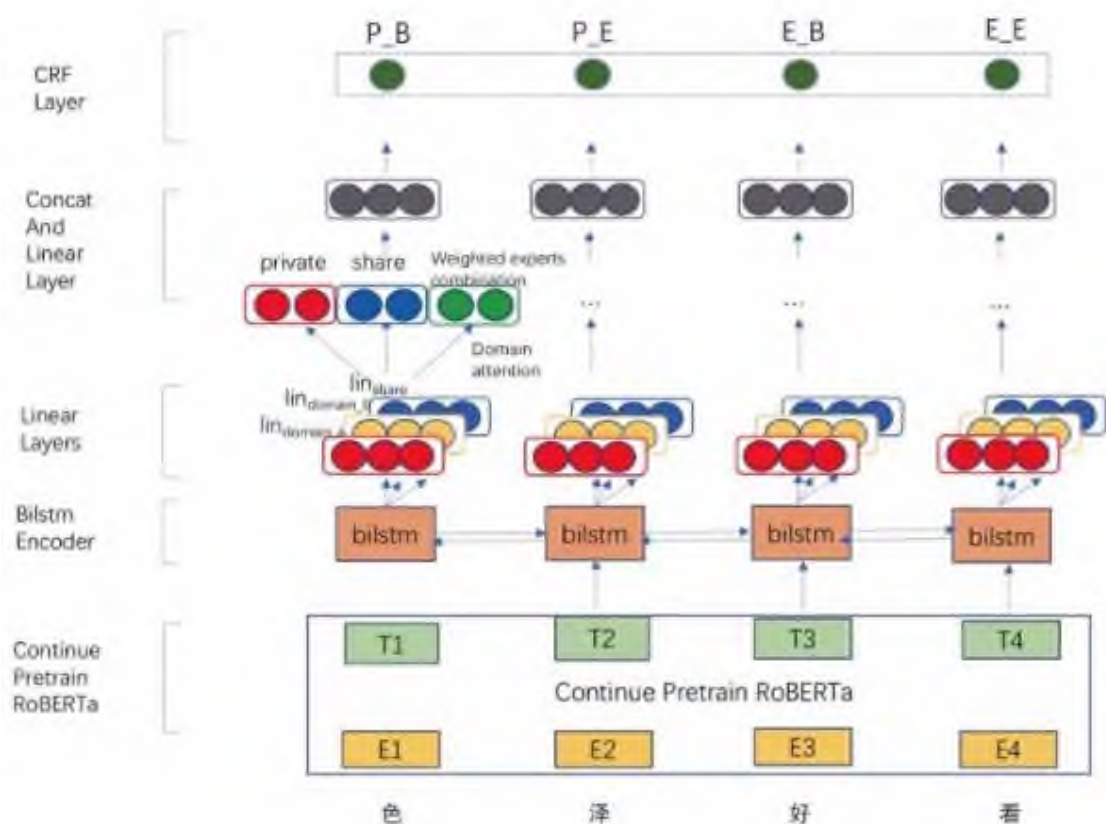


多领域的属性&情感词联合抽取

对无结构化的评价文本，通过建模抽取句子中的属性词和情感词，是一个典型的序列标注的任务。

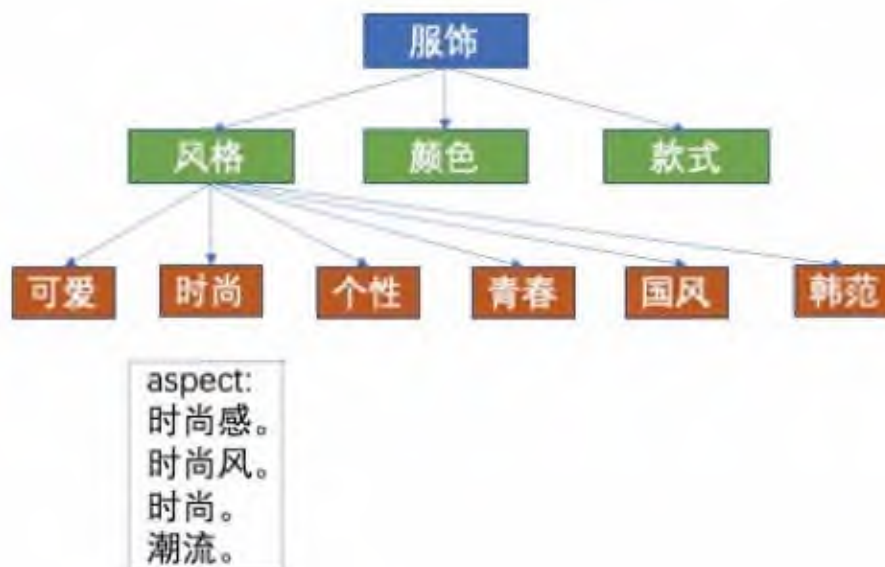
在电商的评价的属性、情感抽取场景，语料包含了几十多个类别，每个类别只有数目有限的标注语料，各个类别之间的属性情感既有相同的部分（比如价格、质量、服务态度等），也有差异很大的地方（比如食品和锅具评价中的味道）。

具体实现：在常用序列标注任务基线模型Bert + Bilstm + CRF的基础上，首先通过在评价语料上的continue pretrain，F1值有0.6%的提升。其次多领域迁移问题进行优化，通过把Istm输出的向量映射到领域共用和私有的空间，对每个领域，都有共享的和私有的两部分表示，来获得领域共性和差异性的表达，同时领域私有的表示会和其他领域表示做attention，最终把三部分表示concat起来，实验叠加bert的continue pretrain，有F1值1.8%的提升。



属性归一化

属性归一化是为了解决同一个属性存在多个相似表达的问题，如时尚感、时尚风、时尚、潮流都是在说时尚这一属性，我们把他们归一化成时尚这一aspect。



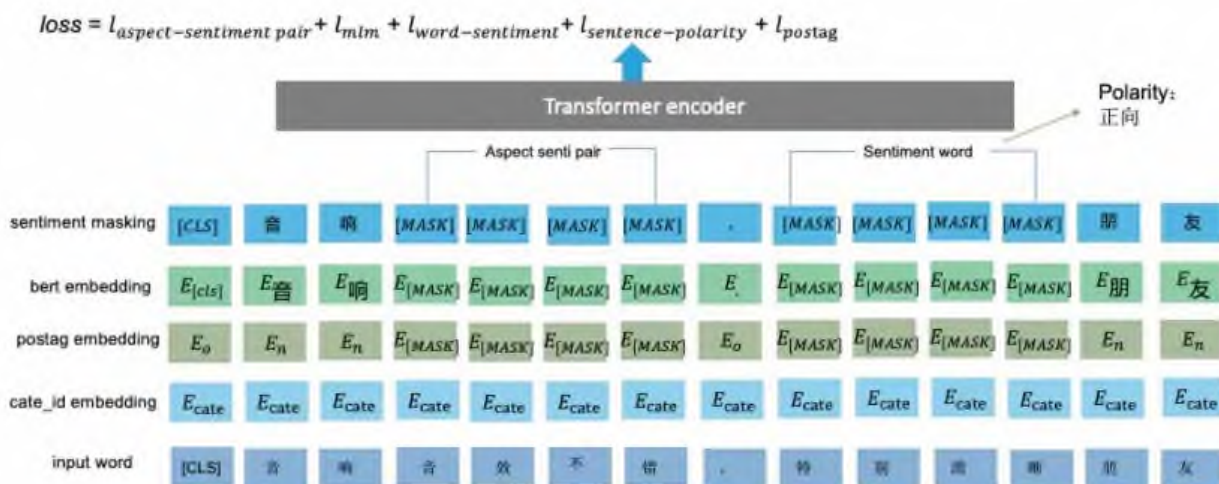
具体的做法是对评论预料进行continue pretrain后，对抽取出来的属性用sentence-bert进行finetune，然后对属性进行分布式表示后通过dbscan对所有的属性进行聚类到归一化属性。

细粒度情感极性分类

预训练在情感计算领域有大量成功案例，在标注语料偏少情况下，对领域数持续预训练能取得很好的效果。在我们场景会面临标注成本高，标注语料少，还有就是真实的评论里正向/负向情感比例分布不均衡问题（不同类目10:1-20:1）导致的负向情感性能偏低，因此做预训练是非常有必要的。电商领域下的任务如atsa, acsa, 句子级情感分类，评论问答等任务中直接依赖和情感表达相关的知识(如属性/情感词)，但原生的bert, roberta预训练仅仅mask了字级别，且有字之间的独立性假设，SpanBert mask了词，但不一定是和领域相关，有实验表明不做phrase级别的mask会导致抽取不完整/错误。

在属性级情感分类中，我们采取了融入知识的roberta continue pretrain+ bert finetune的范式来解决面临的问题：

- 1、获得领域global 表达，提升部分类目少样本性能，
- 2、引入item-aware的知识，缓解不同实体相同情感表述，表达的极性不一致问题，
- 3、针对疑难问题，比如否定表达情感分类效果提升。预训练模型结构如下：



模型融入知识的方式是在embedding层显示的嵌入类目信息，词性信息。mask阶段根据挖掘出来的情感知识库：属性-情感pair，情感词，情感词对应的情感极性，优先mask属性-情感pair，情感词。优化目标上除了mask language model loss还显示地预测aspect-sentiment word pair，情感词，情感词对应的情感极性，词性预测任务。

$$l_{aspect-sentiment\ pair} = -\sum_{i=1}^{l=2} y_s \log(\text{sigmoid}(\bar{x}_1 W_{ap} + b_{ap}))$$

$$l_{word-sentiment} = -\sum_{i=1}^{l=n} m_i y_l \log(\text{softmax}(\bar{x}_l W_{ws} + b_{ws}))$$

$$l_{postag} = -\sum_{i=1}^{l=n} m_i y_l \log(\text{softmax}(\bar{x}_l W_{postag} + b_{postag}))$$

$$l_{sentence-polarity} = \text{softmax}(\bar{x}_1 W + b)$$

$\bar{x}_1$ denotes the output vector of [CLS],
 $\bar{x}_l$ represent the masked sentiment word

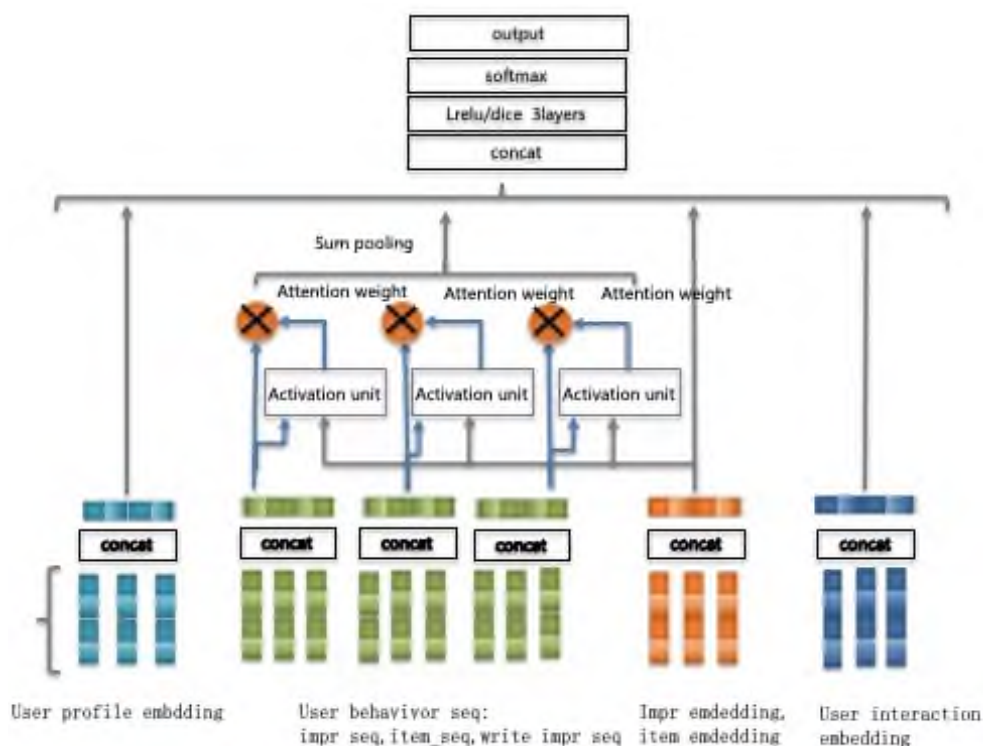
融入知识的情感预训练模型在负向情感f1上提升了4个点。融合各类情感知识预训练的模型效果如下：

Pretained Model	Negative-F1	Positive-F1	Macro-f1
Bert base	0.8677	0.9932	0.9306
Roberta continue pretrain	0.8866	0.9944	0.9405
Roberta continue pretrain + sentiment word mask	0.9013	0.9955	0.9484
Roberta continue pretrain + sentiment word mask + sentence polarity	0.889	0.9935	0.9413
Roberta continue pretrain + sentiment word mask + aspect sentiment pair prediction + sentiment word polarity + sentence polarity	0.9122	0.9945	0.9534
Roberta continue pretrain + sentiment word mask + aspect sentiment pair prediction + sentiment word polarity + sentence polarity + cate_id embedding	0.9126	0.9947	0.9537
Roberta continue pretrain + sentiment word mask + aspect sentiment pair prediction + sentiment word polarity + sentence polarity + cate_id embedding + pos embedding	0.9129	0.9956	0.9543

印象词个性化排序

印象词直接按照按标签出现的频数topk进行展示，有一定的缺陷：挖掘的丰富的印象词因频数低获得曝光机会偏少，不同的用户关注的印象词可能不一样而导致用户需求得不到满足。

排序模型我们使用了经典的排序模型din，通过使用个性化排序算法，CTR +25%，uvctr +20%。模型结构如下：

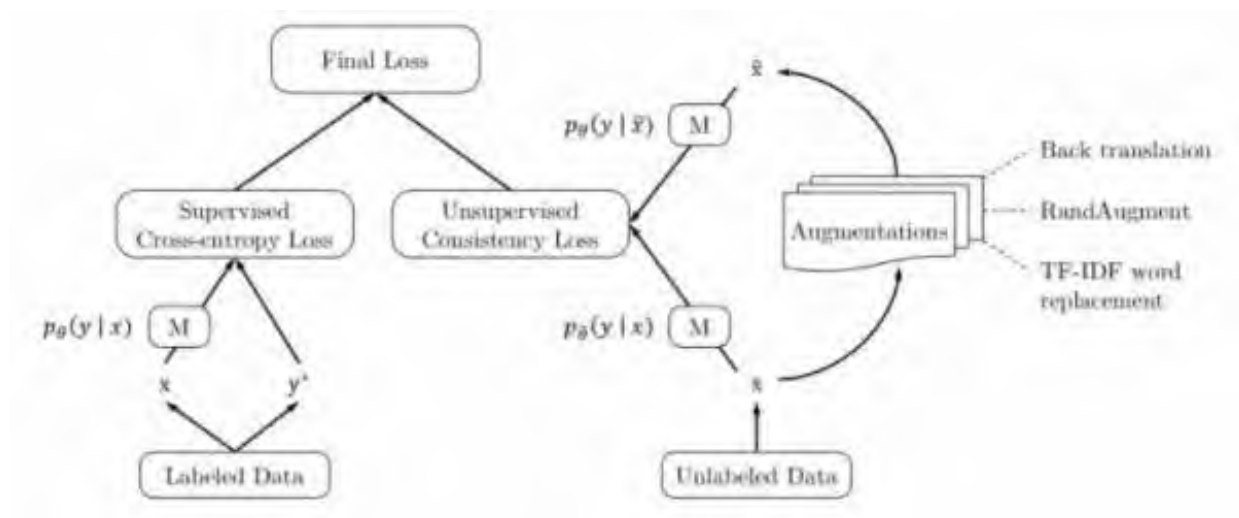


用到的主要特征有用基础的用户画像特征、用户在印象词的一些行为序列、商品侧特征和一些用户写评论时所关注的属性等特征来捕获用户对商品的关注点。

Feature Group	Feature
User Profile	age
	gender
	buyer_power
	...
User Behavior sequence	clicked item sequence
	clicked tags sequence
	...
Item Features	price
	category
	ctr
	cvr
	...
User write reviews	write reviews' aspect-opinion words

句子级情感分析

句子级（sentence-level）情感分析，主要分析句子作为一个整体的情感倾向，我们分为2分类：正向、负向。训练语料构建：在句子级情感分类上我们利用淘宝评论中好、中、差标签，结合天猫评论用户打分的结果，取出置信度较高的评价作为训练集，并通过回译的方式做了数据增强，通过采用bert+uda的方式在全量评论数据上产出评论级情感分类模型，正向f1值:0.986，负向f1值:0.931。



损失函数：

$$\min_{\theta} \mathcal{J}(\theta) = \mathbb{E}_{x \sim p_L(x)} [-\log p_{\theta}(f^*(x) | x)] + \lambda \mathbb{E}_{x \sim p_U(x)} \mathbb{E}_{\hat{x} \sim q(\hat{x}|x)} [\text{CE}(p_{\theta}(y | x) \| p_{\theta}(y | \hat{x}))]$$

有监督损失项（Supervised Cross-entropy Loss）：计算有标签数据的误差，用交叉熵作为目标。

无监督损失项（Unsupervised Consistency Loss）：用来计算无标签数据的误差。具体的方式是，对于任何一个无标签数据，首先运用回译对于该数据进行一次操作。接下来，利用模型对于增强前后的数据分别进行预测，从而获取到模型对于同一条数据在增强前后的不同结果。最后，利用KL 散度来度量这两个预测结果的距离。整个训练的过程就是要让这个距离尽可能小。

业务场景落地

大家印象

评价标签的pctr从提升了400%，uctr从提升了200%，标签体系覆盖了淘宝所有的商品类别。

首猜智能UI素材

为雁西，岚宝，竹堂老师提供情感筛选后的评价作为素材，在首猜智能UI场景投放，dpv+0.49%，pv+0.16%。

直通车咸鱼场景

投放淘宝商品时加入标签做展示文案，效果：CTR提升了+6%。

手淘miniDetail页输出正向评论

PV点击率 +2.46%。

短视频全屏页飘屏

PV点击率 +2.61%，UV点击率 +0.67%。

PGC标签

针对PGC达人文章场景标签需求，迁移到PGC场景标签抽取模型，覆盖商品百万。

短视频评论氛围优化

占首猜负向视频大池子10.3%。

评论级情感分类

用于评价排序。评价对外输出的原文级情感。商家端评价管理，判断评价内容的情感倾向测算正负向评价对商家转化影响等。

支持智能UI等场景的负向情感过滤

总结与展望

本文主要针对淘系评论场景，对情感计算在ugc的应用进行了一些理解探讨，从而引出了属性级情感分析和评论级情感分析与相关应用，在印象词业务上及其他业务上都取得了不错的效果，但是依然很多的不足。

在后续工作中，比如多实体基于属性的情感分析、对比句、转折句等复杂表达，以及减少pipeline系统的累积误差的Aspect Sentiment Triplet Extraction，属性级情感分类End2End任务，我们将进行更加深入的探索。期待各位有兴趣的同学进行探讨。

• EdgeRec：电商信息流的端上推荐系统

- 作者：绎兮
- 出品：淘系技术

本文将介绍在我们电商首页部署的端上推荐系统EdgeRec，以及在端上推荐系统中的算法实践。

前言

双十一期间，我们业务完成了全面上云，云计算的发展保障了大促期间业务的稳定运行，承接了巨量的流量。在过去的这十年里，依托于大数据，云计算取得了非常耀眼的发展。随着云计算的发展，也面临着一些问题：互联网应用每天面临的大量请求，以及日益普遍的大规模神经网络在线部署，对云端计算产生巨大压力；随着5G的普及和带宽的增加，云端同时面临着巨大的存储压力；对一些实时性要求比较高的应用来说，与云端巨大的通信开销也是交互和体验瓶颈；同时云端“中心化”的计算模式也会带来运维成本和故障风险。

近几年，终端设备的计算与存储能力的快速发展，智能手机的性能成为主要卖点，各种CPU、GPU的快速迭代，内存越来越大，为计算带来了新的空间，端侧的工作不再只是。相对于云计算，端计算的优势在于下面四点：

- 1.数据本地化，解决云端存储及隐私问题；
- 2.计算本地化，解决云端计算过载问题；
- 3.低通信成本，解决交互和体验问题；
- 4.去中心化计算，故障规避与极致个性化。

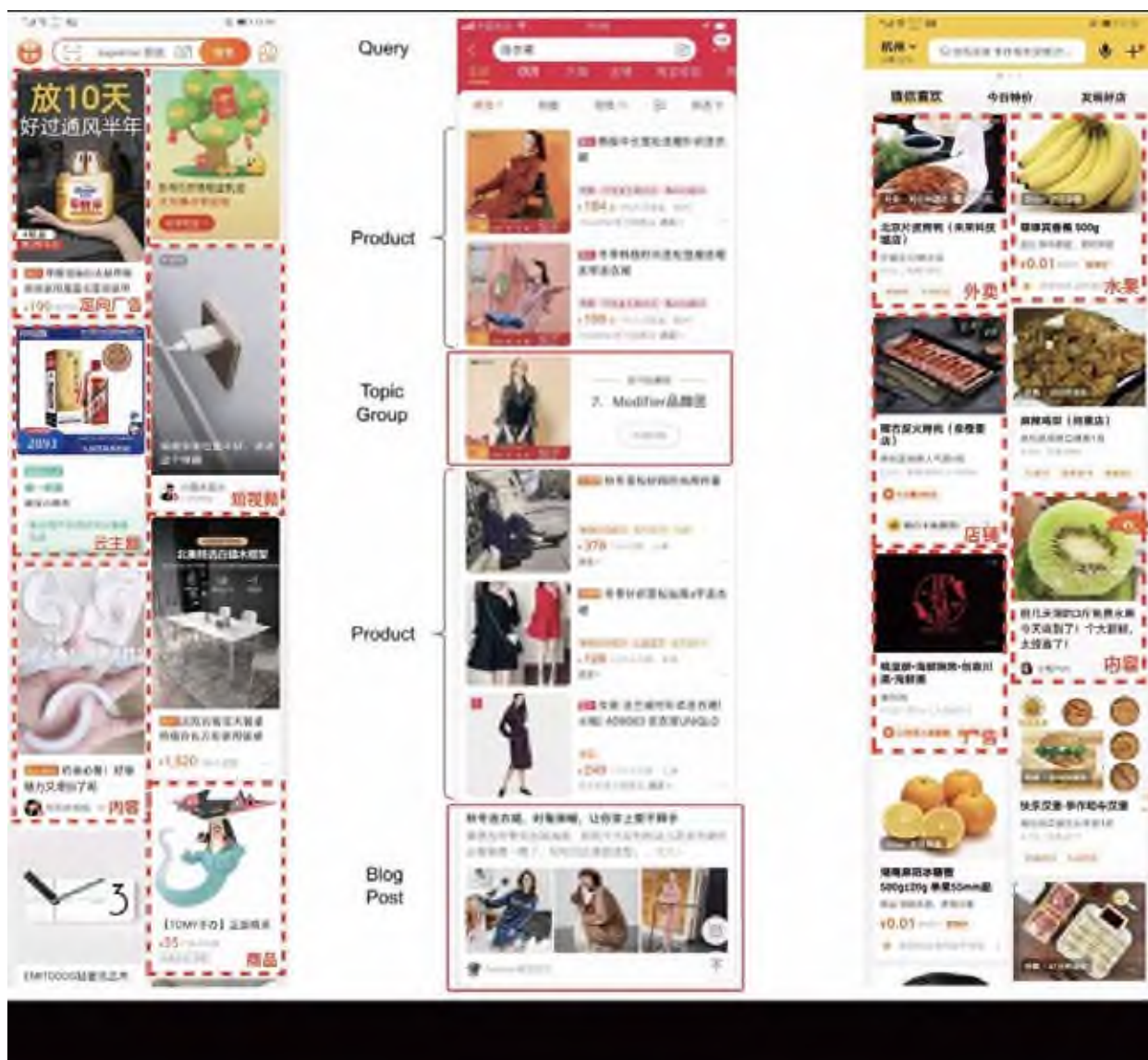
实际上，端计算与云计算不是割裂的。在云计算时代，端侧主要负责信息的收集与展现，云侧负责大规模的计算与存储；在端计算能力逐渐丰富的今天，端侧的计算和存储能力也被利用起来，更多的端云联动能力被挖掘出来。比如，拍立淘将类目检测和物品检测能力放到端上，而将图片-商品召回、排序能力则放到云上；在谷歌输入法Gboard上，智能手机会根据用户的实时反馈首先更新用户个性化模型，并将更新信息回传云侧，云侧则会根据所有用户的反馈，聚合成新的统一输入法排序模型下发给每个用户，完成模型的迭代。结合端侧能力，提升互联网服务的质量与消耗，端云联动的能力越来越多的体现出来。

本文将介绍在我们电商首页部署的端上推荐系统EdgeRec，以及我们在端上推荐系统中的算法实践（部分已发表CIKM2020论文[1]）。第2部分将介绍我们搭建的端上推荐系统EdgeRec整体架构，第3部分将介绍端上重排的问题与端上行为序列建模的算法方案，第4部分将介绍端上智能请求的背景问题与算法实践，第5部分将介绍我们在千人千模算法问题上的探索与算法实践。目前，EdgeRec日常为淘宝首页信息流带来13%的GMV提升。

端上推荐系统EdgeRec

信息流推荐系统的瓶颈

目前，信息流是业界搜索推荐产品的主流形式，其展现信息丰富、可持续交互的特性，使得包括我们电商首页瀑布流、我们电商搜索瀑布流、美团瀑布流、抖音全屏页等，都使用信息流作为主要推荐方式。当然，信息流推荐系统也面临着一些痛点，阻碍信息流的进一步优化。



淘宝首猜

淘宝搜索

美团猜你喜欢

痛点1：用户实时感知弱 / 决策慢

信息流推荐基于“分页请求”机制，仅在分页附近有调整机会。受限于资源消耗和用户体验，一般分页不会特别小。带来问题有：1. 分页过程中无法实时调整策略；2. 实时的意图获取和策略的调整次数太少。

用户的实时行为感知弱：上行数据延迟高：依靠端上实时上报日志->服务端实时解析->推荐系统实时查询的链路，回传延迟一般在10s-1min。

痛点2：云端存储和计算瓶颈

以我们电商平台首页信息流场景为例，日常2w+ QPS，大促峰值10w+ QPS，巨大的在线infer压力导致日常超时的情况时有发生，而大促期间也会面临降级；另外，一般需要存储60天的样本，云端面临巨大存储和训练资源消耗。

痛点3：“千人一模”的局限性

目前的推荐模型使用统一的模型，而“千人千面”源自于特征的个性化；高活用户主导全局推荐结果，20%的用户贡献了80%的训练数据；而每个用户数据分布并非IID，使用同一模型拟合并非最优。（图片引用自[2]）。

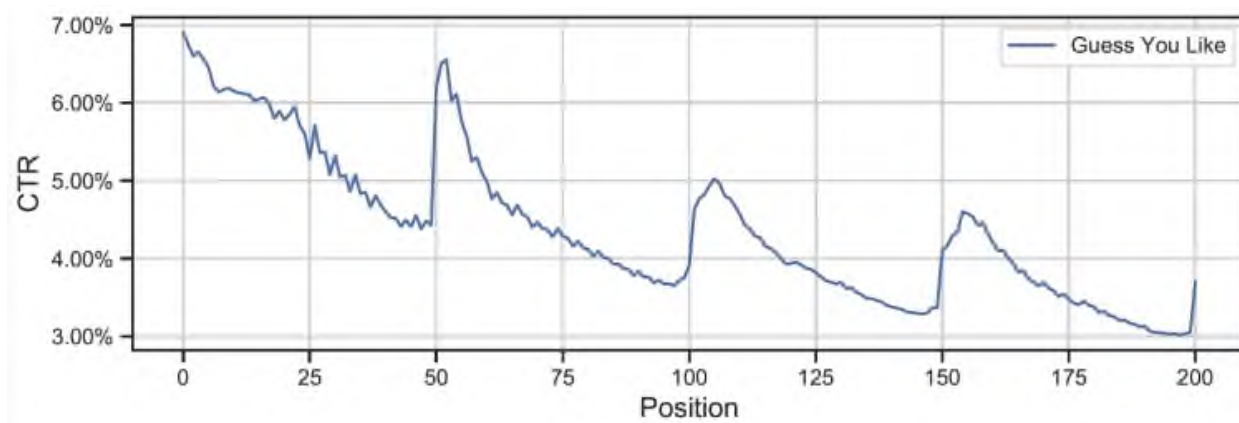


将以上三个问题进行细节分析。导致第1、2个问题出现的原因有两个，一是分页请求固定的分页时机，二是分页后固定的展现顺序。

分页时机在分页推荐系统中是由规则决定的，一般是固定的分页大小。每次分页请求后，服务端可以拿到用户实时的意图，并基于用户意图进行召回排序。如下图，每次分页后用户点击率会增高很多，则是分页时机固定带来

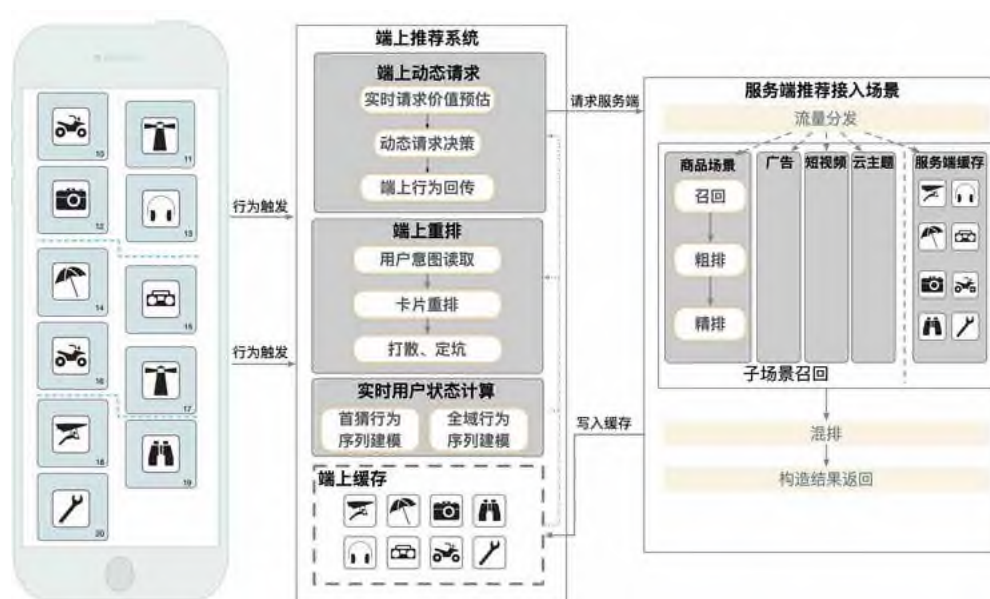
的。因此，分页大小固定带来的弊端是，分页时机与用户意图变更时机不匹配，同时资源消耗在不同用户中消耗是一致的。

展现顺序的固定，则导致了在分页内部卡片的顺序是无法调整的，无论用户进行了何种交互，推荐系统都要等到分页时才能重新调整排序顺序。如下图，每个分页过程中，用户的点击率是持续下跌的。



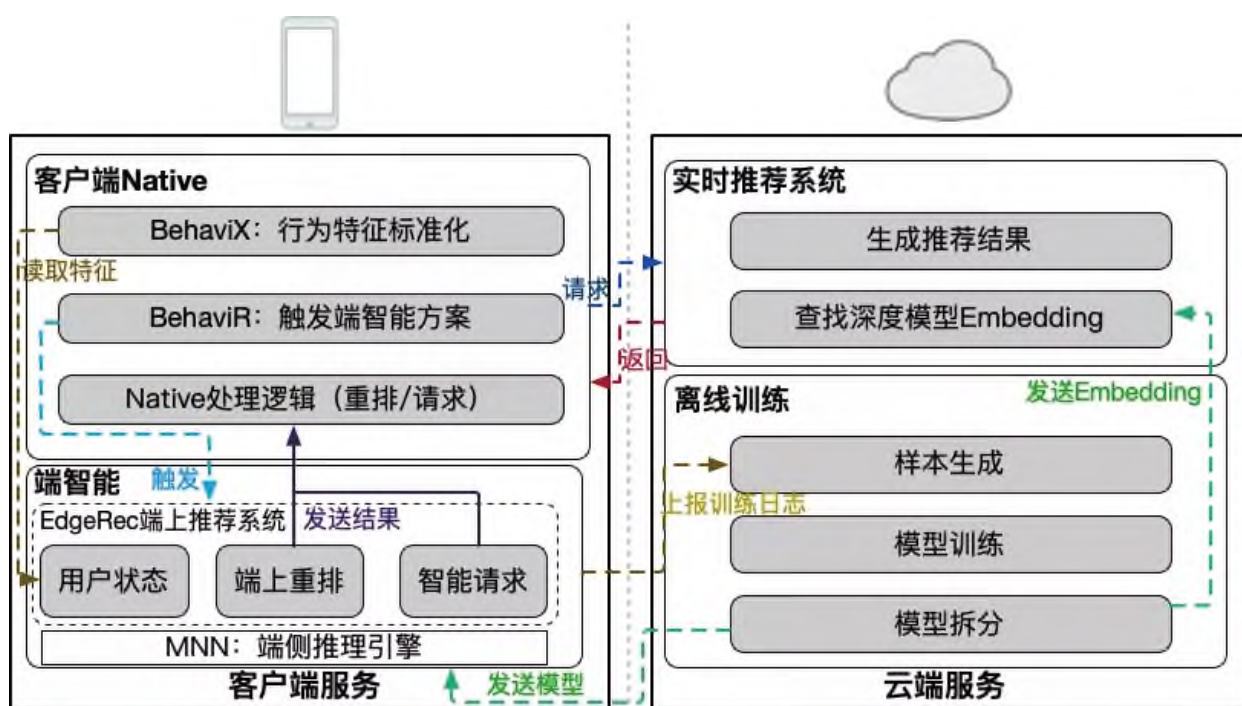
总结下来，分页推荐系统感知/决策慢、资源消耗固定的问题，根本原因是用户意图的变化与推荐系统的请求时机和商品展现时机不匹配。端上是请求的发起者和商品展现的终端，对用户行为有最实时的感知能力，因此，我们希望将请求决策和展现决策这两项最接近用户的任务放到端上，利用端智能的能力，进行进一步的优化。基于这个思想，在我们电商首页信息流推荐里建设了端上推荐系统EdgeRec，通过端上重排能力做展现决策，通过端上动态请求能力做请求决策，整体提升推荐系统的实时感知和实时反馈的能力。

端上推荐系统介绍



在**展现决策**方面，我们研发了**端上重排**能力，能够实时的对信息流中未曝光的卡片进行重新排序，实现用户意图的实时反馈，具体算法方案将在后续部分单独介绍。在**请求决策**方面，我们增加了**端上智能请求**部分，根据用户近期的行为，以及**端上缓存**中的卡片，实时的增加端上智能请求，将用户意图实时带回服务端进行重新召回与排序，具体算法细节也将在后续内容中进行介绍。

端上算法模型上，我们首次在端上进行12G级别的大规模深度神经网络的推理，通过端云联合部署，使得该神经网络可以在端上进行实时预估；同时，我们还基于端智能的分布式部署能力，研发了千人千模的能力，来提升模型的个性化效果，具体算法方案见后续部分。整体的端云联合部署方案、依赖的基础服务等如下图所示。



为了实现EdgeRec端上推荐系统，我们依赖了许多淘系的基础服务，其中包括：

- 1.MNN：淘系自研轻量级端侧推理引擎。为算法的脚本、模型和工具库提供运行环境。
<https://github.com/alibaba/MNN>
- 2.BehaviX：淘系全域端上用户数据中心。端侧用户数据与特征中心，定义一套方便算法使用的数据标准，提供行为本地标准化处理，为算法提供了行为、特征、样本的通用基础数据源。
- 3.BehaviR：淘系端上智能调度框架。串联前台交互、用户感知、用户触达。对用户的实时行为调用端侧算法模型预测意图，进行本地重排、智能刷新等实时决策。

端上重排

为什么要做端上重排

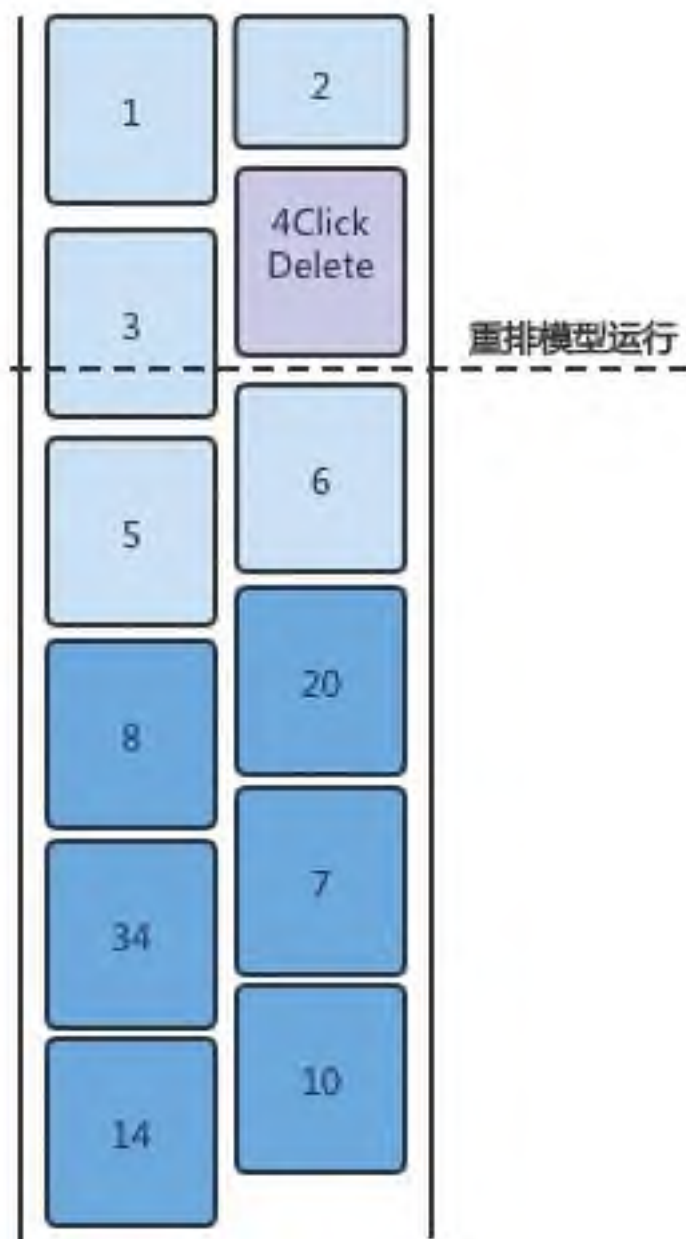
服务端决策延迟

目前，信息流推荐的分页普遍较长。以我们电商首页信息流为例，采用50一分页的分页机制，一屏展示约4个商品，用户需要划十几屏后才能发起新的请求，看到下一页的商品。长分页导致用户在刚分页时点击了某个商品之后，直到下一分页，服务端才能根据该行为下发用户可能感兴趣的相关商品。从上文不同坑位的点击率曲线可以看出，在分页刚开始时点击率最高，然后由于决策滞后，点击率逐渐下降，知道下一次分页请求决策时，点击率才回升。

服务端感知不及时、不完整

服务端对用户行为的感知依赖于行为记录、行为上报和行为解析，整体链路包含了以下问题：

- 1.感知不及时：服务端想要获取用户的行为，需要等待行为从用户手机端上报到服务端，然后服务端对行为加以解析，这些时间延迟导致用户的实时意图无法及时被服务端感知，在大促期间由于流量压力，这种延迟更为明显。
- 2.感知不完整：目前的搜索、推荐、广告算法模型如DIN、BST等，主要特征包括以商品为粒度的曝光序列、点击序列、收藏序列、加购序列、购买序列等序列。而算法模型的进一步升级需要描述某一种行为的“强与弱”。以“商品点击序列”这个行为为例，从细节来看，为了描述一次商品点击的强弱，我们需要详情页的细节行为来描述用户对这次“商品点击”的意图强弱，比如详情页各区块停留时长、各控件点击等；从上下文来看，我们需要这次点击的来源场景信息、APP来访等信息，通过上下文信息来描述一次商品点击的用户意图。而端上是对用户意图进行感知的最实时的位置。



端上行为序列建模

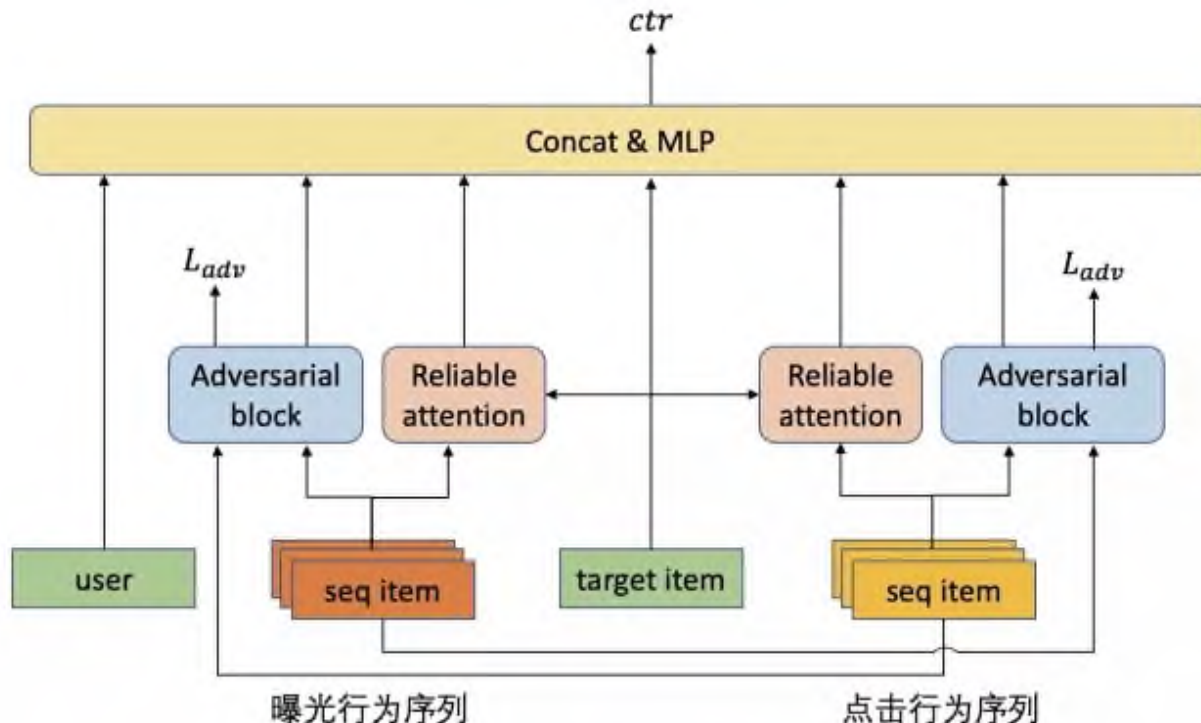
在端上重排模型中，对于用户全域行为的序列建模分为encoder和decoder两个部分，前者整体建模用户状态，可用于重排、混排和智能请求等，后者则基于重排候选商品逐一进行点击率预估，最后将排序结果展示给用户。下面分别介绍这两个模块。详细内容见我们在CIKM2020上发表的论文[1]。

RNN-based encoder。端上重排的曝光序列、点击序列等多个行为序列，首先会分别使用GRU模块进行encode，保留每一个state得到encode之后的各个序列。

Attention-based decoder。排序模型中行为序列decoder常用的target attention模型，会使用目标候选商品与行为序列中的商品计算相关性，然后对序列进行加权。目前attention-based decoder基于对各个序列进行建模时，存在两个问题：

1. 每种序列行为内都存在多种细节行为，如曝光行为序列中的曝光时长、曝光次数、滚动速度和点击行为序列中的点击来源、各区块点击情况、各区块曝光时长等，这些细节信息如果只是简单的concat到行为商品信息中，无法被充分建模
2. 行为序列之间割裂严重，多个序列通常在分别建模之后才在MLP阶段进行隐式交互，没有充分利用到行为与行为之间的关系。

针对以上的两个问题，我们设计了行为内的细节行为Reliable attention建模以及行为之间的序列对抗建模。整体模型架构如下图所示，Reliable attention建模以及行为之间的序列对抗建模，将在后面进行分别介绍。



行为细节建模——Reliable attention

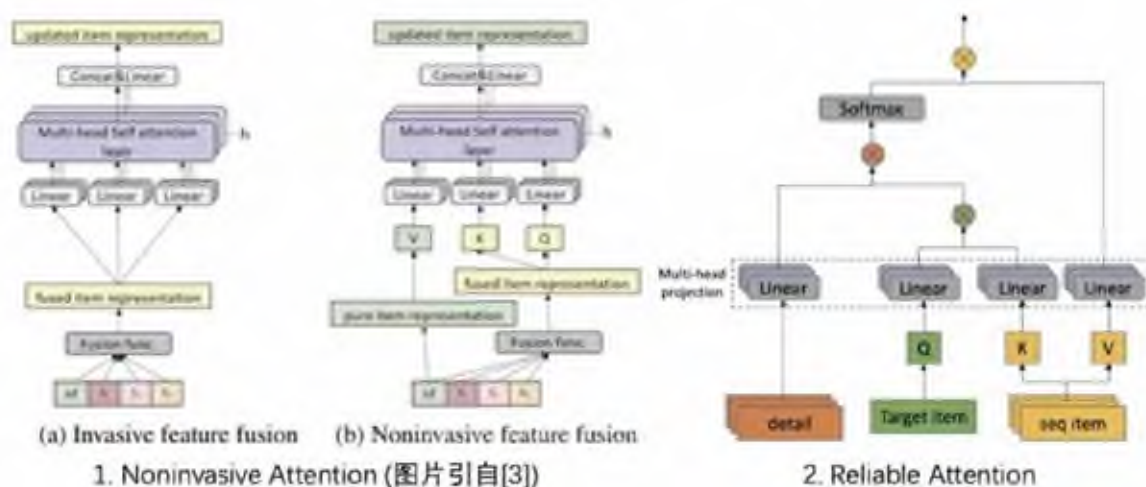
Target attention加权方式在计算权重时只考虑到商品与商品之间的关系，没有考虑到用户在该行为中发生的行为细节。因此，我们首先实验通过noninvasive attention引入细节行为，证明了细节行为的有效性。进一步地，基于行为细节对行为的可信度刻画，我们提出了Reliable attention，对序列进行可信度加权，取得了更优的效果。下面具体介绍这两个模型

Noninvasive attention

noninvasive attention论文[3]中指出，序列建模最终输出的是基于item的兴趣序列的表示，如果将side information混入到item表示中，则会对序列建模带来噪声，使得side information的收益被消减。而noninvasive（非入侵）的方式则是根据混杂了side info的item表示来计算相似性/相关性，再用于对纯净的item representation进行加权处理，从而做到不污染序列item表示又考虑side information带来的收益。下图1表示了invasive attention与noninvasive attention两种方式的对比。

Reliable attention

Noninvasive attention的缺点在于该方法需要在序列decoder前加入一层self attention进行进一步的encode，计算消耗较大，而由于目标候选商品缺乏后验细节行为，所以直接将目标候选商品作为noninvasive attention query用于target attention decode则会出现映射空间不一致的问题。同时，item信息和细节信息fusion之后，由于部分细节信息特征空值率较高，projection输出易受item信息主导。拆分开来计算可以保证不会细节加权和item相似度计算不会互相干扰。因此我们设计了multihead reliable attention的方式，使用行为细节直接在target attention阶段进行序列可信度加权，优化target attention结果。下图2展示了reliable attention模块的计算细节。



多序列建模——序列对抗建模

目前的排序模型中为了全面建模用户的行为信息，常常使用多个行为建模从各方面建模用户，如电商排序模型中的曝光序列、点击序列、加购序列、购买序列等，行为与行为之间显然存在许多关联，因此多个序列之间的交互对于用户的兴趣建模十分重要。目前多序列建模中常见的建模方式为：

1. 隐式交互。各个序列分别独立建模，然后将结果进行拼接，在最后的MLP阶段再进行隐式交互。
2. Cross attention建模。以曝光序列和点击序列为例，在Attention中，曝光行为序列作为query，点击行为序列作为key和value，得到曝光行为序列对点击行为序列的attention结果，再调换一下得到点击行为序列对曝光行为序列的attention结果，最后再将这两个结果进行拼接。

以上第一种隐式交互建模方式对于多序列的建模较弱，而cross attention的方式只是选择当前曝光序列与当前点击序列彼此之间相近的部分，如果当前的曝光序列与当前的点击序列关联性不强，则难以有效挖掘曝光序列中的点击成分、挖掘点击序列中的负反馈成分。

因此，我们设计了一种序列对抗的方式，来挖掘出行为之间内在的联系，从而寻找当前曝光序列当中有哪些行为是真正的负反馈，而哪些行为与点击有更相近的关系，相应地，在点击序列中寻找哪些行为是有效点击，哪些行为是无效点击。

具体地，以点击行为序列对曝光行为序列进行对抗建模为例，对抗模块（Adversarial block）网络结构如下图所示：



其中，Pool操作表示使用mean pooling将序列信息进行聚合；Split操作表示把点击行为序列中分割为不重叠的部分。Discriminator使用的是MLP，Discriminator将输入的两个向量进行concat，并输出两个向量来自同种行为序列的概率。

在序列对抗中，输入Discriminator的两个向量只能包含行为的item信息，不能包含细节行为的具体embedding信息，因此Selector使用的是Reliable attention，其中Query来自点击行为序列聚合后的结果，而Key和Value都来自曝光行为序列。该模块期望通过Selector选出曝光行为序列当中接近点击成分的信息，而Discriminator负责鉴别两个不同的序列是否属于同一种序列。对抗模块计算出 p_{true} 和 p_{fake} 之后，对抗loss为：

$$\begin{aligned} L_{adv} &= cross\_entropy(p_{true}, 1) + cross\_entropy(p_{fake}, 0) \\ &= -\log(p_{true}) - \log(1 - p_{fake}) \end{aligned}$$

对于Selectot模块，需要最大化该对抗损失 L_{adv} ，Discriminator模块则需要最小化该对抗损失 L_{adv} 。

离线效果与线上效果

双十一期间，端上重排给我们电商首页信息流带来11%的GMV提升、12%的成交笔数提升。在端上重排模型进行端上行为序列建模优化的实验对比为

	AUC
baseline	0.7497
with noninvasive attention	0.7501
with reliable attention	0.7508
with reliable attention & seq adversarial block	0.7518

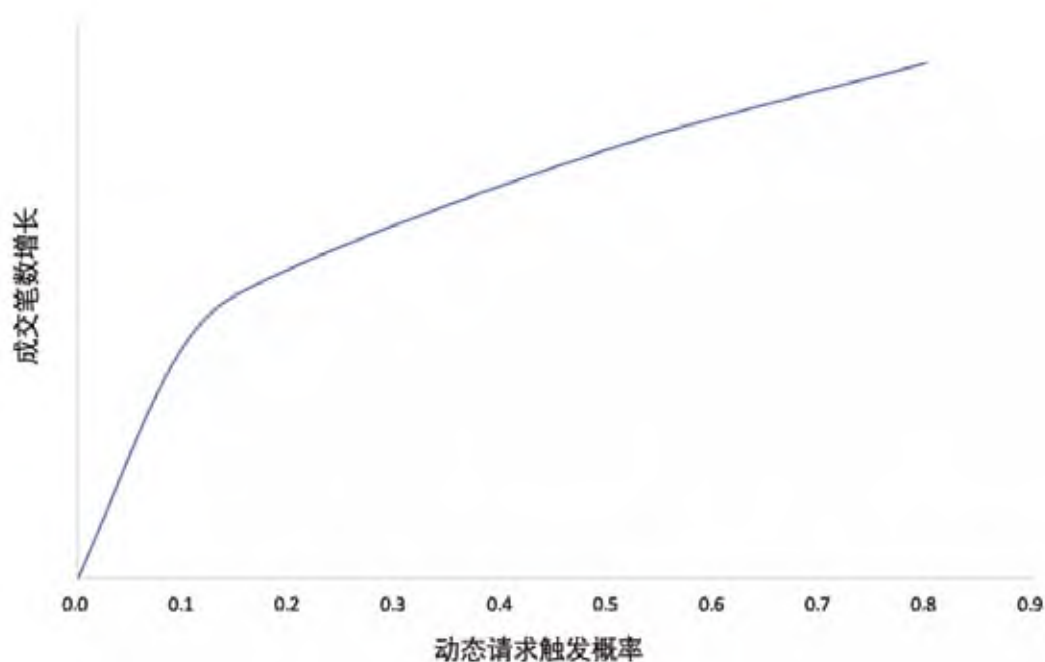
从上表对比可以看出，在端上重排模型中加入Noninvasive attention建模细节行为，离线AUC提升较为微弱。使用Reliable attention建模细节行为，离线AUC提升为0.0011。在Reliable attention的基础上，加入序列对抗建模，离线AUC提升从0.0011变成0.0021。上线后成交笔数提升2%，GMV提升2%。

端上智能请求

背景

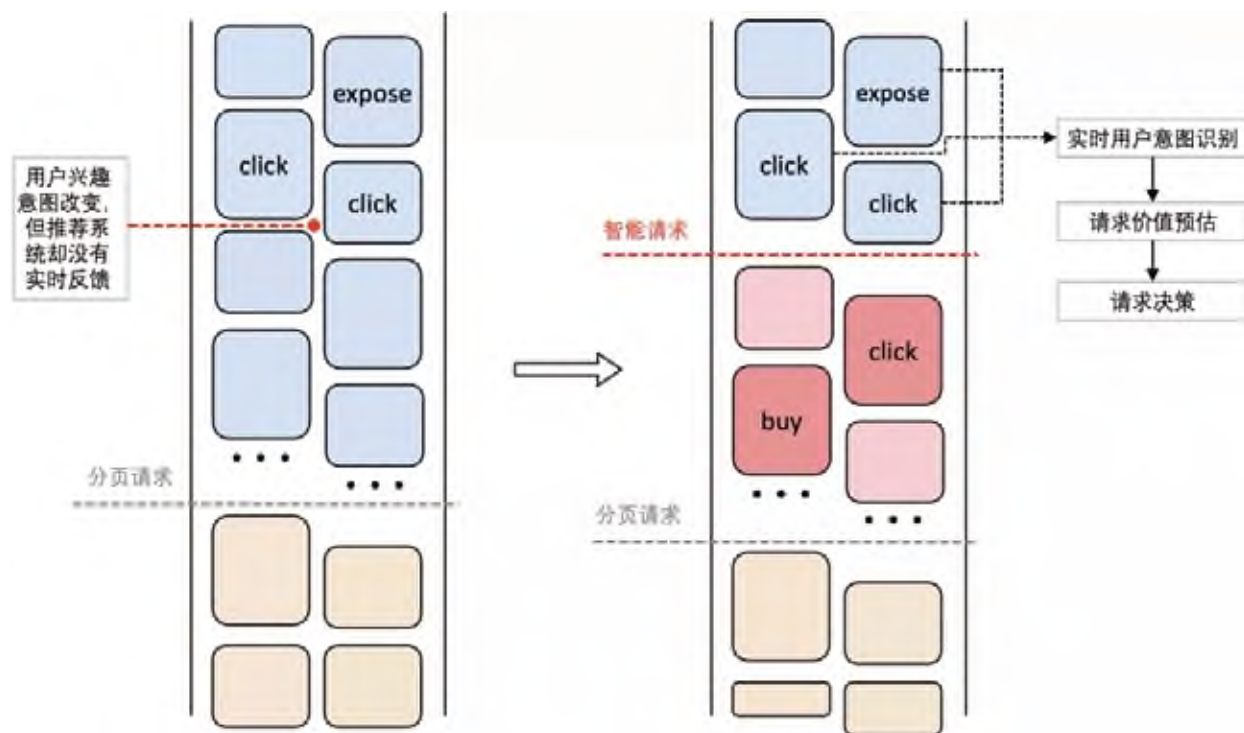
在信息流推荐中，分页请求机制极为常见，一方面受限于网络带宽，为了减小延迟优化用户体验，另一方面为了能够留下调整推荐内容的空间，针对用户的推荐内容会被分批推送到端上。但考虑到QPS压力，一般分页的长度固定，而且不会特别小，这就带来一些问题：

1. 推荐内容更新的时机与用户意图变更的时机无法匹配。如图1所示，用户的兴趣意图和目标商品随用户浏览而变化，但分页请求导致端上商品池更新时机固定，云端的推荐内容排序无法实时的对用户意图改变做出反馈。
2. 无法实时的对用户意图进行检测。只有实时的用户行为特征才能反映用户当下的需求。每次分页请求后，服务端可以拿到用户的行为意图特征，并基于此进行召回排序。但由于分页请求时机固定，加上网络的带宽和延迟，用户的实时行为中大量全域细节行为无法上传，导致云端推荐模型对用户意图的推测出现偏差。
3. 无法动态分配请求资源，导致大量无效请求，而且无法实时调节QPS压力。如下图，在分页请求的基础上，在每一个分页中以不同概率随机插入动态请求，随着概率的上升，用户的购买率也随之上升。由此可见更多的请求能够带来更好的推荐，导致更多的购买。由于不同用户兴趣意图变更时机不同，不同时刻对不同用户施加一次请求的产生的价值也不同，非个性化的统一分页请求会导致大量请求资源的浪费，线上效果提升与资源分配不匹配。



端上是请求的发起者和推荐内容展示终端，对用户行为有最实时的感知处理能力，因此我们希望通过端上智能请求来做个性化请求决策，提升推荐系统的实时感知和实时反馈的能力，同时预估请求价值，合理分配请求资源。

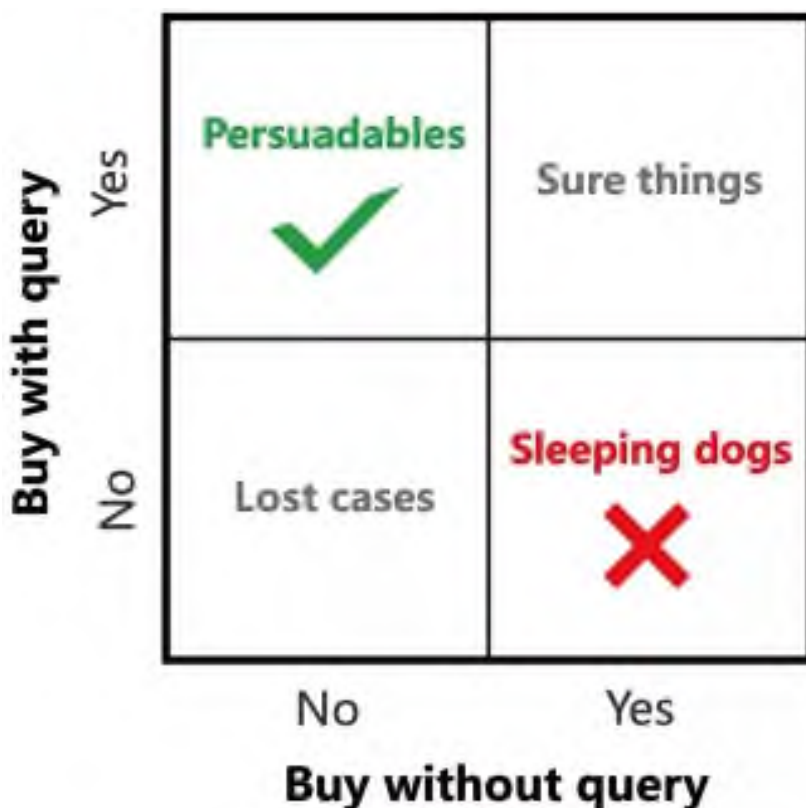
端上智能请求会检测用户在端上检测用户兴趣意图的变化，通过请求价值预估得到在当下给该用户施加请求产生价值大小，再通过请求动态规划算法来做出是否请求的决策，让请求产生真正的线上收益。如下图所示。



双11期间，在我们电商首页信息流推荐场景中，端上重排结合智能请求算法和请求价值预估模型，相比只结合动态请求规则(每页固定增加一次请求)，线上成交金额提升超过3%，说明智能请求能够实时感知用户兴趣变化，在更好的时机更新端上商品池，为用户带来更好的推荐，促使购买行为的发生。

请求价值预估Request Uplift

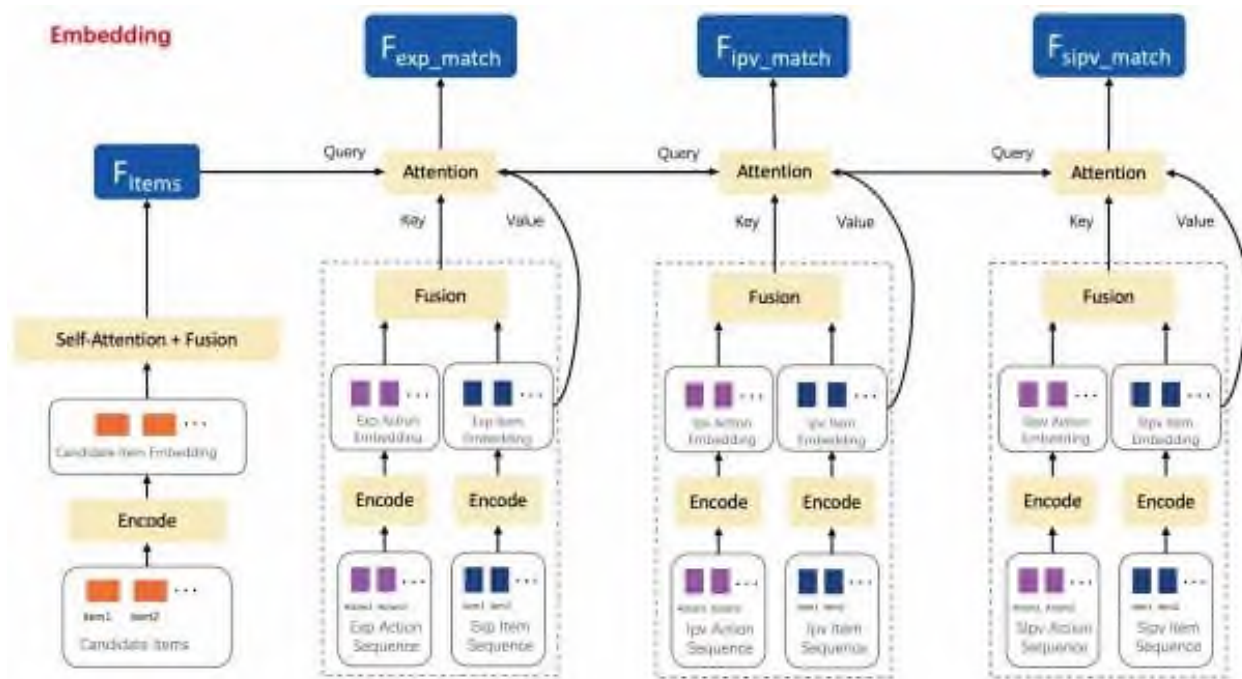
在请求价值预估中，我们使用因果推断-请求价值增益模型，来预测请求导致的购买率增益，而不是购买率本身。如下图所示，由于不同用户对请求的敏感程度有差异，只有部分用户或者在部分情况下，给与更多的请求能够使线上产生真正的收益。部分情况下用户的行为不会因请求而改变，而有一小部分情况下请求甚至会有负面影响[4][5]。所以我们用因果推断-请求价值增益模型分别建模施加请求和不施加请求下的购买率和对应的购买增益，推测出在特定情况下，由于增加请求导致的真实购买增益，来作为当前请求的价值。



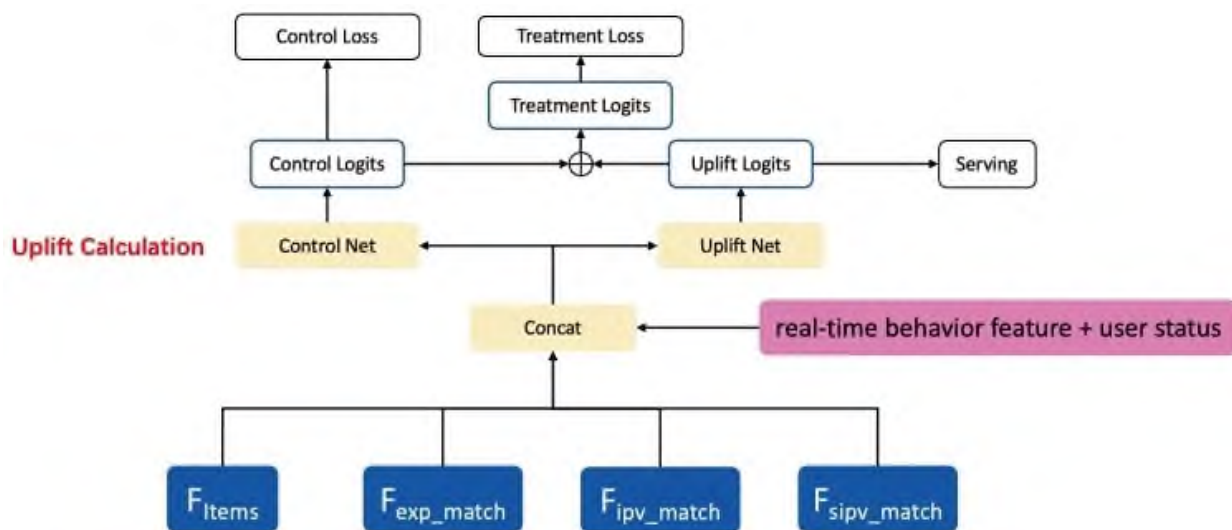
请求价值预估模型 M 会分别预测不施加智能请求下用户的购买预估 $Logits_{control}$ 和对应的购买率 $P_{control}$ ，以及购买增益，也就是请求价值预估 V_{uplift} 并同时推断出施加智能请求下用户的购买预估 $Logits_{treatment}$ 和对应的购买率 $P_{treatment}$ 。 $\widehat{CATE} = P_{treatment} - P_{control}$ 就是对 $CATE$ 的预估，而 V_{uplift} 与 $\widehat{CATE}$ 成正相关，所以我们会直接用 V_{uplift} 来作为请求决策的依据。

模型上分为两部分，第一部分中与端上重排模型一样需要进行行为序列建模，一方面，使用 embedding table 和 user state 模型预先处理用户在端上的行为特征 F_{beh} ，得到包括用户曝光商品特征序列 F_{exp_seq} ，用户历史点击商品特征序列 F_{ipv_seq} 和用户实时点击商品特征序列 F_{sipv_seq} (指本次 session 内的点击商品特征)，这三种特征。这三种特征代表了用户潜在的兴趣意图和购买倾向。另一方面，通过 GRU 将待重排商品池中的商品序列 $Item_{seq}$ 编码成特征序列 F_{item_seq} 。

在后续的模型中，我们通过 attention 等操作来挖掘出重排商品池中的商品序列和用户潜在兴趣的匹配程度，当匹配程度较小，说明若此时增加一次智能请求，能带来的增益将较大。首先，我们通过 self-attention 和均值操作将 F_{item_seq} 融合编码成 F_{items} 。之后把 F_{items} 作为 query，把 F_{exp_seq} 、 F_{ipv_seq} 、 F_{sipv_seq} 同时作为 key 和 value，通过 multi-head attention，并用求和操作将其融合编码成 F_{exp_match} 、 F_{ipv_match} 、 F_{sipv_match} 以此来代表商品池中的商品和用户兴趣的匹配程度。之后， F_{items} 、 F_{exp_match} 、 F_{ipv_match} 、 F_{sipv_match} 将会和上端上场景下的实时上下文 F_{cont} 并在一起，来作为模型第二部分的输入。



在模型第二部分中，模型第一部分的输出会被输出到两个分支control net和uplift net中，分别用来预测不施加智能请求下用户的购买预估 $Logits_{control}$ 和智能请求价值预估 V_{uplift} ，并以此来计算出对应的 $P_{control}$ ， $Logits_{treatment}$ 和 $P_{treatment}$ 。这里，我们简单的用2个MLP来作为control net和uplift net。



模型第二部分输出的 $P_{control}$ 和 $P_{treatment}$ 会和对应的control组和treatment组的数据中的label来计算 control loss和treatment loss，最终的loss就是该两个loss的加权和。这里我们用cross entropy来作为loss函数。最终线上部署时，我们仅部署Uplift Logits的通路部分模型，通过Uplift值来决定请求。

离线指标与线上效果

我们把模型做出是否请求的决策之后 n 坑内是否发生购买行为作为Label。control组，treatment组和总体的AUC作为离线指标。

all	control	treatment
0.79564	0.81003	0.78040

双11期间，端上重排[1]结合智能请求模型相比只结合智能请求规则(每页固定增加一次请求)，线上**成交笔数提升2%**，**成交金额提升3%**，说明智能请求能够实时的检测用户兴趣意图变化，在更加合适的时机发起请求来更新端上商品池，为用户带来更好的推荐，刺激购买行为的发生。同时，我们将智能请求的业务效果与规则请求持平后，发现**请求QPS能够降低32%**，对云端计算资源缩减也有很高的价值。

智能请求模型会在更合适的时机做出施加请求的决策，促使请求之后的购买率远大于不请求。如表中所示，模型决定请求之后10坑内的购买率(%)是不请求的360%，是随机请求的259%。数据表明智能请求模型大大提高了请求之后的购买率。

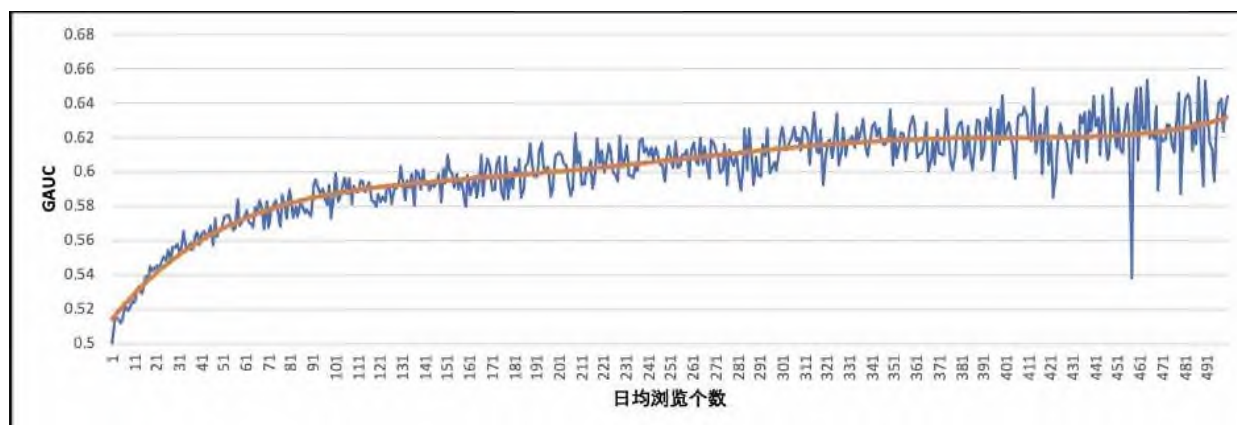
	无请求	请求(随机)	智能请求(模型)
5	0.04816	0.06685	0.17521
10	0.09295	0.12935	0.33508
15	0.13439	0.18854	0.47695
20	0.17371	0.24266	0.83394

MetaFusion个性化模型融合算法

背景

“千人千面”是过去几年搜推广业务增长的关键点，如何个性化信息检索结果成为了算法的工作重点，给搜推广业务带来了巨大的业务增长。目前，“千人千面”是源自特征层面的个性化，比如普遍使用的用户历史行为特征、用户长短期统计等，而模型都是使用统一的大模型。同时，用户的行为数据会从客户端传回服务端，然后在云端训练统一的推荐模型。大模型的方式，能够很好的表达Item-to-Item、User-to-User的相关关系，在保持个性化的能力下提升模型的泛化性。

然而，这种千人一模的方式，也带来了推荐公平性的问题。目前大部分场景里，20%的高活用户贡献了80%的训练样本，这也导致了模型会更加偏向于高活用户的习惯进行推荐，从下面图中曲线可以看出，日均PV较低的用户其模型打分准度（GAUC）也较低。另外，千人一模的方式，也会有数据隐私保护的问题。因此，我们希望能每个用户提供其个性化模型，以达到极致的个性化。



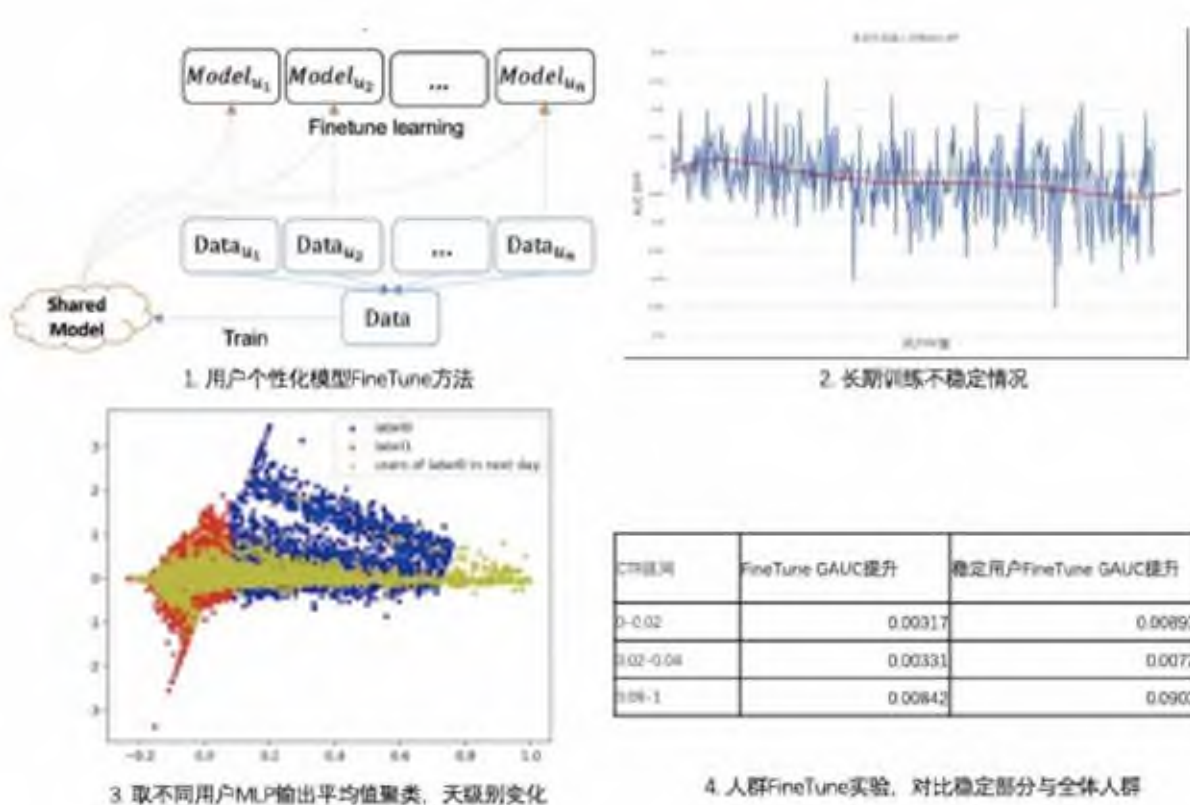
训练个人模型的难点

虽然个人模型有很高的研究价值，但是该工作的难点也很多。首先我们使用了最常用的方法，基于所有用户日志训练出元模型，并在单个用户日志上Finetune的方式，产出用户个性化模型，整体流程如下图1。通过该方式发现的问题有：通过该方式发现的问题有：

- 1.长期训练不稳定：如下图2，随着个人数据的多次训练，逐渐过拟合于用户本人局部数据，导致用户-用户的协同过滤信息缺失，最终auc反而会变差。

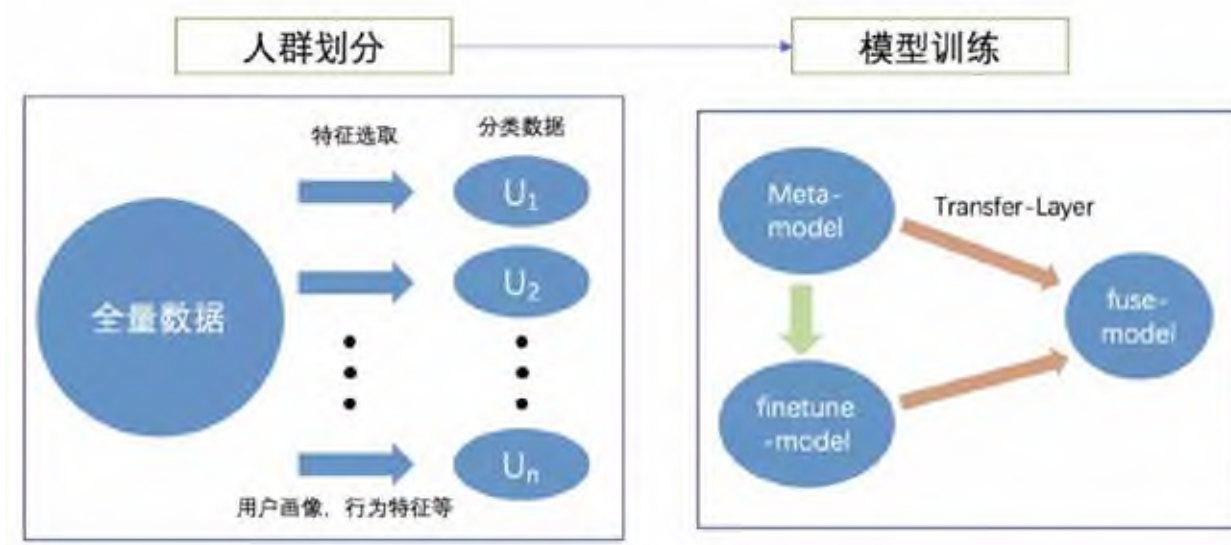
2. 用户行为不稳定：如下图3，取第N天大模型每个人用户表征层的输出值平均值，拼成一个向量做聚类，通过Kmeans算法聚成两个类别，并通过PCA降维到两维，可视化成红色和蓝色两类。而当使用第2天蓝色部分用户的数据，继续通过该模型输出用户表征层输出值时，却发现表现为黄色部分。即不同天之间用户的行为有很大的变化。

为了验证该问题，我们取了不同Ctr区间的人群，使用N-1天全人群大模型做Meta Model，并使用单独人群第N天数据做FineTune得到FineTune Model，验证FineTune模型对比MetaModel的在N+2天数据上评估的提升。并分别对比该人群全部用户和稳定用户（第N+2天用户Ctr仍在该区间内）的效果，整体。结果如下图4，发现稳定用户部分的提升，相比全部用户部分，GAUC提升会高出不少。而这块也是用户个性化模型的增长空间。



MetaFusion个性化模型融合算法

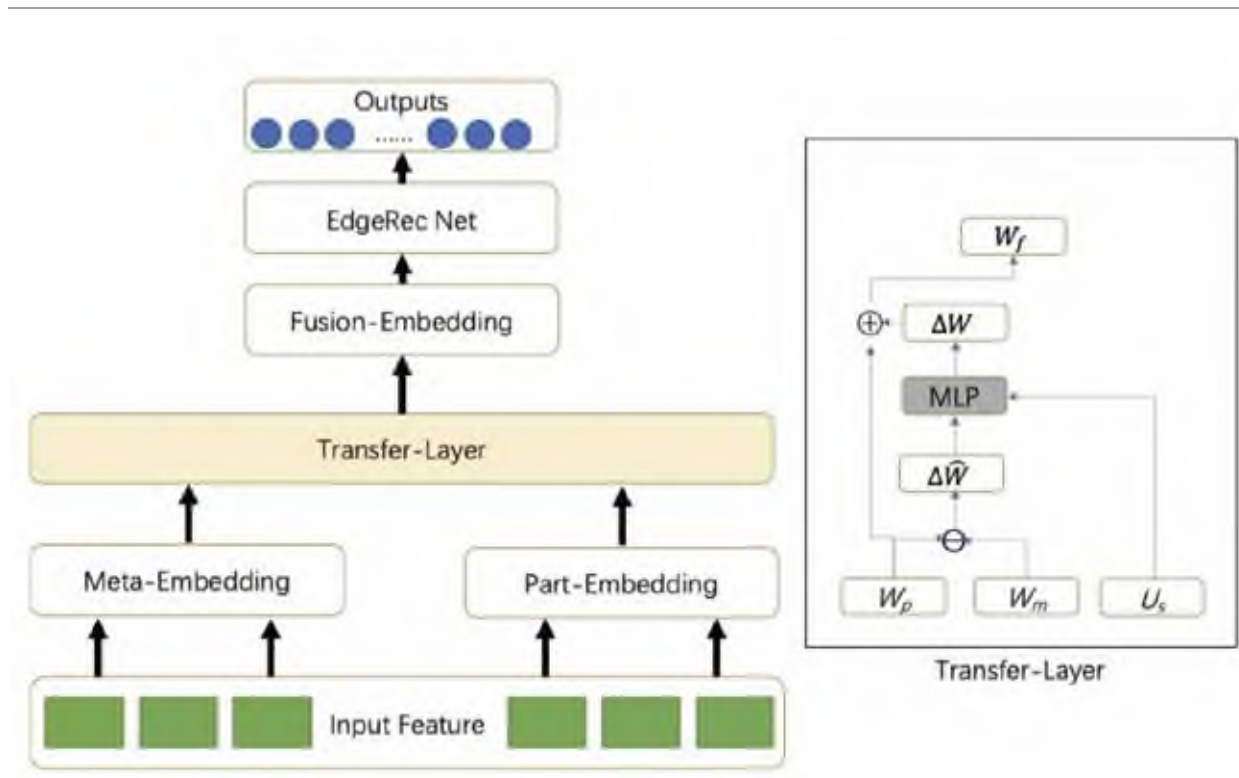
为了解决以上提出的两个问题，我们提出了MetaFusion框架和相应的算法，其中框架如下图所示：



为了解决单人数据训练过拟合的问题，我们决定使用人群来训练人群的FineTune模型，通过将单个用户扩大到用户群体，可以一定程度解决训练样本稀少和数据不均衡问题，对于人群的划分，其目标同个性化模型Fine-Tune类似，是为了缓解在推荐中高活用户的主导现象，同时也旨在为存在冲突性行为的用户建立更加个性化的推荐模型。

为了解决用户行为不稳定的问题，我们根据用户自身信息来做Meta模型和FineTune模型的个性化融合，识别用户的不稳定程度，并对两个模型进行融合。其原因是，即使在分类人群的数据样本 U_i 上训练拟合了，但 U_i 在后一天的数据上的特征可能已经变化了（即有一部分用户已经不在之前划分的区间标准内了），如5.2部分的图3所示，因此依旧用拟合训练集的模型不一定有效，需要学习用户特征的变化情况来提高分类模型的稳定性。

在原有全量数据的基础上，我们可以得到集中元模型，其模型参数用 W_m 表示，对于每一个人群而言，我们利用其对应部分的第 T 天的数据作为训练集样本， $T+1$ 天的数据作为测试集。在训练集上得到模型参数 W_p ，该模型我们认为其拟合第 T 天的用户行为，为了使模型在 $T+1$ 天保持有一定的稳定性，引入了Transfer-Layer，并结合用户稳定性统计特征 U_s ，融合 W_p 和 W_m 的模型参数，学习 W_p 和 W_m 的模型变化和用户稳定性统计特征之间的关系，最终融合出 W_f 。我们认为Transfer Layer是有更高的泛化性的，能支持多天之间的用户状态变化。模型结构如下图所示。



离线效果上如下表，可以看到MetaFusion后，GAUC提升比单独Finetune后都有相应的提升。可以认为“**稳定用户FineTune GAUC提升**”是用户个性化模型的能力上限，Meta-Fusion后的确向该值逼近了一些。我们将通过Meta-Fusion产生的用户个性化模型利用端智能的分离部署能力，部署在了用户个人手机上。在线效果上，在不同人群上普遍有2%的Ctr和点击量的提升。

CTR区间	FineTune GAUC提升	稳定用户FineTuneGAUC提升	Meta-Fusion Model GAUC提升
0-0.02	0.00317	0.00893	0.00567
0.02-0.04	0.00331	0.0072	0.00423
0.09-1	0.00842	0.00903	0.00875

总结

EdgeRec建设的三年内，每一年都有新进展，双十一GMV提升从19年的+5%，到20年的+8%，到今年的+13%，每年都展现了其巨大的增长空间，成为推荐系统中不可或缺的系统。截至目前，端上推荐系统体系的潜力做了一定兑现。接下来，端智能需要与云上算法做端云算法间的协同，端上发挥其信息量大、信息实时的优势，云端发挥其候选存储空间更大的优点，通过端云协同提升业务效果。

● 端智能研发核心套件，MNN 工作台深度解析

- 作者：鹿尤
- 出品：淘系技术

MNN 工作台开源地址：

开源地址：<https://github.com/alibaba/MNN>

背景

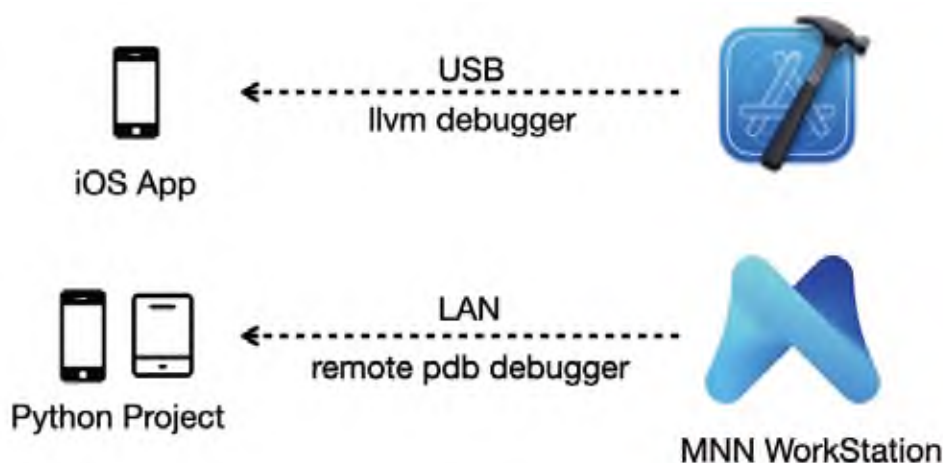
随着移动互联网的快速发展，人工智能在移动端上的应用越来越广泛，端智能在图像识别、视频检测、数据计算等核心场景发挥着重要作用。众所周知，Python 是算法进行端智能研发的首选语言，目前阿里巴巴内已经建立了端智能的研发生态，包含 Python 虚拟机，系列的数据/视觉等运行时 Python 扩展库、Python 任务调度库，以及配套的任务发布系统等等。

端智能的主要场景在数据计算、视觉内容理解这两个领域，如果算法研发完全在 PC 端的会简单的多，因为 PC 环境对于 Python 的研发有官方天然的支持，但在移动设备上，进行算法的部署、调试、验证，仍处在“刀耕火种”的时代，目前算法主要通过两种途径：

- 1.日志调试：完全脱离运行时代码，通过在代码中插入日志，验证程序的运行逻辑和结果。
- 2.PC 模型端侧环境：为了能够进行代码的断点调试，在 PC 侧搭建和安装端侧 Python 代码依赖的所有库，让 Python 程序能独立地运行和在和调试在 PC 端通用 IDE（如 PyCharm）上，数据输入则使用 Mock 的方式。

通过打日志当然也能验证结果和定位问题，但一旦工程稍微复杂点，生产效率会非常低；而脱离移动端环境，在 PC 侧独立运行，并不能反应真实的运行环境（实时数据、运行性能等），无法保证 PC 和端侧数据的一致性。

上面两种研发方式的主要问题是脱离了代码当前的运行环境，因而会产生调试信息的缺失。我们知道，每一种语言在开发者之间的流行都离不开配套的 IDE 调试工具，就像可以使用 Xcode 调试手机应用的 OC 代码一样，我们需要一种能真正进行端智能算法在端侧部署和调试的工具和方案，来提高 AI 研发生产效率。



MNN 工作台

MNN 是阿里巴巴开源的一个轻量级深度学习端侧推理引擎，核心解决深度神经网络模型在端侧推理运行问题，涵盖深度神经网络模型的优化、转换和推理。

开源地址：<https://github.com/alibaba/MNN>

除 MNN 引擎建设外，MNN 团队还深度参与了阿里内部的端AI应用实践。AI 应用门槛高、算法模型部署链路长是长期困扰我们的问题。为了解决这两个问题，我们将上百次实践中长期积累的方案沉淀为一站式 AI 研发范式——MNN 工作台。（www.mnn.zone 或点击文末阅读原文，下载 MNN 工作台）

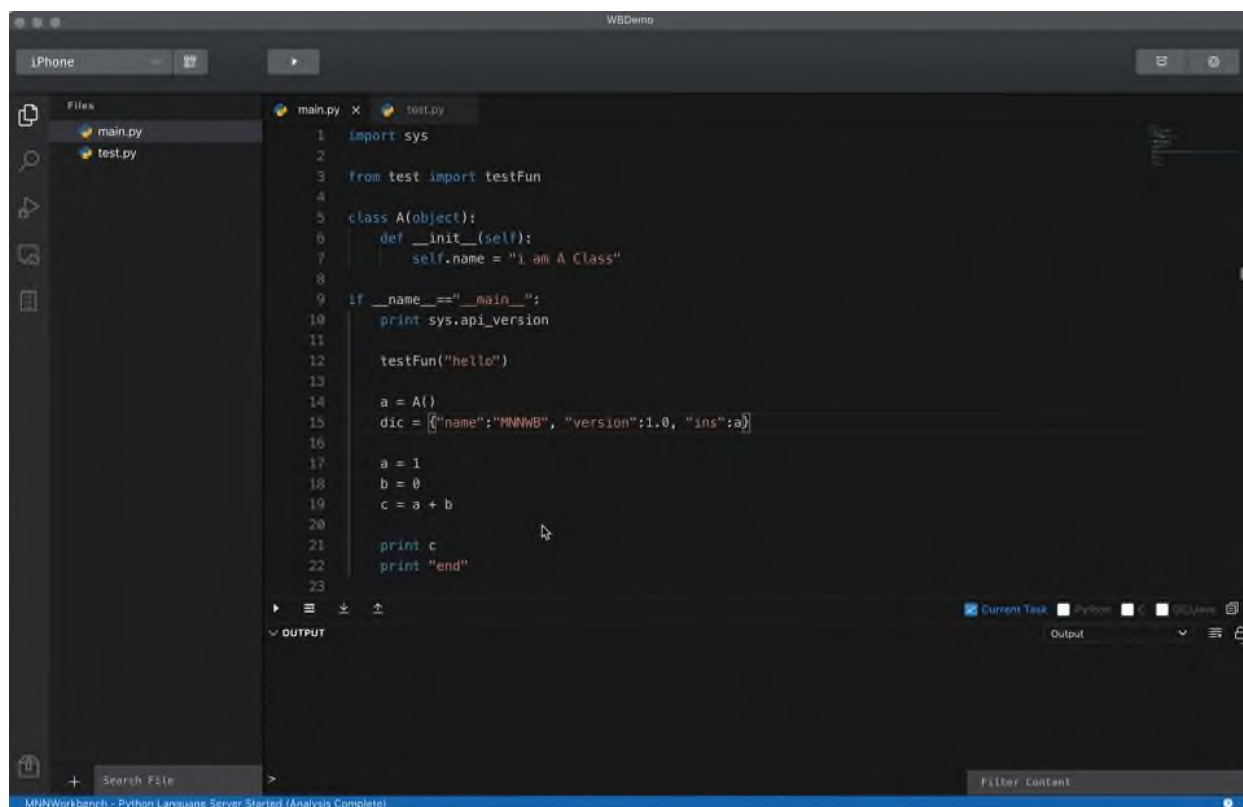
MNN 工作台提供了端 AI 的研发范式，同时它也是一个对移动端 Python 的通用调试编辑器，工作台不仅包含标准的端侧 Python 调试能力，还覆盖了阿里集团内端智能算法常用的研发部署场景，是算法研发的效率工具。



工作台提供了 VSCode 风格的 Python 集成开发环境，它的使用过程很简单，一般分为三步：



- 1.建立连接：应用通过扫描工作台上的二维码，建立局域网内手机和PC的连接
- 2.代码推送：工作台上的工程通过局域网推送到端侧应用中
- 3.运行：端侧应用触发执行端侧 Python 代码，如果设置了断点，就可以进行端侧代码的调试了



端计算研发

MNN 工作台同样为阿里内重要的端智能场景 - 端计算的研发提供了高效的本地调试部署方案，覆盖了 Walle/Jarvis、CV 这几个主要的端计算场景。

研发流程



在此之前，端计算的研发是依赖于平台发布的，在开发阶段将调试日志加到代码中，通过预发/beta发布到平台，然后再扫码连接到平台将日志输出，通过日志来定位和分析问题，以此往复。这种方式相对比较原始，开发调试周期也比较长，一次修改或验证就是一轮上传发布的操作，而且调试信息只有日志，整体开发效率比较低。



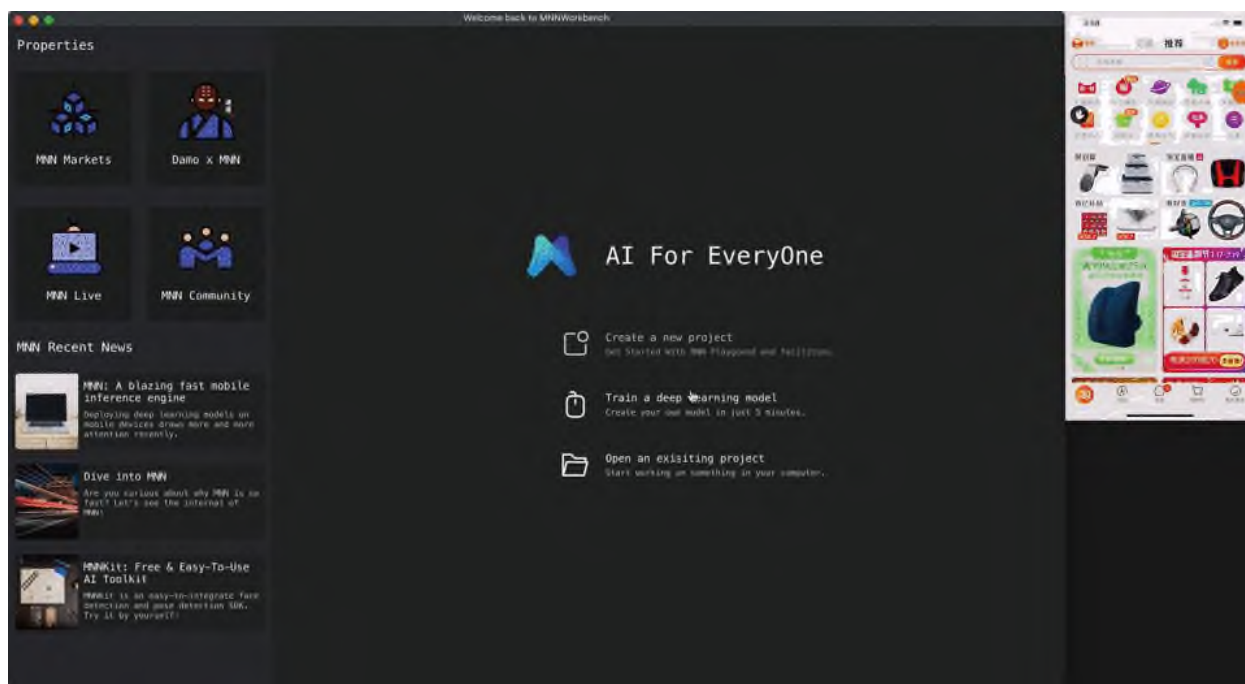
上图为算法使用工作台进行端计算研发的一般过程：

- 1.新建端计算工程，或打开从平台 clone 的端计算工程
- 2.应用扫码连接工作台，此时工作台会实时输出算法的日志
- 3.在工作台 IDE 环境中开发代码，修改配置，添加资源
- 4.运行任务，此时任务会通过本地推送部署到端上。线上/预发环境都可以，任何代码都不会影响到线上，无需担心稳定性问题
- 5.代码断点调试、查看堆栈、变量等
- 6.工作台本地测试通过后，提交代码，再进行后续的平台发布操作

工作台有效解决了算法在开发阶段的调试部署问题，它的主要优势在于：

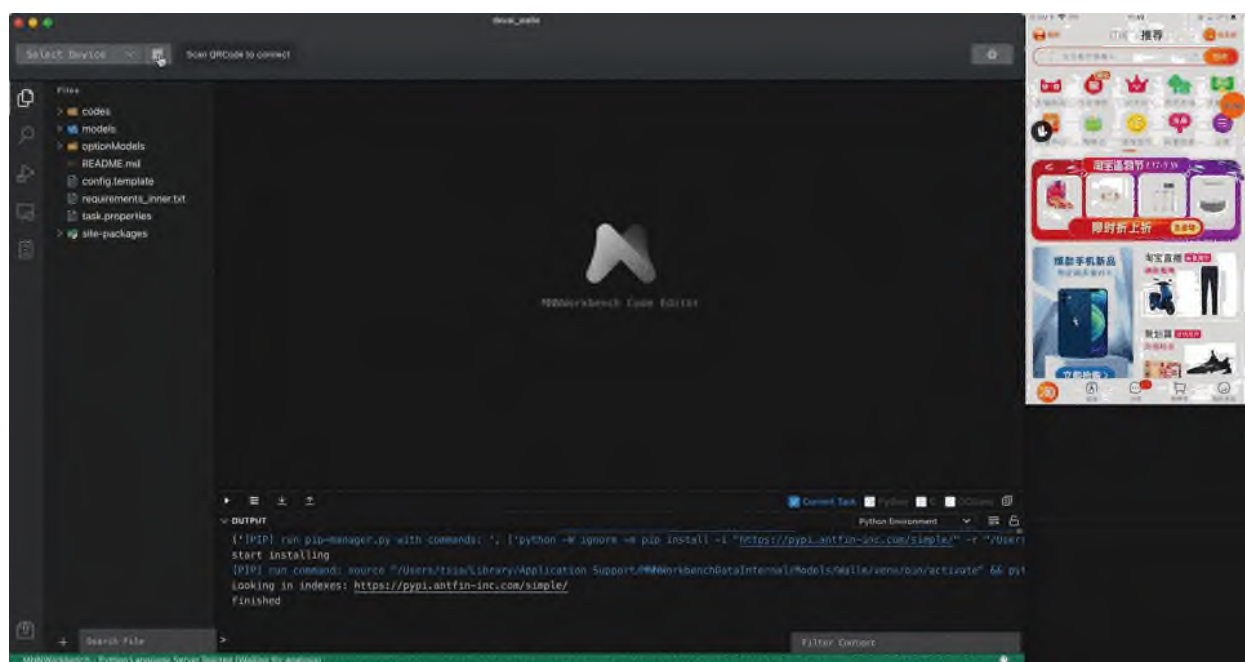
- 1.本地部署，开发阶段可以完全脱离平台
- 2.支持端计算工程的实时 Python 调试
- 3.端上部署不受预发/线上环境，或窗口期的限制
- 4.实时查看端侧算法的运行性能

持久化调试



端计算任务的触发一般都是 UT，也有工程主动调用的，不管触发方式如何，代码只要推送一次，后面每次触发都会运行调试，除非代码有修改才需要重新推送，这就是工作台的持久化调试能力。

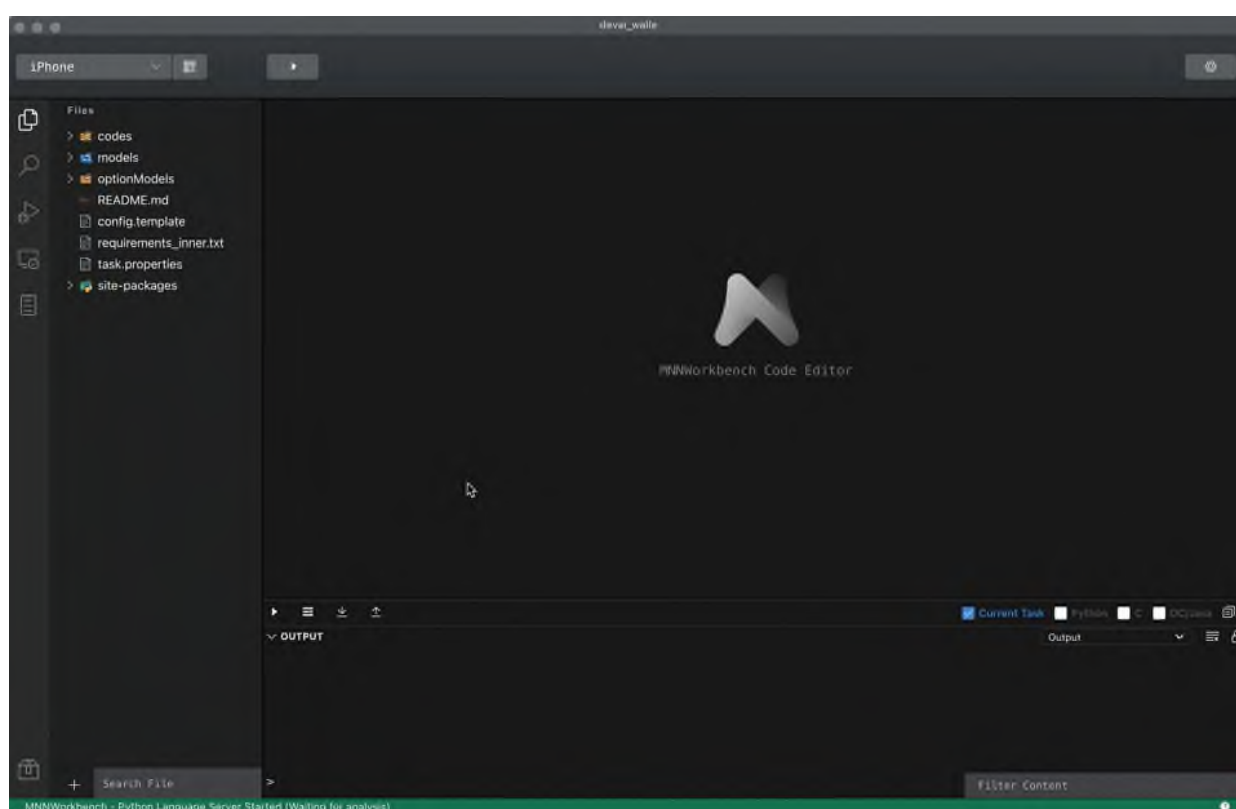
实时日志



工作台支持各种类型日志的实时打印，在设备与工作台连接成功后就可以开始使用，支持的日志类型包含：

- 1.Current Task：本工程日志，包含本工程普通 print 日志和工程的运行日志
- 2.Python：应用中所有的 Python 日志
- 3.C：通过端计算提供的 C 接口打印的日志
- 4.OC/Java：通过端计算中 Java/OC 接口打印的日志

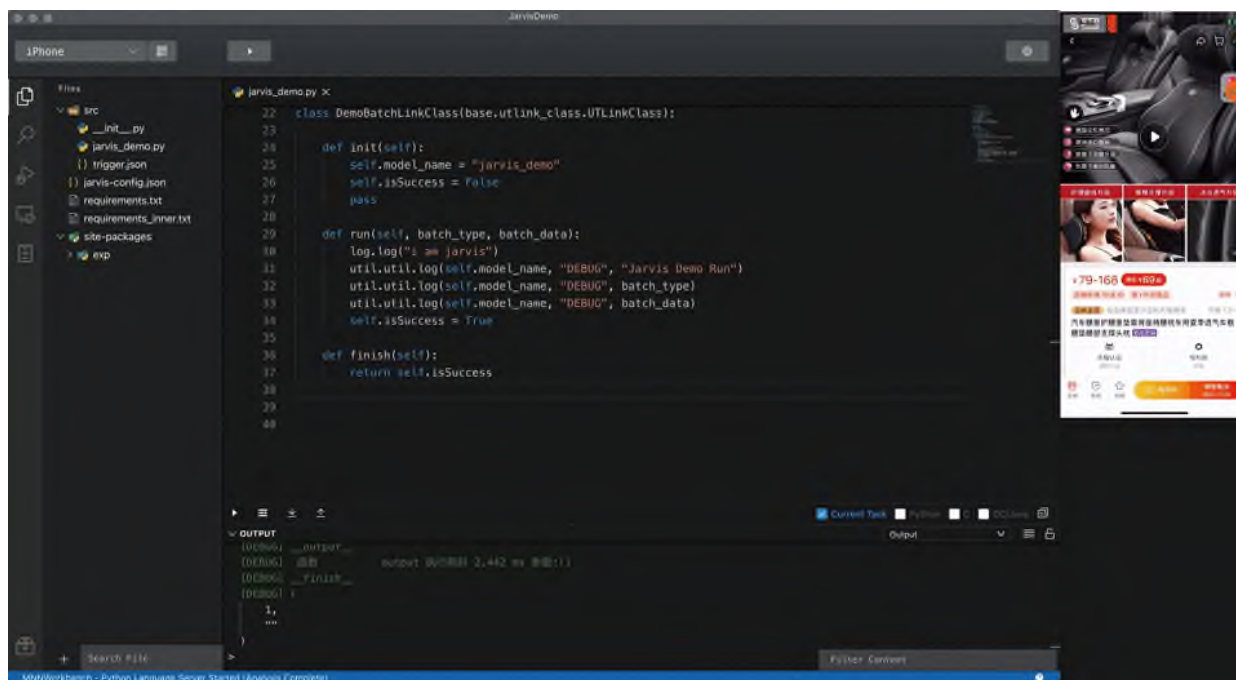
端文件浏览器



端计算研发中，经常需要查看或处理设备的本地文件，如数据库文件、深度学习模型、资源图片、脚本等等。工作台提供了可视化的设备文件浏览器，iOS 设备上可以查看应用的沙盒目录，Android 设备上的浏览和 Android Studio 一样，同时支持常用的目录/文件的本地保存和路径拷贝。

依赖三方库联调

有时算法需要去查看依赖库代码的运行情况，工作台同样能支持依赖 Python 三方库的研发，本地部署时会将算法指定的依赖库和工程一起推送到端上，调试的时候也能断点到库代码中。



上图展示了一个 Jarvis 工程依赖 Jarvis 基础库中的代码进行联调的过程，log.log 为 Jarvis 基础库中的打印方法。依赖二方库联调只影响到本工程，对其他工程不会产生任何影响。

视觉内容理解研发

Walle/Jarvis 都属于端计算的数据场景，除此之外很大一部分算法处理的都是视频、图片，最常见的是摄像头/视频流输入、内存中的图片，典型的算法如图像分类、目标检测等，他们都是算法在计算机视觉（CV，Computer Vision）场景中的具体应用，简称视觉算法。

CV 部署方案

端计算框架为数据算法提供了丰富了 Python 运行时扩展库，算法可以基于这些库开发各种 Python 计算任务，然后部署在端上执行。而针对视觉场景，图片和视频流的处理一般都涉及到计算机视觉库 OpenCV 以及函数计算库 Numpy，之前的端侧 Python 运行时库中并不包含，所以算法在 PC 上使用 Python 开发的视觉算法并不能直接运行在端上，所以一般都是使用 SDK 集成的部署方式：

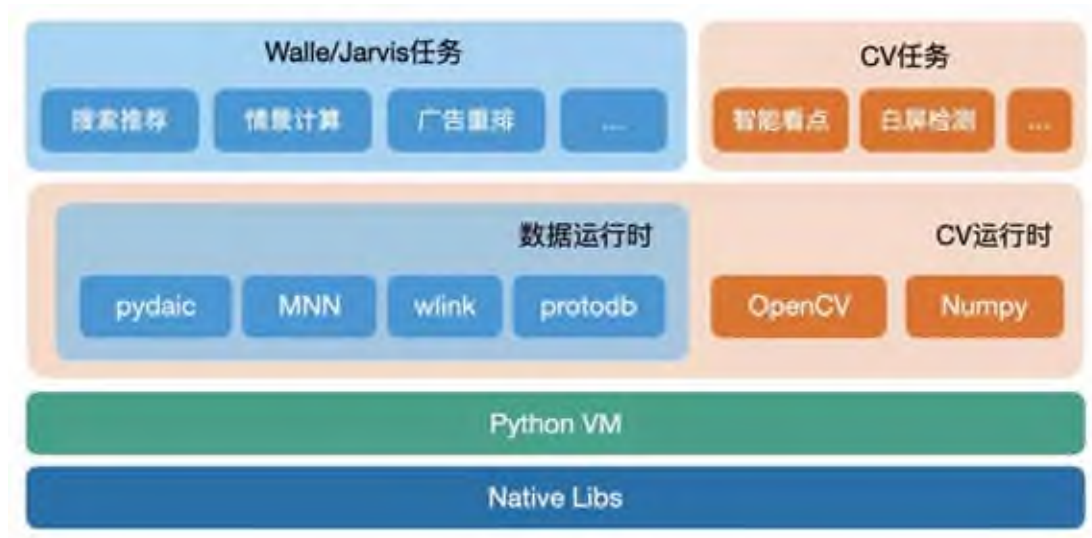


算法使用 Python 在 PC 环境开发，通常都会基于 OpenCV/Numpy 等基础库，移植到端上时，虽然端上这些基础还未支持，但可以通过一些方式转换为类似的 C/C++ 实现，如格式转换使用 MNN CV 替换 OpenCV 的实现，同时也要针对端侧的运行环境做一些性能优化。后面蓝色的部分都需要工程同学深度参与，封装 Android/iOS 上的 SDK，然后搭建本地的 Demo 应用验证，再集成到应用中。

其主要缺点在于：

1. 视觉算法需要针对端侧部分重新实现和优化，成本较高
2. 算法严重依赖工程协作，工程维护 SDK 的成本也比较大（以算法为核心的 SDK）
3. 算法的测试验证需要让工程开发 App Demo，且算法一般不能直接在手淘等应用环境中测试算法代码
4. 发布后一旦出现缺陷，无论是算法还是工程问题，迭代周期依赖应用的版本计划，一般超过 1 周

MNNKit 就是视觉算法最典型的 SDK 集成方式。



以往视觉算法的部署只能通过算法 C/C++ 的形式集成到应用，不过目前端侧运行时已经建设完善了 CV 运行时能力（OpenCV/Numpy/MNN/MNNRuntime），这样视觉算法就可以像一个普通 Walle 任务一样，通过工作台或发布平台动态地部署到端上，典型的业务如猫客中的白屏检测，淘宝直播中的智能看点等，都是通过这种方式部署的。

通过运行时部署 CV 算法的主要优势在于：

1. 算法可以自己搭建视觉验证 Playground Demo（见下文三端一体）
2. 和普通端计算任务一样，CV 算法使用 Python 开发，动态发布，迭代周期短
3. 可以使用工作本地部署，在应用中调试算法，预发/线上环境都可以

CV 研发流程



上图是使用工作台进行 CV 算法研发的完整过程，算法初步产出后，会经历三个研发测试阶段：

- 1.单元测试：算法在独立 Demo 中验证视觉效果的正确性；
- 2.预集成测试：算法在目标应用中和关联业务一起进行本地测试，因为算法还未使用平台部署方式，所以称为预集成测试；
- 3.集成测试：算法通过平台部署，预发回归测试，然后线上灰度/正式发布；

算法在上线前在平台进行集成测试，而单元测试和预集成测试都可以在工作台环境中进行。

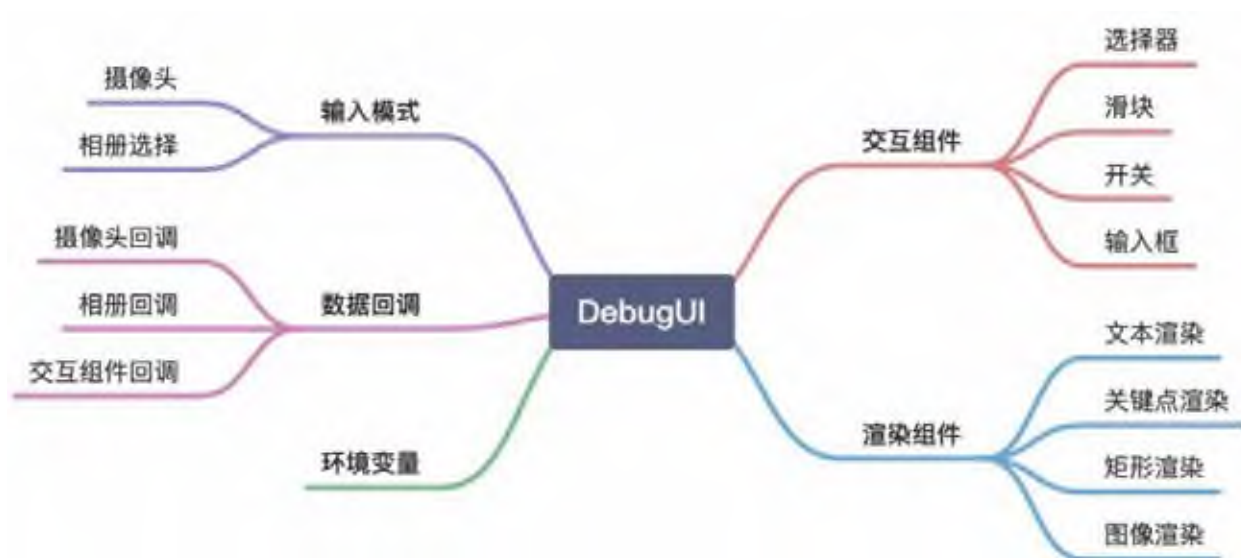
三端一体

与端计算数据算法有所不同的是，新开发的 CV 模型，一般都会先在本地进行效果验证，它是独立于集成的应用的，就相当于在 Demo 中的单元测试。比如你开发了一个图像算法，如目标检测，你想在端上看一下检测的结果对不对，定位的坐标准不准，那你就需要一个 Android/iOS 的应用，把模型和代码集成进去，然后再去开发上层代码（典型的如 MNNKit Demo）。作为算法并不了解 Android/iOS 开发，往往需要工程同学协助，协同成本较高。

有没有一种方式能让算法脱离工程，自己去写一个 Android/iOS 上的 Playground 验证应用呢？

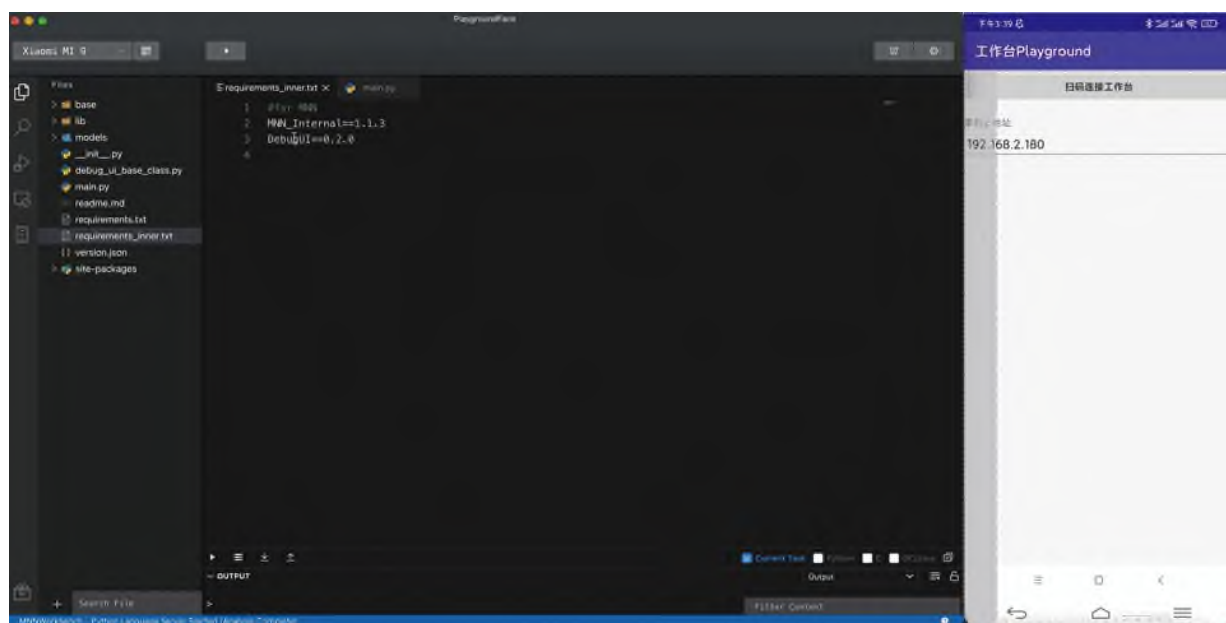
跨平台效果调试库 DebugUI

MNN工作台针对算法在模型效果验证中的痛点，提供了“三端一体”的解决方案。简单来说，我们提供了一套跨平台效果调试的 Python 扩展库 - DebugUI，算法可以使用 DebugUI 的 Python API，快速搭建一个视觉验证应用，并且能在一端编写，三端运行。



DebugUI 扩展库抽象和提取了视觉效果验证的核心链路，为算法提供了了精简又实用的 Python 接口，覆盖了目前视觉算法实际的验证场景：

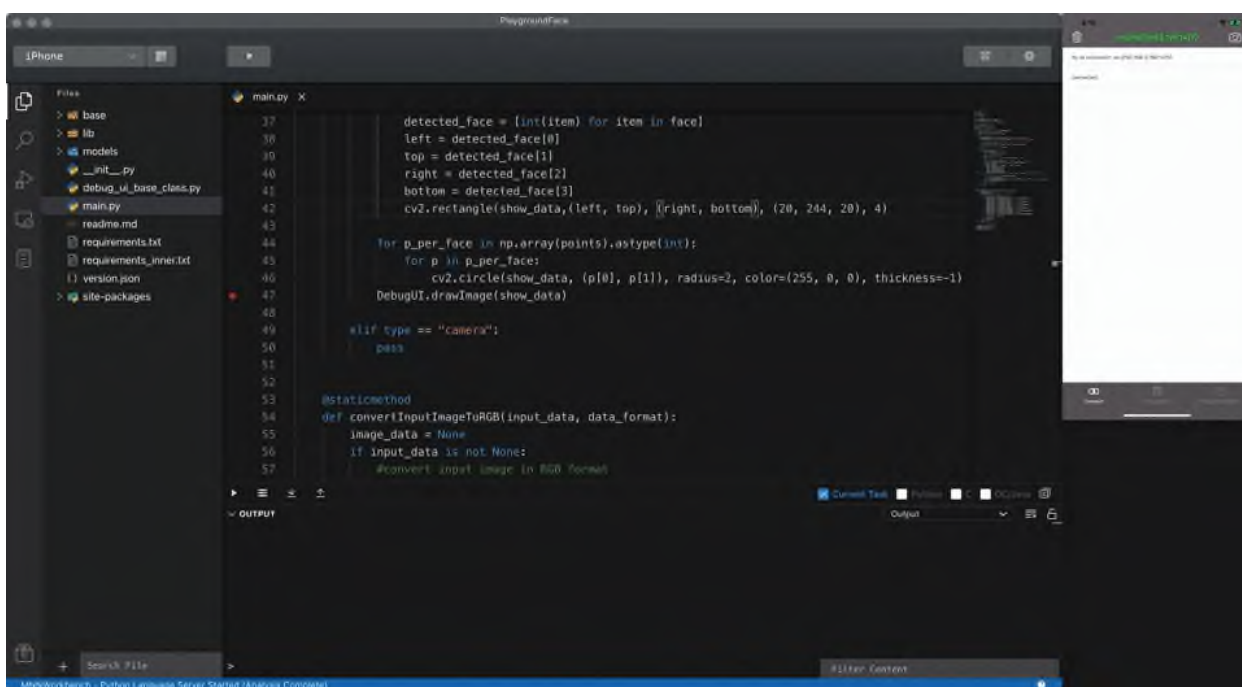
- 1.输入模式：一般数据来源只有图片和视频
- 2.交互组件：用于简单的控制，比如可以通过滑块调整阈值，通过选择器切换使用的模型，通过开关打开/关闭某个功能，等等
- 3.渲染组件：用于展示结果，比如分类得分值可以用文本渲染，人脸点位可以使用关键点渲染，人像分割可以用图像渲染，等等
- 4.数据回调：摄像头、相册选择，交互组件的事件，均会产生数据回调交给算法处理
- 5.环境变量：算法可以获取工程设置的变量，比如某个本地文件路径，等等



工作台模型市场内置了常用的一些开箱即用算法，如漫画脸、人脸检测、OCR、人像分割、卡通风格等等，它们都是算法使用 DebugUI 搭建的 Playground 示例应用。上图中展示的就是人脸检测算法的 Playground 示例，在 Mac、Android、iOS 上运行都能得到相同的检测结果，这就是“一端编写，三端运行”。

Playground 代码调试

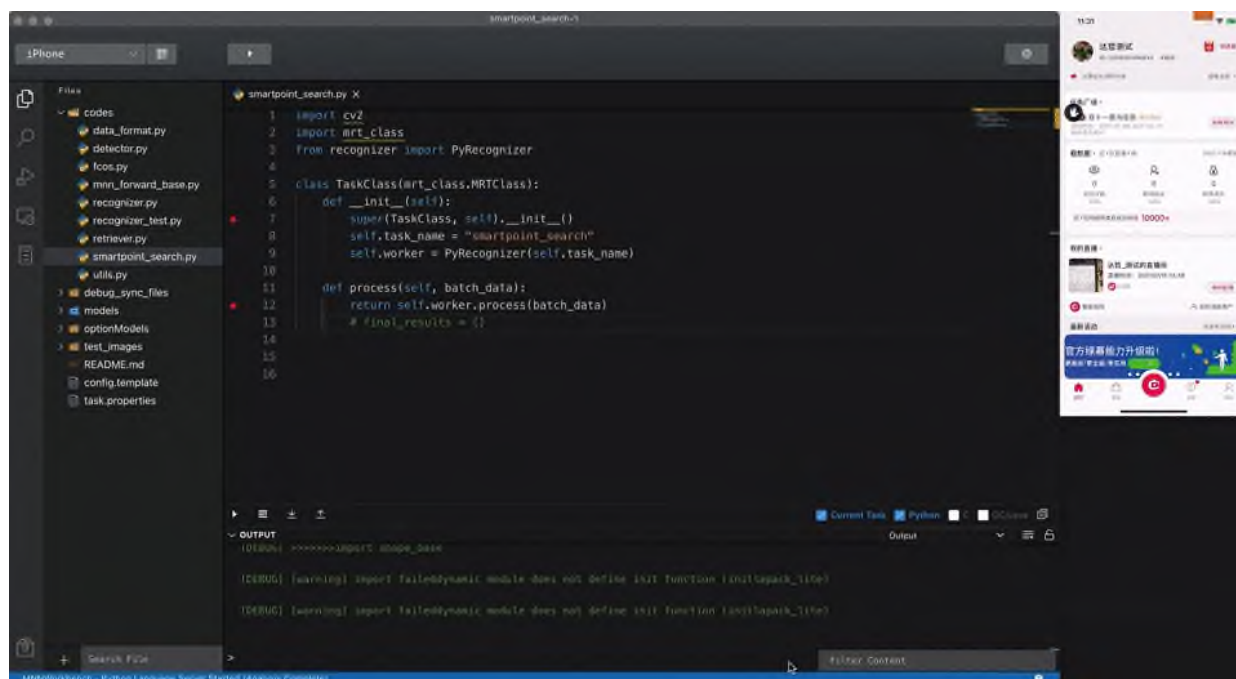
使用 DebugUI 搭建的 Playground 同样支持 Python 代码的断点调试，这对于算法在模型单元测试阶段的开发是很有帮助。



CV 算法调试

使用三端一体快速搭建视觉 Playground 应用验证算法效果后，接下来就是将算法模块通过工作台本地部署到集成应用（如手淘）中，应用环境会包含业务模块的主动调用触发算法，这样就可以在集成应用中进行算法调试了，这个就是预集成测试的阶段。

算法通过工作台本地调试通过后，在平台进行预发/beta的发布测试，这个就是走平台部署的方式了，只不过是在应用的预发环境部署，这就是集成测试阶段。集成测试通过之后，算法正式发布上线。

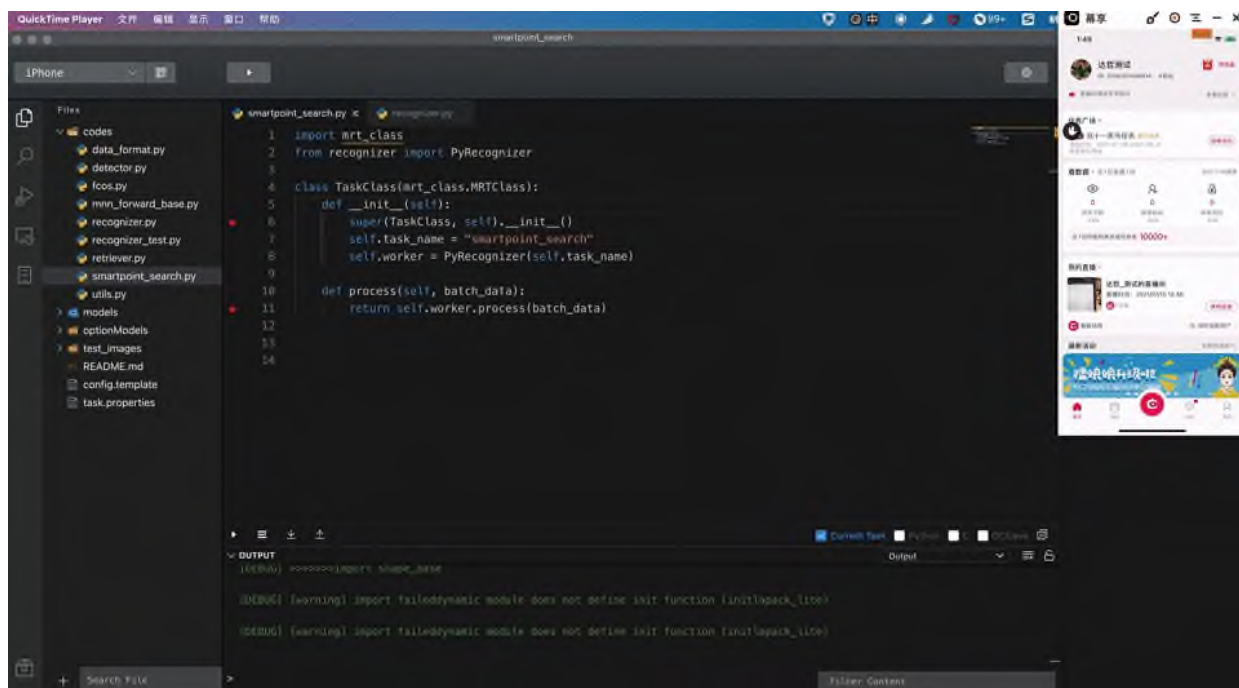


如图示例淘宝直播在预集成测试阶段，使用工作台进行 CV 算法的本地部署，工程主动调用算法的初始化、摄像头每隔几秒执行推理，都会触发进入到对应方法的调试断点，这样就可以方便的在应用中调试 CV 算法了。

性能测评

算法经常需要评测代码的运行性能以针对性的进行优化，尤其是对于 CV 实时算法。MNN工作台提供了端上 Python 代码运行实时的性能显示，通过“profile”方式运行 Python 工程，可以看到代码的执行路径，以及路径中每一行代码的执行时间和次数。这对于在实际应用环境中评测算法的运行性能，分析性能优化的瓶颈很有帮助。

如图示例了淘宝直播中使用 profile 来评测 CV 算法的性能，运行时间和次数会实时地显示在对应的代码行。profile 适用于所有能通过工作台部署的 Python 工程，Walle、Jarvis、CV、Playground 等工程都可以进行性能评测。后续我们会在性能评测中加入更多的端侧实时信息，如内存、CPU 占用等，帮助算法更好的分析代码性能。



结语

端智能的研发过程中，算法的开发只是一小部分，大部分工作其实都在算法之外，MNN 工作台不仅降低了普通开发者的 AI 研发的门槛，同时也是算法研发的提效工具，它有效解决了端 Python 的开发部署困难，让算法可以脱离工程自己搭建 Demo 应用，也支持了阿里集团内主要的端智能的研发场景，这样算法同学可以更加专注于算法本身の開発，从而提高 AI 的研发生产效率。

• 计算广告与推荐系统有哪些区别？

- 作者：铁丑
- 出品：淘系技术

在某角度上可以把推荐系统理解成计算广告的一部分。不考虑品牌广告等其他广告形式，在我看来，计算广告和推荐系统的联系在于业务上的融合和技术上的重合。



我接触到的计算广告一般以上图中两种业务形式体现：

一类是信息流中的个性化推荐，比如在短视频平台上的信息流广告，淘宝首页推荐的广告商品，一般叫展示广告或者推荐广告；

另一类是在搜索中的定向投放，比如Google百度搜索的内容广告，淘宝搜索结果页中相关广告商品，一般叫搜索广告。在这个层面上，计算广告依附与搜索推荐产品，这也是为什么很多人经常把搜索推荐广告放在一起叫搜推广或者搜推广的原因。

虽然计算广告和搜索推荐在产品形式上有融合，但是他们三者有本质的区别。搜索推荐可以理解成是为用户服务的产品，计算广告可以理解成是为商业服务的产品。也就是说，搜索推荐的目的是培养产品心智、提升用户体验的；计算广告的目的是平衡多方利益，持续赚钱，赚取更多的钱。

对于计算广告来说，本质上要处理的是用户、广告主、广告平台（媒体）这三方利益的协调问题。可以看成是一个充分博弈的商业产品，也正因为广告的商业属性，每个互联网公司都会投入大量的资源，因此计算广告相对来说也更严谨，也是技术发展最快速的，毕竟每一个点击率、转化率的提升，都是在提升公司的利润。

对于搜索推荐来说，本质上要处理的是用户和内容的匹配问题，电商领域就是用户和商品的匹配。这其中，搜索和推荐也有些差异，搜索的目标是为了让用户更快更好的找到想要的内容，需要重点关注精准度和覆盖度，特别是电商领域，我上架一个商品怎么搜都不到就不合适；推荐的目标是为了培养产品心智，留住用户，让你每天逛淘宝，一直刷抖音。

业务上的融合相对好理解，那怎么理解技术上的重合呢？

回顾一下，其实最早利用机器学习技术和思想的应用领域是搜索，尤其是近二十年兴起的搜索引擎，在此过程中也衍生出了一系列的算法和模型，其中最重要的排序学习算法（Learning To Rank）更是将搜索算法带到了一个新高度。近十年来由于场景不同，推荐系统和计算广告也快速发展，由于这三者要解决的问题都包括召回和排序的最优化，这也促使他们之间相互借鉴和启发。

抛开广告系统中竞价、审核、结算、作弊处罚中偏商业的这一部分。在投放层面，搜广推因为业务上的融合，也决定了他在技术上的高度重合，事实上大多互联网公司这三块业务也都是用的同一套工程系统。

对于系统工程师而言，无论是搜索推荐还是计算广告，他们都需要有召回粗排序、精排序、机制策略重排序，当然也包括移动设备上端云结合的边缘计算。召回粗排对应的是检索引擎，差异无非是基于关键词的搜索引擎还是基于协同过滤的推荐引擎；精排对应的是训练平台和打分引擎，主要还是点击率预估、转化率预估；重排序对应的重排服务，即面向场景的机制策略。

对于算法工程师而言，单纯对比计算广告和推荐系统可能不如直接对比展示广告和搜索广告来的直接，或者从搜索和推荐的角度看更合适。

1、召回Recall：召回阶段搜索是带有关键词的，文本匹配在其中发挥决定行作用；推荐系统因为缺少关键词所以会更多的挖掘用户的兴趣偏好，常用的召回策略包括，基于内容或者用户的协同过滤，Graph Embedding等向量召回技术和基于地域和时令的热门推荐。随着场景优化的深入和计算能力的提升，十几二十路的召回也已经是常规操作了。

2、排序Rank：排序阶段一般优化的都是CTR/CVR，业内比较普遍的排序模型比如传统的LR+GBDT，还有目前比较火的基于深度学习的序列建模。在这个层面三者差异都不大。

3、重排Rerank：重排阶段要更场景化一点，一般会根据场景的特有约束进行一些过滤、打散、加权、定坑等机制策略。

当然作为三种不同形式的业务的存在，他们之间肯定会有各自特有的技术栈。

对于搜索引擎来说，为了能够让用户更快更准的找到自己的想要结果，就会花很多工作在文本处理上，分词、Query理解、用户意图分析等很多NLP领域的相关技术会被更多的运用；在工程层面，由于搜索的数据规模巨大，这是推荐和广告不能比的，所以对搜索引擎检索效能的要求也更高，像索引压缩，索引截断等技术也算是比较成熟了。

对于计算广告来说，涉及到协调三方利益就会产生很多博弈。广告主之间为了竞争广告位就会有价格博弈，广告主和广告平台因为利益冲突也会有博弈，对应的就是出价策略（bidding strategy）。同时，广告平台在同一时段的广告位资源是有限的，为了在保证一段时间内整体收益最大，广告平台需要对广告进行排期，对应的就是排期优化（yielding optimization）。

对于推荐系统来说，为了培养产品心智、提升用户体验，不仅要考虑用户的短期行为，还要去挖掘用户的长期兴趣，要关注多样性、时效性和新颖性，对应的就是探索与利用（exploitation & exploration）和冷启动问题。在推荐中power law非常明显，这种长尾效应必然会带来训练数据倾斜，如果不处理用户能够反馈的内容会越来越集中，对应就是稀疏数据处理。

● 目前工业界常用的推荐系统模型有哪些？

- 作者：豆苗
- 出品：淘系技术

一般来说，工业界推荐系统采用召回->排序两阶段的架构。

召回阶段从海量内容池中召回数千条内容生成候选集，排序阶段利用用户、内容侧丰富的特征、上下文信息和复杂的模型对候选集中的内容进行打分排序，最终为用户返回数十条内容。

本人主要工作在召回侧，所以下面简单列出召回相关的策略和模型。

召回环节的优化通常经历三个阶段：**基础个性化召回**->**深度个性化召回**->**多目标召回**。

基础个性化召回：

基于统计和传统模型（如swing、userCF、itemCF等）产出各种x2i数据，进行以i2i为主，c2i、b2i补充，热门兜底的多路召回，各召回链路可单独优化；

深度个性化召回：

将user、item转化为向量，通过向量检索技术召回topk，变“精确匹配”为“模糊查找”，提升召回的扩展性。首先需要尝试的是FM，可以参考

张俊林：推荐系统召回四模型之：全能的FM模型

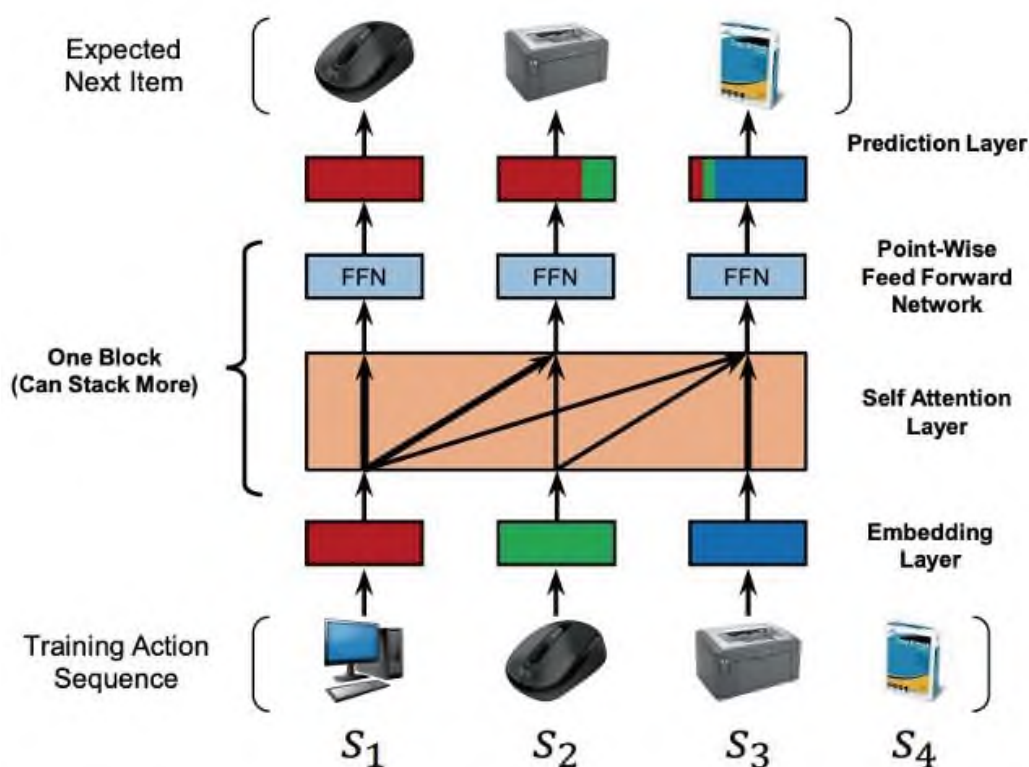
2792 赞同 · 336 评论 文章



还有基于用户行为序列、基于多兴趣拆分的、基于graph embedding和和基于知识图谱融合的。

用户行为序列：

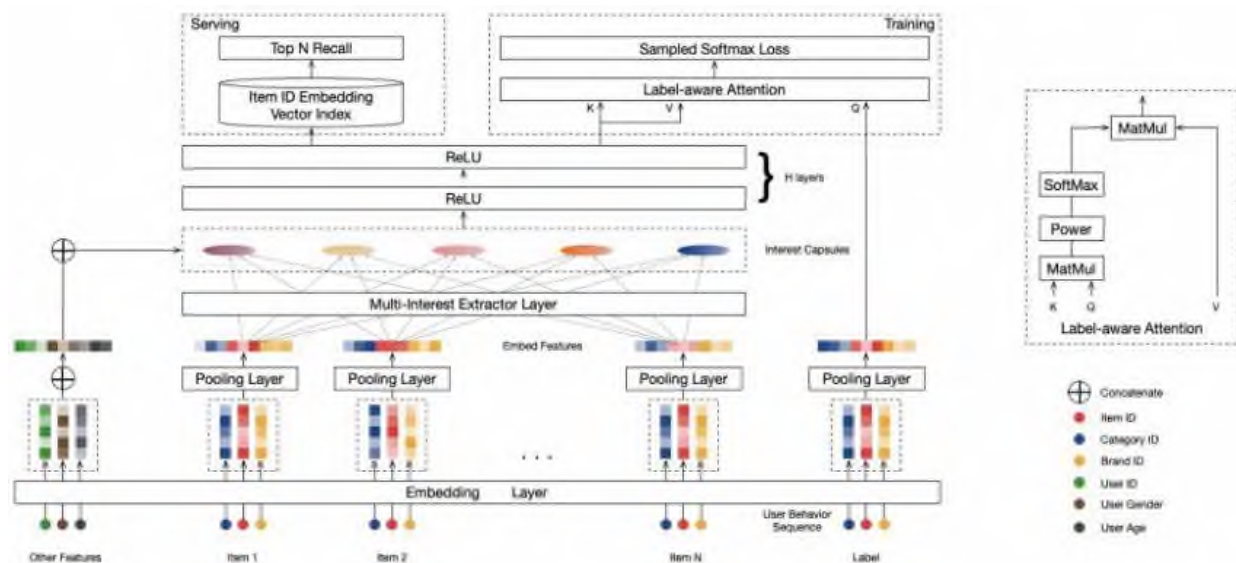
这类模型代表性的方法是SASRec，这是18年的发表在ICDM上的一篇针对召回的工作，主要是借助transformer来对用户的历史行为序列建模，提取更为有价值的信息。整个模型基本上是套在transformer这个框架下的，所以有position embedding、block的stacking等等，需要注意的是作者也有作一些调整，比如只有当 $j > i$ 的时候 Q_i 和 k_j 才会进行相关性的计算，并且采用非固定的Position embedding。



附上论文：<https://cseweb.ucsd.edu/~jmcauley/pdfs/icdm18.pdf>

用户多兴趣拆分：

这类模型核心的思想就是认为单一的向量无法充分表征用户，从而采用多个向量表征用户的多峰兴趣，最为典型的的就是MIND，它利用动态路由技术，对用户兴趣进行拆分。模型的输入包含三部分：用户属性特征、用户行为特征（例如点击过的商品/内容）和label特征。首先，所有的id类特征都经过embedding层，其中对于用户行为特征对应的embedding向量还会进行average pooling。接着，传递给multi-Interest extract layer，生成的interest capsules。最后将interest capsules与用户属性特征的embedding concatenate起来经过几层全连接网络，得到用户多个兴趣表达向量。值得注意的是，在模型的最后有设计一个label-aware attention层，用以针对每个用户从多个兴趣中挑选出与其下一次点击行为最相关的兴趣，送入到后续的训练。



附上论文： Multi-Interest Network with Dynamic Routing for Recommendation at Tmall

其他相关的多兴趣拆分的方法有：

基于注意力机制

基于层次凝聚聚类

Graph Embedding

这类模型是近期推荐系统召回环节较为热门的研究方法，并且也在线上拿到了收益。例如EGES、PinSAGE、PinnerSAGE等。

多目标多策略召回

在以点击率为优化目标的基础上，尝试优化发现性、成交率等其他目标。以及将积累的各种召回模型，根据业务实际需求实行多策略召回。

• 2021年，作为算法工程师的你们会在CV业务落地上用Transformer吗？

- 作者：继科
- 出品：淘系技术

CV领域的多条赛道现在都被transformer霸榜，不玩transformer都不好意思说自己是搞CV的，但业务上真的需要transformer吗？作为淘系的一线技术小二，这里分享一点自己的落地心得。

先说结论：判断新技术能否实现商业落地，本质上都是ROI的问题。Transformer的ROI还远未达到能够落地的程度。

一、Transformer带来的技术进步程度还不够高

@花花 的回答中有句话很形象：“我认为目前transformer在CV领域应该还处于AlexNet阶段，距ResNet阶段还有一定的距离”。Transformer能够在公开数据集上提升多个CV任务的精度，已经是公认的结论了。然而，其技术进步程度还要从以下几个角度来衡量。

首先，transformer对long sequence的建模优势就不说了，CNN在这块是天然的劣势。但transformer目前带来的精度提升，既来自self-attention的理论优势，也不能否认模型参数规模的贡献。CNN当年也是这样一路走来的。VGG那样的参数量也是随着模型结构研究的进展而降低到现在MobileNet、YOLO系列的水平。这个过程其实是研究人员对模型结构理解逐步深化、去芜存菁的过程。目前已经出现了许多针对transformer结构效率优化的研究，例如轻量级attention（e.g. 稀疏attention变体）、分而治之的方法（e.g. 循环和分层机制）等。这些研究能否解决transformer模型大、效率低的问题，也是产业界关注的重点之一。

其次，在工业数据集上，transformer带来的精度提升也还处于量变阶段，尚不能对核心业务带来显著的增量价值。举例来说，图像分割算法在淘宝的自动化内容生产中扮演着重要角色。目前最新基于transformer的语义分割SOTA方法，在公开数据集上的mIOU提升幅度有2个百分点左右。然而，若使用淘宝的人工审核通过率（可以简单理解为审核人员一眼看上去能否发现错误）作为评判标准，则对整体的通过率基本没有影响，从而也无法提升审核效率或降低审核成本。

此外，transformer的理论研究目前还处于百花齐放的阶段，需要一个类似ResNet这样的milestone来作为学术与工业界的连接点。一种新的结构要进行产业落地，需要上下游进行非常多的相关开发，例如移动端推理框架对新算子的支持与性能优化、训练集群规模的扩大与调度、训练数据的管理与共享、模型的压缩与分发等。目前

在不同CV任务上，基于transformer的解决方案还存在很大差别，这导致配合的算法工程团队难以着手，从而无法落地应用。

二、Transformer带来的成本增加是巨大的

虽然transformer结构会对CV任务带来一定的精度提升，但同时，使用transformer也需要付出一定代价。仅举几例如下：

- 1、Transformer模型巨大的参数量将成倍地增加服务器成本以及随之而来的运维难度。
- 2、Transformer在小数据集上很难训练，极易过拟合，因此训练所需的数据量显著增加。并且，有些业务场景天然难以提供大量数据。
- 3、Transformer的训练时间长，相比CNN，transformer有时甚至需要3~5倍的训练时间，这一方面阻碍了许多需要快速响应的需求的开发（例如某些热点事件的跟进），同时也隐形地增加了公司的人力成本（阿里的算法工程师工资可是很高的哟）。

这些成本增加能否接受，则取决于应用该技术的业务是否具有足够的价值了。国内各大互联网公司中，都有transformer的落地先例，例如阿里的搜索推荐、美团的搜索排序、腾讯的广告推荐等。这些无一不是能够带来巨大商业价值的应用场景，在其中取得很小的精度提升，就可以cover相应的改造成本。能否找到类似价值体量的应用场景，也许是transformer能否顺利落地在CV领域的关键。

前端篇

• 新一代图片编解码技术在淘宝的应用及落地

- 作者：岳溪
- 出品：淘系技术

本文回顾淘宝图片发展的历史，阐述了新一代图像编解码格式AVIF在淘宝业务场景中的应用及落地方案，节省流量，为用户提供更好的看图体验。

背景

淘宝图片空间下行链路承载着集团图片的访问请求，包括手淘、飞猪、闲鱼等业务场景，核心工作在于提升用户访问体验，在图片质量不降低的情况下降低访问带宽。常态下CDN侧千万级别 QPS（每秒请求数）图片访问，未命中时的回源量也在百万级别，通过引入新的编码技术，可大幅降低访问带宽，节省流量成本，同时体积的减小也能带来访问提速，提升用户看图体验。

淘宝图片空间下行链路架构

淘宝图片访问模型如图1所示，用户发起请求，在CDN（content delivery network，用于图片缓存）缓存未命中的情况下，请求回源到源站，源站从OSS（object storage service）存储上拉取商品原图，按照用于请求格式进行实时处理（包含解码、缩放、编码等过程， $RT < 100ms$ ）后返回，常态下CDN命中率约为95%，此时用户访问的RT小于20ms。

由于访问端（PC、手机）及场景（首页、详情页）的多样性，相同的图片存在多种尺寸、格式，如淘宝首页访问的图片尺寸相对较小，而详情页尺寸较大，通过在图片源站侧进行实时计算（本质上是计算换存储的方式），支撑相同图片内容的多种格式访问。

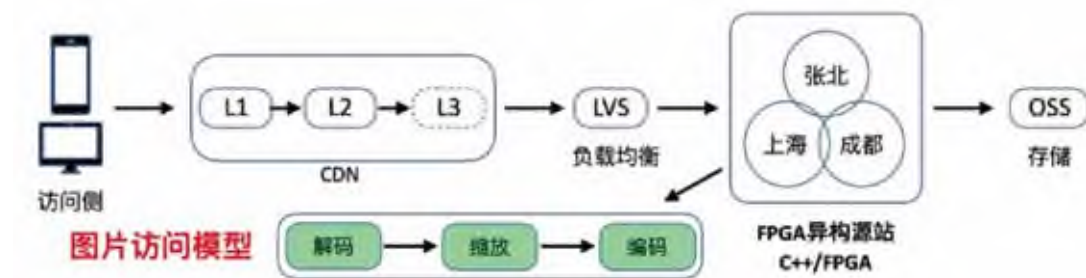


图1：图片空间下行链路架构

回顾过去的工作，我们对淘宝图片下行处理做了大量的优化：

2018年，通过引入FPGA异构计算，HEVC关键帧图片编码格式开始在淘系大规模商用，相比WEBP编码格式，降低带宽约30%，相比JPEG降低超50%，采用高密度存储、异构计算技术的新集群规模只有老集群的十分之一，高密度存储、异构计算分别使单位存储、计算成本降低超80%；

2019-2020年，图片空间FPGA集群极致优化，通过将JPEG解码、PNG解码、高质量SCALER、WEBP编码 Offload至FPGA，同时利用Nginx多worker负载调配实现FPGA板卡IP的负载均衡，单机性能由2000提升至5500 QPS，在保证图片质量的情况下，计算成本降低超60%，同时处理耗时大幅降低；

2020年，FPGA集群容器虚拟化，通过设备虚拟化技术，采用设备直通方式实现PCIe设备（FPGA板卡）的虚拟化。接入集团云原生应用统一基础设施ASI（Alibaba Serverless infrastructure），可根据业务流量快速调整不同服务集群的比例，实现FPGA服务间的快速扩缩容。最终业务部署至ASI pouch集群，为业务的灵活部署及调度提供了可能。

基于上述工作的积累，我们开始寻求在压缩率及用户体验上更进一步的方案，新一代视频图像编码技术进入项目组视野。

新一代图片编解码技术

技术选型

以Google为首的多家公司成立的AOM（Alliance for Open Media，开放媒体联盟）组织，制订了一套替代HEVC技术的免费开放视频编码标准AV1（AOMedia Video 1）。随着18年AV1格式的定稿，其图片格式AVIF在18年12月份由Netflix推出，2020年8月chrome及随后的Firefox 86版本已支持AVIF，即将推出的Android 12也将原生支持其解码显示。鉴于其以下特点，我们选择AVIF作为下一代图片格式，进一步降低带宽，提升用户体验。

- 高压缩：压缩率相比heif能提升25%-30%（通过测试，相同质量下，AV1相比线上HEIF压缩收益超30%）；
- 互联网生态较好：netflix内容、chrome平台均已开始支持，android也开始支持AV1/AVIF解码，mac M1开始支持AV1视频解码，不仅端侧可支持，web端也可将图片格式由webp升级至AVIF；
- 专利free。

进行图片压缩率测试，结果如图2所示，原图为无损编码格式，分别对比不同PSNR下VVC、AV1不同工具以及

手淘目前在用的HEIF格式，考虑到PSNR高于40dB时编码后的图像质量非常接近原始图像，我们选取39dB和41dB（线上平均为41-42dB）计算压缩率指标，相比HEIC格式，AV1的压缩率提升约26%-32%。

算法	psnr-39(/109)		psnr-41(/108)	
	bpp	ssim	bpp	ssim
原图(/113)	7.35859	--	7.23652	--
vvc(/113/113)	0.58138	0.97179	0.71791	0.97973
av1_0_9(/111/112)	0.64178	0.97206	0.78335	0.97989
av1_0_5(/110/112)	0.64273	0.97207	0.78392	0.97988
av1_0_all(/111/113)	0.64278	0.97206	0.78398	0.9799
av1_0_12(/111/113)	0.64502	0.97196	0.78681	0.97981
av1_0_4(/110/112)	0.64937	0.97187	0.79156	0.97975
av1_0_7(/111/112)	0.65136	0.97184	0.79376	0.97975
av1_0_1(/110/112)	0.65174	0.97207	0.79307	0.97984
av1_0_10(/111/112)	0.65561	0.97079	0.79749	0.97917
av1_0_13(/111/113)	0.66764	0.97233	0.81233	0.98011
av1_0_6(/111/112)	0.66909	0.97152	0.81276	0.97953
av1_0_8(/111/112)	0.66909	0.97152	0.81276	0.97953
av1_0_2(/110/112)	0.6727	0.97226	0.81938	0.98009
av1_0_3(/110/112)	0.69923	0.97051	0.84776	0.97898
av1_0_11(/111/113)	0.70407	0.9709	0.85049	0.97918
heif(/110/110)	0.96287	0.97129	1.1583	0.97963

图2：不同格式压缩率测试



图3：压缩率对比

IE	Edge	Firefox	Chrome	Safari	Opera	Safari on iOS	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
		2-76	4-84		10-70											
6-10	12-90	7-86	85-90	3.1-14	71-76	3.2-14.4		2.1-4.4.4	12-12.1				4-13.0			
11	91	89	91	14.1	77	14.6	all	91	62	91	89	12.12	14.0	10.4	7.12	2.5
		90-91	92-94	15-TP												

图4：浏览器对AVIF的支持情况（绿色表示已支持）

除上述特性外，AVIF能支持动图、PQ/HLG等高动态EOTF场景（兼容高亮度高动态显示器）、BT.2020广色域、10bit/12bit显示、alpha/depth图，为后续的场景效果扩展提供便捷。

技术特点

高压压缩收益带来的是计算复杂度的提升，AVIF使用AV1关键帧编码技术，通过帧内预测去除图像的空间冗余以提升压缩率。包括方向预测的粒度进一步升级，而非仅仅是方向性的预测，同时纳入了梯度和相关性，亮度的一致性和色度也得到充分利用，主要包括以下技术特点：

帧内预测方向的增强

为了在方向纹理中实现更多种类的空间冗余，在AV1中，将方向帧内模式扩展到更具精细粒度的角度集。将最初的VP9帧内编码的8个角度设为名义角度，基于这些角度引入步长为3度的精细角度微调，即预测角度由名义内角度加上角度增量表示，该角度以 $-3 \sim 3$ 的倍数为步长。为了以通用方式在AV1中实现定向预测模式，这48个扩展模式由统一的定向预测器实现，预测器将每个像素链接到像素边缘的一个参考像素位置，并通过2 tap的双线性插值对参考像素进行插值，最终AV1帧内编码共启用了56个定向内部模式（HEVC帧内包含33个角度+DC+Planar）。

无方向平滑的帧内预测器

AV1通过添加3个新的平滑预测器SMOOTHV，SMOOTHH和SMOOTH扩展了无方向帧内模式，它们在垂直或水平方向，使用二次插值或其平均值预测了像素块。

基于递归滤波的帧内预测器

为了捕获边缘参考块的衰减空间相关性，FILTER INTRA模式通过将亮度块使用二维不可分离的马尔可夫过程。AV1预设计的五个滤波器帧内模式中，每个模式用一组八个7-tap滤波器表示，反映 4×2 色块中的像素与相邻的7个像素之间的相关性。一个内部块可以选择一种帧内滤波模式，并以一批 4×2 补丁进行预测。

从亮度预测色度

Chroma from Luma (CfL) 是仅用于色度的帧内预测器，通过以重建后亮度像素的线性函数建模得到色度像素。重建的亮度像素被二次采样为色度分辨率，然后去除直流分量以得到交流分量。为了从交流分量中估计色度的交流分量，AV1 CfL根据原始色度像素确定参数并在比特流中用信号发送。这降低了解码器的复杂性并得到了更精确的预测。对于DC预测，它是使用内部DC模式计算的，该模式对于大多数色度内容来说已经足够，并且具有成熟并能快速实现的方法。

调色板作为预测指标

对于屏幕截图和游戏之类的人造视频，可以用少量颜色的近似块来替代，AV1帧内编码引入调色板模式进一步提升压缩率。

帧内块拷贝

AV1允许其帧内编码器参考同一帧中先前重建的块，类似于帧间编码器引用前一帧中块的方式，对于包含重复纹理的屏幕内容视频非常有用。IntraBC新预测模式的引入，在当前帧中拷贝一个重建的块作为预测。参考块的位置由位移矢量指定，类似于运动补偿中的运动矢量压缩。

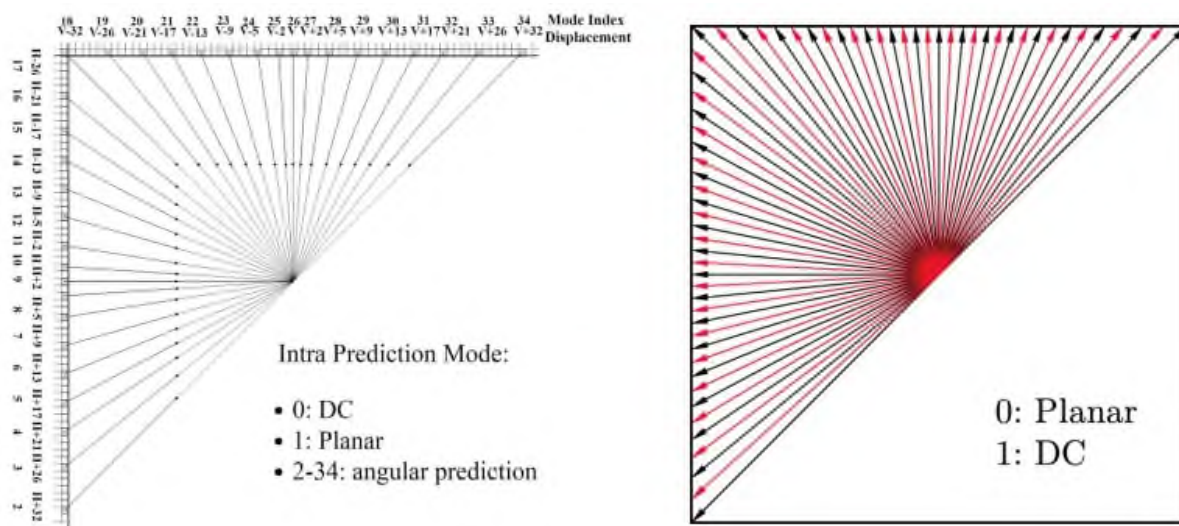


图5：HEVC与AV1帧内预测

落地计划及服务端方案设计

复杂的帧内编码需要耗费大量的计算，软件方案编码一帧图像耗时达到了秒级，显然无法满足实时处理的需要，大规模落地必须借助专用硬件方案，其开发工作是项目的重点和难点。另一方面，新格式的落地也需要客户端解码的支持，前期计划先基于自研高性能解码器进行解码，后期考虑切到系统原生解码（Android 12支持AVIF解码）。

落地计划

项目落地上按4个版本迭代，核心的开发包括服务端软硬件编码以及客户端的解码，服务端采用先软件后硬件的思路，通过基础版本、全功能版本、稳定版本、优化版本的迭代实现新格式的平滑升级。

里程碑节点	基础版本	全功能版本(独立灰度)	稳定版本(小流量灰度)	优化版本(逐步放量)
业务目标	服务端软件编码 Android端软解码	服务端硬件编码 Android端软解码 cdn回源至预发机器	服务端硬件编码 Android端软解码 cdn回源至线上机器	服务端硬件编码 Android端软解码 cdn回源至线上机器
评价指标	相比heic压缩率提升20% 图像质量效果无异常 服务端RT不做要求	相比heic压缩率提升20% 图像质量正常 服务端RT < 100ms	相比heic压缩率提升20% 图像质量正常 服务端RT < 100ms 测试通过(压测、稳定性、用户规模)	相比heic压缩率提升25% 图像质量正常 服务端RT < 100ms

图6: 项目落地计划

服务端方案设计

服务端沿用图片空间服务端图片转码架构, 使用阿里云AIS自研的AV1硬件编码器进行服务端编码, 集成AV1/HEVC/WEBP等多种硬件编码IP, 业务部署至ASI Pouch集群, 同时进行新格式的线上全量PSNR校验, 如有异常可一键降级为HEVC编码方案。

系统架构

服务端完成图像的转码(如JPEG转码为AVIF), 系统分为Nginx统一接入、资源解析、图像处理、FPGA硬件加速等模块, AVIF编码Offload至FPGA, 为防止硬件图像处理出现异常, 在图像处理层对硬件编码的数据进行decoder, PSNR对比编码前和解码后的YUV数据, 确保硬件处理的正确性。

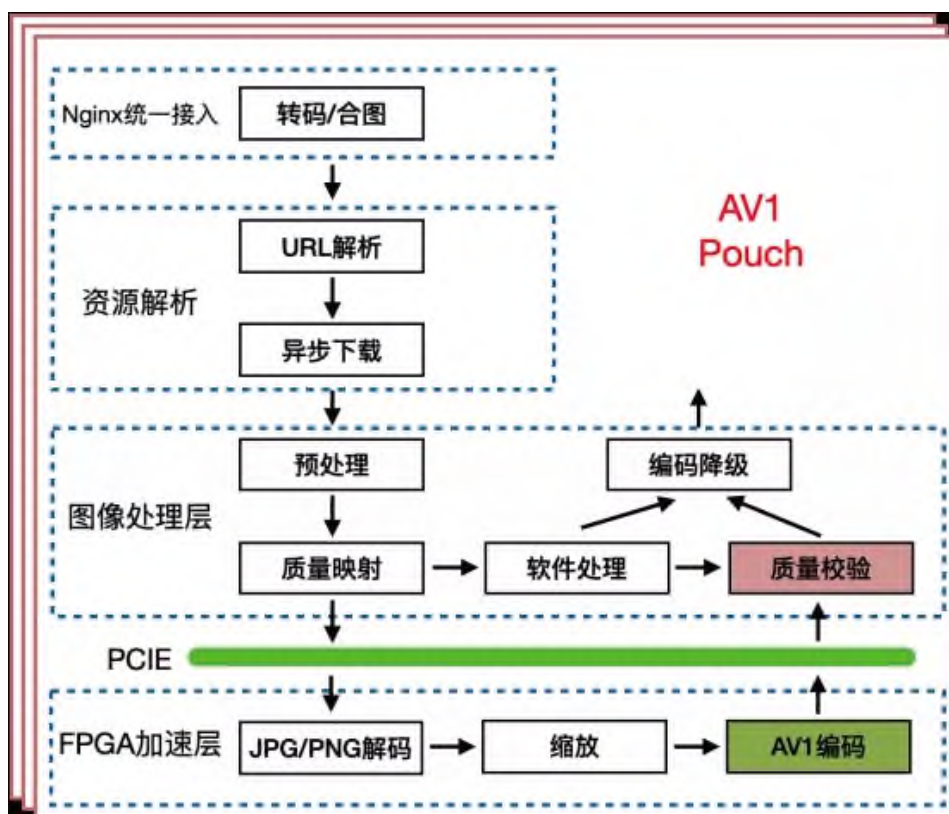


图7：服务端架构

在不同Pouch上部署不同的FPGA逻辑版本，做到新老版本的隔离，不影响老业务稳定性的情况下，做到新业务的快速迭代。

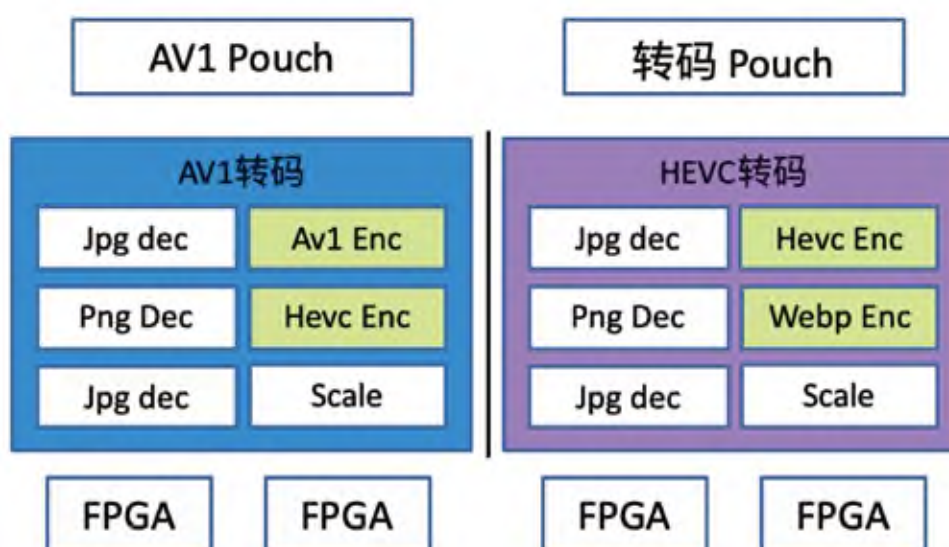


图8：容器化FPGA部署

稳定性保障

图像质量和稳定性是追求极致图像体验的前提，图片基础设施面向全集团提供图片服务。线上是否有花图，图片是否清晰，用户请求是否有超时，新版本效果是否达标等问题出现在我们面前，必须确保高并发下图片质量的稳定。

通过梳理图像主客观质量测试、线上预警及自动化分析流程，从质量回归、在线PSNR抽样检测、回放压测三个方面来监管系统稳定性：

- 1.质量回归：通过定时脚本每日筛选大量线上日志，分别请求预发与线上，查看PSNR/SSIM、图片格式、图像体积、是否有图等指标报告，定位处理图片质量问题；
- 2.在线PSNR校验：线上FPGA机器有一定的异常概率（如板卡损坏、FPGA掉电），通过各处理模块的在线抽样校验，可及早知晓定位问题，降低风险；
- 3.回放压测：图片下行系统为多核C + FPGA处理系统，通过长时间的回放压测，规避host/chip端内存泄漏及硬件概率性风险；

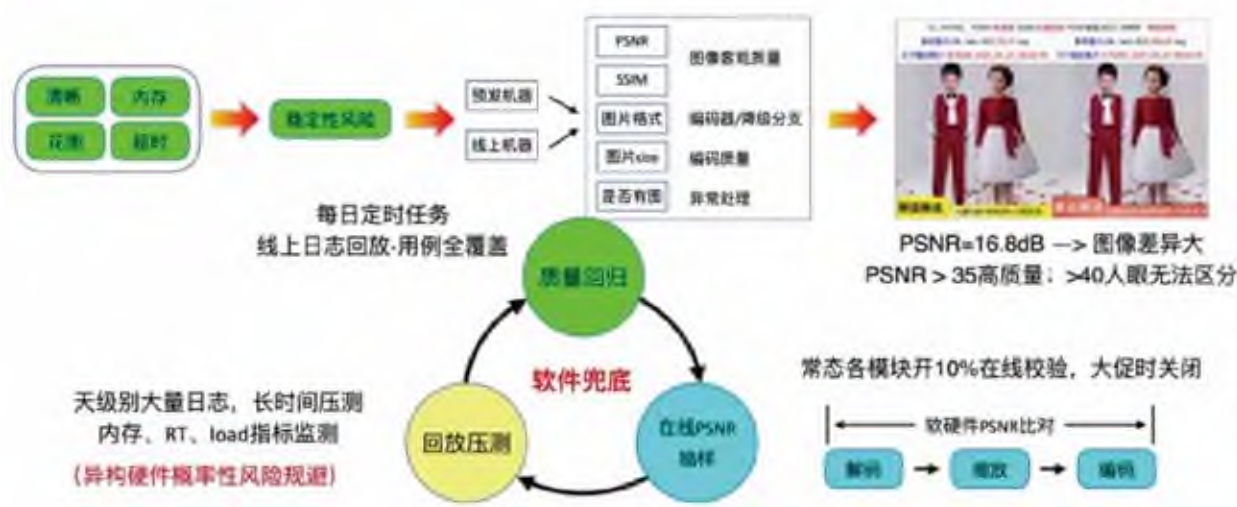


图9: 图片质量保障方案

落地结果

在服务端进行转码后输出AV1编解码图像格式，手淘上对其进行解码。

展望与致谢

图像格式的发展与编解码技术发展息息相关，基本处理十年出一代的节奏，回顾过去的JPEG->WEBP->HEVC格式的发展历程，高的压缩率格式的落地无一不是算法与硬件结合的结果。

得益于团队在算法硬件加速与工程落地的经验，项目组开始将AVIF图像格式进行工程落地，同时由于AVIF格式的良好生态以及扩展特性（目前各大主流浏览器均已支持），我们计划在PC及手机端全线使用AVIF图片格式，同时覆盖动图场景，并做到不同设备平台的高保真支持，希望通过技术升级，进一步降本提效。

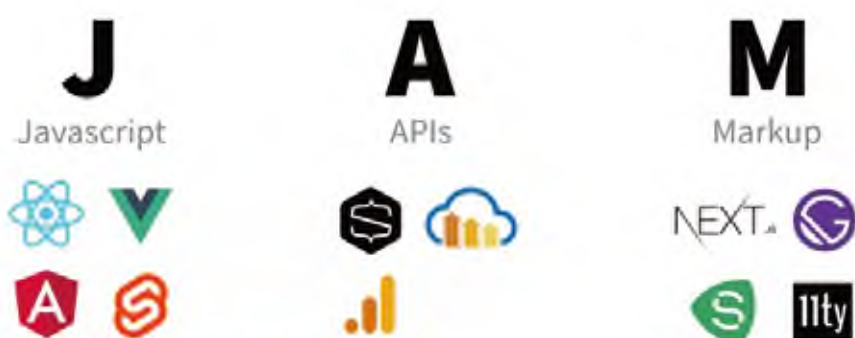
目前软硬件加速方案初始版本已经开始使用，高性能版本（算法及时序优化）开发完成后，再来更新相关信息，敬请期待。

感谢阿里云AIS、淘系图片空间、手淘图片库、阿里云CDN团队相关同学的大力支持。

• 下一代 Web 建站技术栈——Jamstack

- 作者：繁易
- 出品：淘系技术

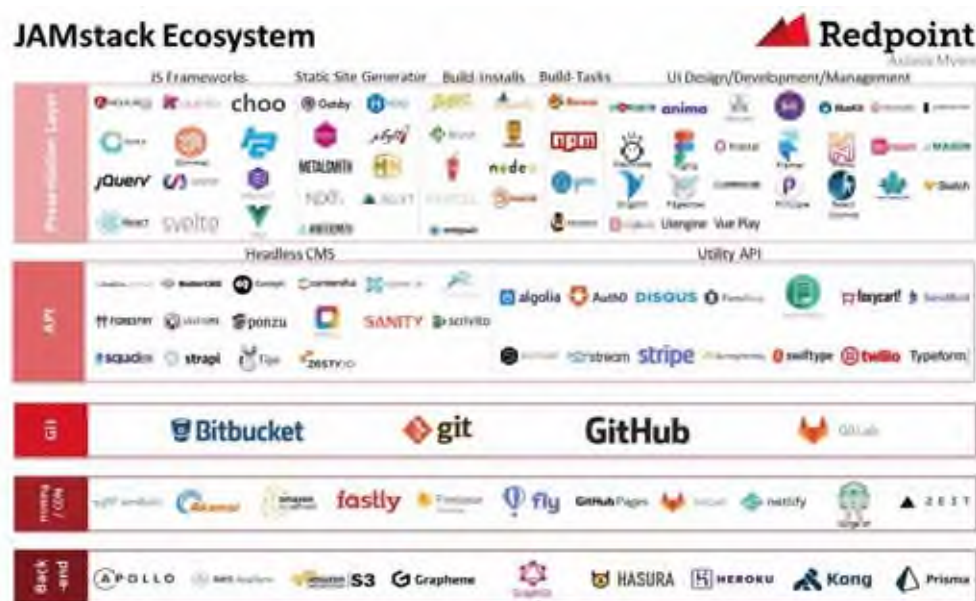
如果你之前从未了解过 Jamstack，我推荐先阅读文章：Jamstack，下一代Web建站技术栈？（<https://zhuanlan.zhihu.com/p/281085404>）。



Jamstack 是什么？

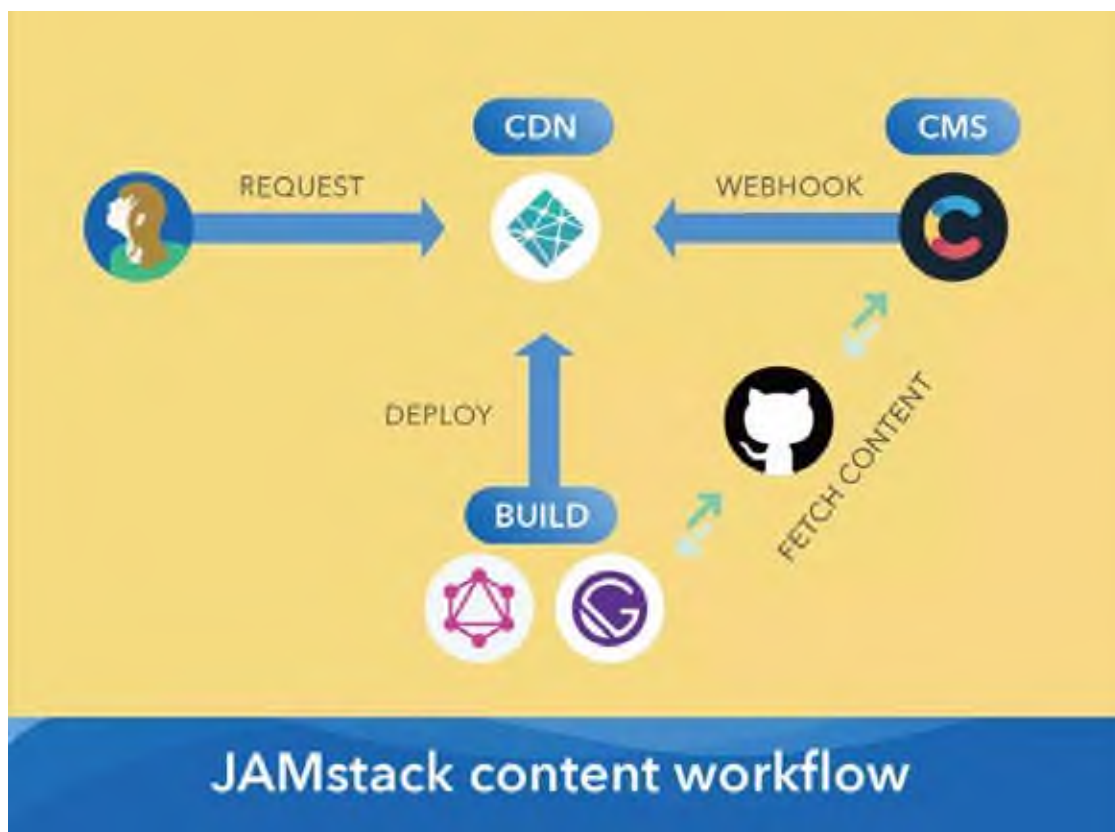
Jamstack 是一套用于构建现代 Web 站点的技术栈，拥有高性能、安全性、易扩展的特性。

Jamstack 技术栈 & 生态



Jamstack 聚合了现代前端开发所需要的脚手架，框架，工作流等，从而最大化的提高工程师的生产力。

工作流



在这里，Jamstack 的核心理念是预渲染、使用 JavaScript 实现动态功能、使用 HTTP Api 连接第三方服务。

举个例子

当你要开发一个博客，在这之前你可能会使用 Wordpress 去搭建你的博客站点，但与此同时，你也需要负责维护 Wordpress 的服务与数据库等。

而如果你使用 Jamstack，你可以使用诸如 strapi 的 headless cms 服务(意为只提供 API 而不提供页面渲染)，用来存放你的文章数据，你在前端可以使用类似 Next.js 的框架去构造站点，通过请求 headless cms 的 Api 来渲染页面。

而在发布时，你将在构建时生成静态页面，并发布至 CDN。因为是静态页面，所以性能好，而托管至 CDN 意味着该页面是只读的，安全性高，且 CDN 是全球部署的话，那么页面也能实现全球部署，拓展性非常好。

误区

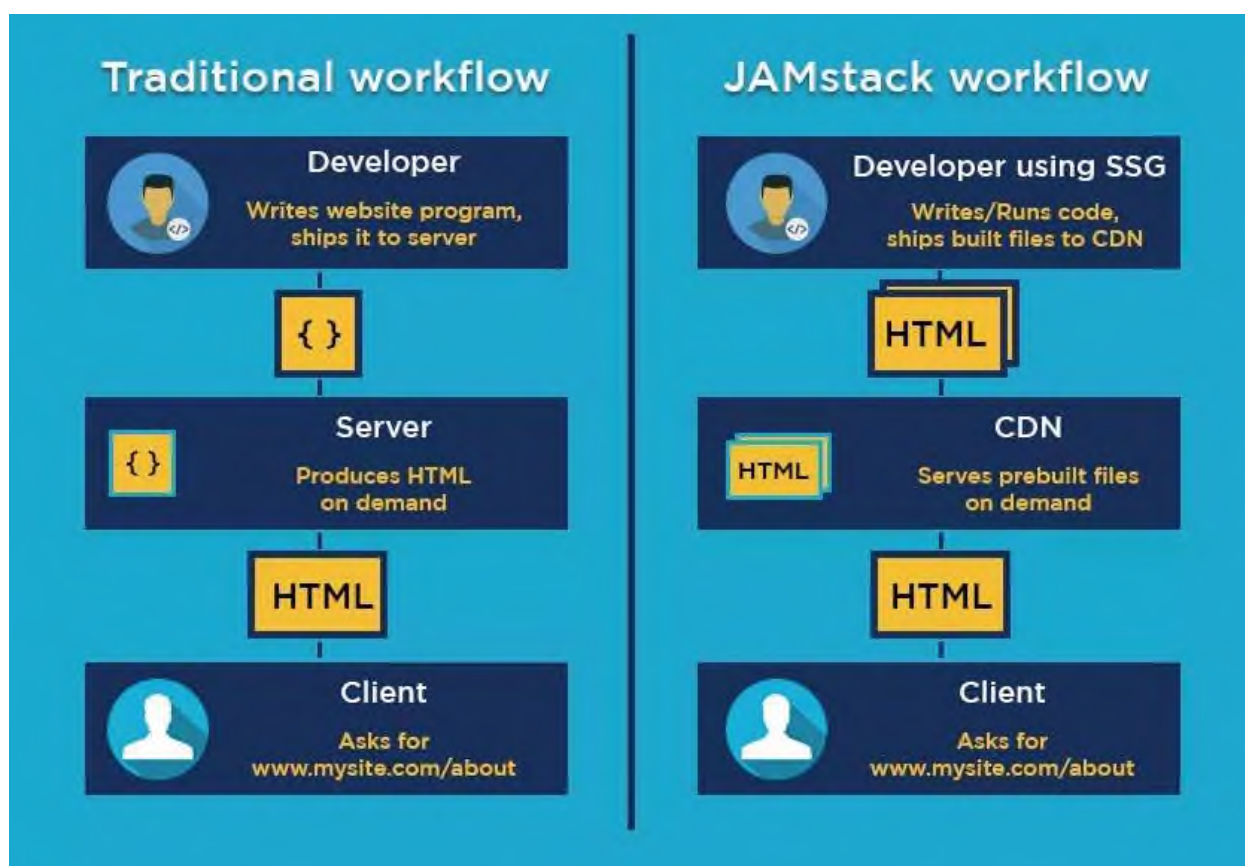
Jamstack 在国内落地时，总是会有同学认为这是新瓶装旧酒，或者是前端炒出来的新概念，但实际上忽略了 Jamstack 自身架构的特性与优势。

Jamstack 是不是新瓶装旧酒

Q: Jamstack 和之前的手动发布页面到 CDN 有什么区别，是不是新瓶装旧酒？

A: 这里我可以很明确的给一个答案：不是。

首先在我看来，Jamstack 虽然表现的都是页面和资源托管至 CDN，是实际上背后的工作流与过往是截然不同的。相较于以往手动开发和发布的模式，Jamstack 是聚合了现代前端框架、工作流、发布平台的，这一点非常的重要。



对于工程师而言，效率就是生产力，之前的页面不一定是基于现代工程的，每次生成静态页面也依赖于自己写的脚本，每次页面更新还需要手动发布，效率偏低。而使用 Jamstack，你可以利用你最熟悉的框架，通过框架去批量生成静态页面，并且聚合 Git 等服务，实现推送即发布的全自动工作流。效率会高很多。

举个相似的例子：使用 React/Vue/Angular 和使用 jQuery 写页面，最终都是操作 DOM，所以 React/Vue/Angular 和 jQuery 是一样的，新瓶装旧酒？

先进的工具带来先进的生产力，即使最终的表现是前端页面和十年前一样，部署至 CDN。但实际上生产效率与质量还有可生产的页面已经发生了很大的进步。

不论是什么前端页面，能打开的更快终究是更好的，因此如果只关注到部署的结果，而没有注意到生产流程与生产效率的变革，便会落入“Jamstack 是新瓶装旧酒”这个论点的坑中。顺带一提，近几年的前端工具基本上也是朝着“性能更好”、“更易用”等方向去前进的。

Jamstack 能适应所有的场景

Q：那是不是什么网站都上 Jamstack 就对了？

A：明确的回复，不对，Jamstack 的优势非常明显，所以劣势也很明显。

如果只关注到 Jamstack 的性能优势，希望将 Jamstack 用于所有场景，其实是不正确的。Jamstack 的站点为了获得性能、安全、可拓展性的优势，需要将页面托管至类似 CDN 的服务中，这个过程中，一个页面需要经过以下两步才会真正的发布到线上。

1. 预渲染
2. 需要提前渲染出最终的页面
3. 发布
4. 托管服务刷新缓存后展示新页面

针对预渲染，由于往往需要在构建时或者运行时实现功能，那么会存在一定的限制。构建时生成无法实现千人千面，运行时生成则需要考虑生成的数量与成本的考虑。假定将每一条微博都生成一个静态页面，诚然性能是好了，但所带来的成本也是不可估量了，且许多微博往往访问的人极少，那么运行时生成的性能可能还不如之前，

在 Jamstack 架构下，CDN 是最常见的托管服务，但 CDN 为了保证性能也存在着缓存的机制，这意味着页面的实时性无法保障。现有的 Jamstack 框架也会添加定时生成的功能，比如每 10S 就刷新一个页面并推送至 CDN，但不论怎么做，在实时性上还是不如实时服务的。

Jamstack 最适合一些内容更新不太频繁的网站（比如新闻、电商、文档）。它不适合 Feeds 流、聊天室、论坛、个性化推荐这样高度动态化的网站，以及邮箱、编辑器这样偏重型的 Web 应用。

Jamstack 会是企业级框架的核心特性而非唯一，混合渲染是未来方向

这个观点是我个人的想法。

在我看来，之所以 Jamstack 在国内难以落地，除去老生常谈的工作流、部署平台、底层依赖的限制外，其实还存在着适用范围单一的问题。

这里我抛出我的观点：在国内的市场下，Jamstack 将会成为企业级框架的核心特性，但并非唯一的特性，混合渲染才是未来的方向。

诚然，Jamstack 的优势非常明显，用过的同学都说省事都说好，但我在前文也提到了 Jamstack 的劣势，这决定了在企业内部错综复杂的场景中，Jamstack 不是那么万能的。

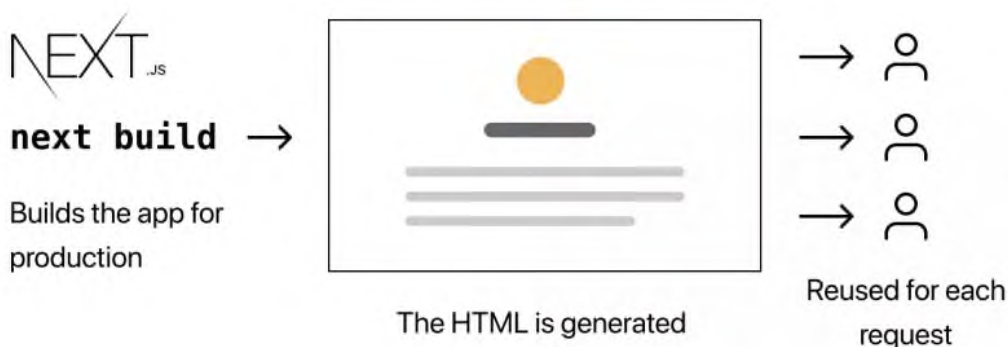
此外，Jamstack 作为一种现代 Web 站点的开发理念，其是易于被框架实现的。这也意味着，在企业级的场景中，往住会是框架去实现 Jamstack 特性，这个过程是新增而不是替换。因此 Jamstack 会是企业级框架的核心特性而非唯一特性。

至于后一句提到的混合渲染，Idea 实际上是源于 Next.js 10。在 Next.js 中，框架配合 Vercel 云服务平台，实现了纯静态页面托管、增量生成、服务端渲染等多种渲染策略的聚合。从这个角度来看，Next.js 相较其他的 Jamstack 框架是更有优势的。

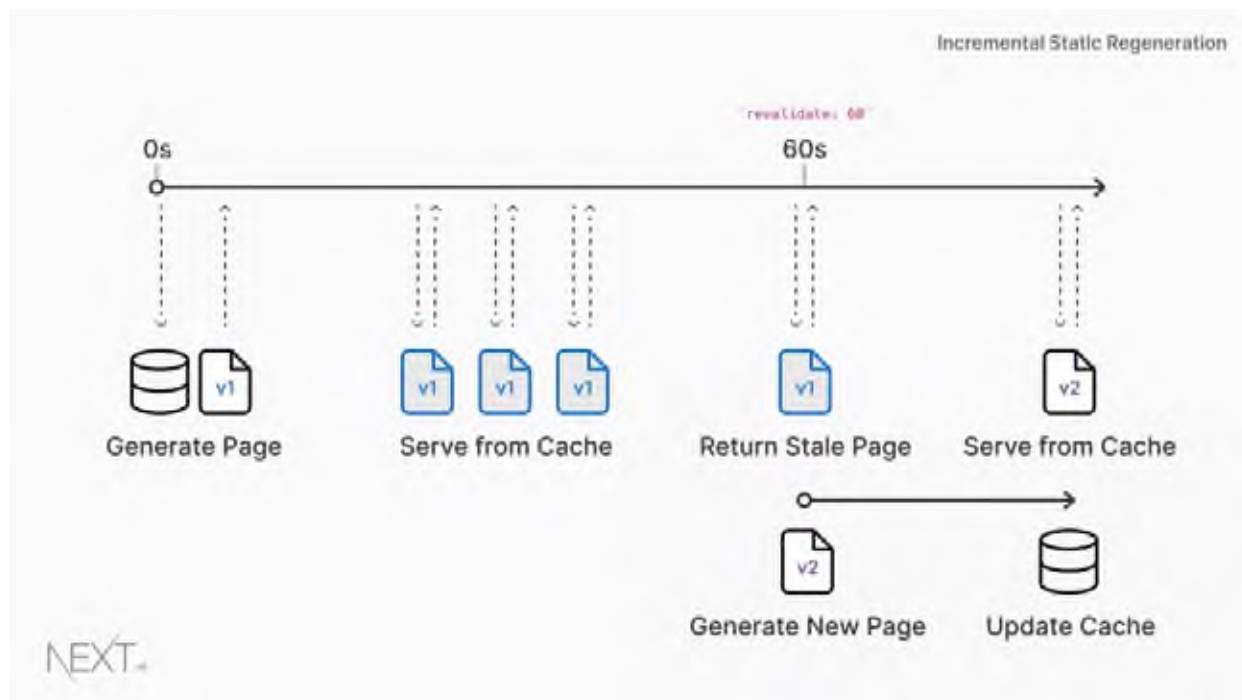
静态生成

Static Generation

The HTML is generated at **build-time** and is reused for each request.



静态生成



先进的框架 + 先进的工作流 + 更多渲染场景的适配，我认为这才会是企业级框架进步的方向。

总结

Jamstack 是一套优秀的现代 Web 站点开发技术栈，在现代前端工程的加持下解决了开发效率与性能的难题。但由于其劣势也非常明显，所以在企业级框架中，Jamstack 会是一种核心的特性，但不是唯一的特性，企业级是需要支持类似混合渲染的开发模式的。

BTW，本文没有提及到使用 JavaScript 与连接第三方服务这两个特性，理由是：构建现代站点，完全脱离 JavaScript 不现实，故略过；预渲染页面时，往往就包含使用第三方服务（当然我觉得这一点的商业价值实际更大），故也略过。

● 前端页面双向滚动方案

- 作者：豆苗
- 出品：淘系技术

脱离canvas后，页面如何实现上下左右双滑动？又如何在安卓系统和iOS系统上实现？

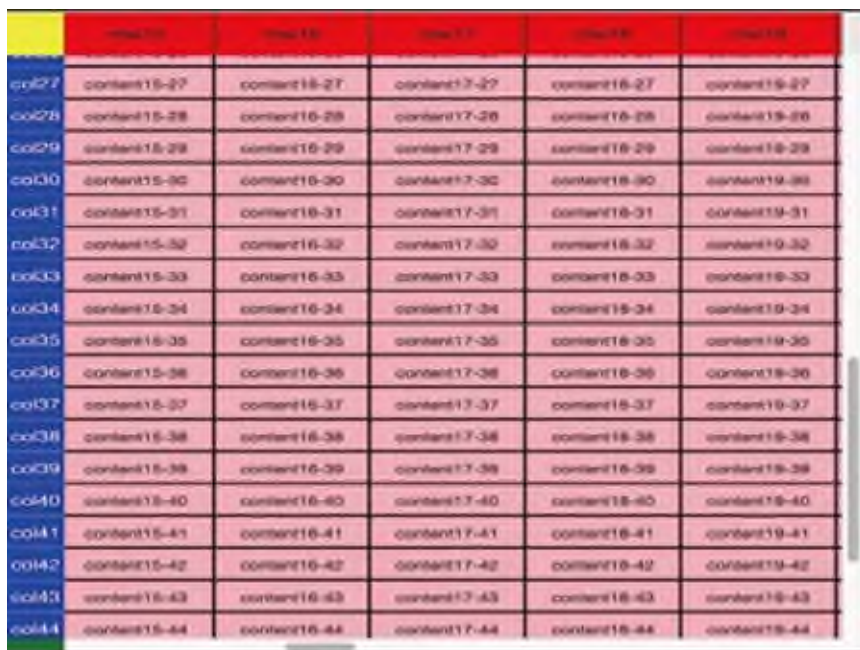
背景

在许多业务场景中会遇到让一个页面在web端可以实现双向滚动，且在滚动的时候可以上部吸顶，左侧吸在最左边这中类似excel表现的诉求。

但是excel移动端本身使用canvas来画的，所以能保持较好的体验和性能，那么脱离canvas，我们是否有别的方案可以去靠近实现呢？如何在移动端实现，并同时兼容安卓和ios系统？

实现目标

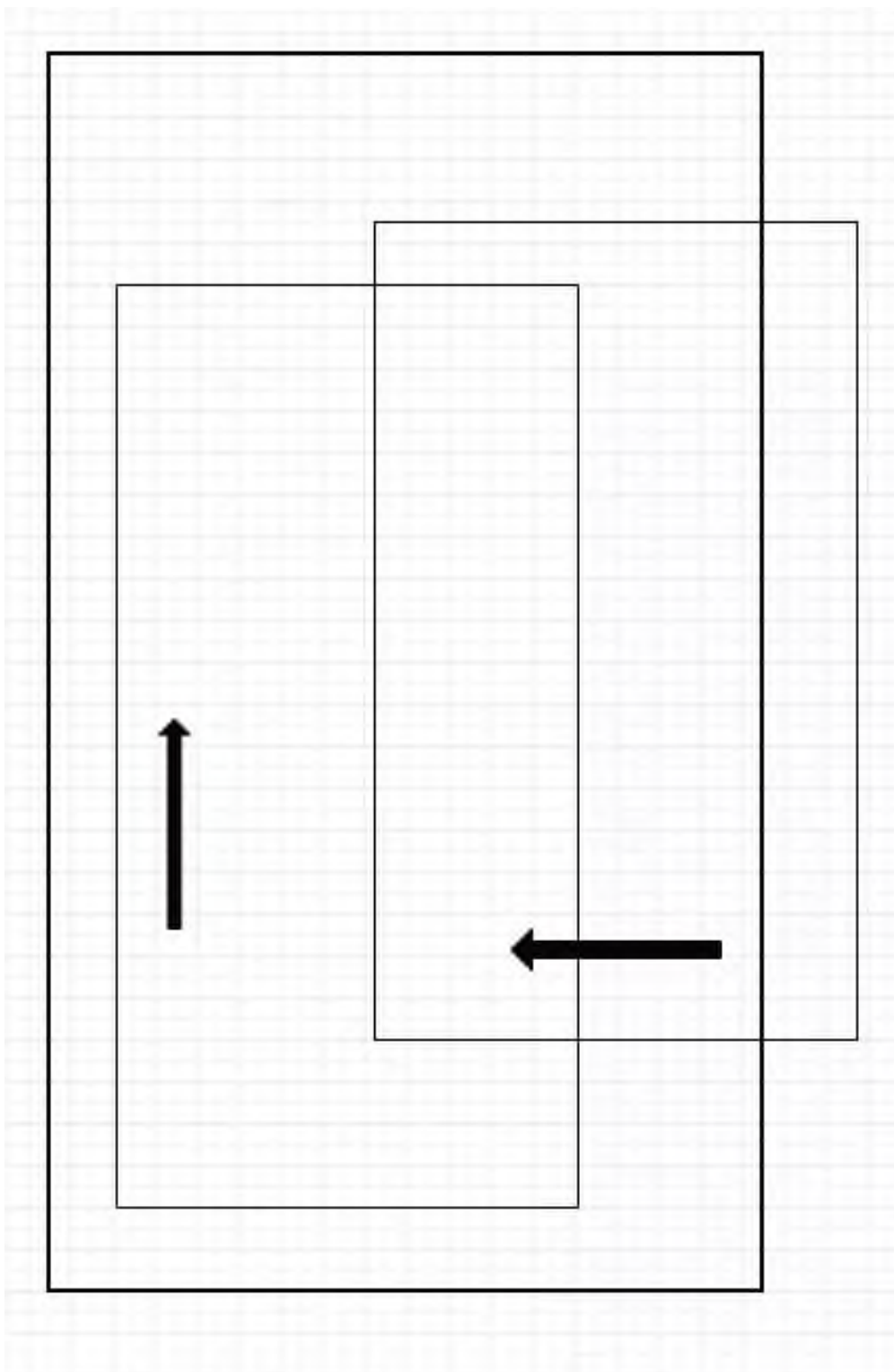
如下图所以，可以左右移动，也支持上下移动，在移动的对应该方向的顶部需要吸顶。在pc端实现较为简单，通过滚动就可以实现，在移动端要想实现这一套方案，并同时兼容不同系统，这里我踩了一些雷，下文将诉说实现这个的全部技术方案的变化和踩雷的一些点。



	col1	col2	col3	col4	col5	col6
row1	content15-27	content16-27	content17-27	content18-27	content19-27	content20-27
row2	content15-28	content16-28	content17-28	content18-28	content19-28	content20-28
row3	content15-29	content16-29	content17-29	content18-29	content19-29	content20-29
row4	content15-30	content16-30	content17-30	content18-30	content19-30	content20-30
row5	content15-31	content16-31	content17-31	content18-31	content19-31	content20-31
row6	content15-32	content16-32	content17-32	content18-32	content19-32	content20-32
row7	content15-33	content16-33	content17-33	content18-33	content19-33	content20-33
row8	content15-34	content16-34	content17-34	content18-34	content19-34	content20-34
row9	content15-35	content16-35	content17-35	content18-35	content19-35	content20-35
row10	content15-36	content16-36	content17-36	content18-36	content19-36	content20-36
row11	content15-37	content16-37	content17-37	content18-37	content19-37	content20-37
row12	content15-38	content16-38	content17-38	content18-38	content19-38	content20-38
row13	content15-39	content16-39	content17-39	content18-39	content19-39	content20-39
row14	content15-40	content16-40	content17-40	content18-40	content19-40	content20-40
row15	content15-41	content16-41	content17-41	content18-41	content19-41	content20-41
row16	content15-42	content16-42	content17-42	content18-42	content19-42	content20-42
row17	content15-43	content16-43	content17-43	content18-43	content19-43	content20-43
row18	content15-44	content16-44	content17-44	content18-44	content19-44	content20-44

技术方案

初期思想



如上图所示，不就是横向滚动和竖向默认滚动吗？

移动端屏幕宽高固定，本就可以左右滚动，那么只需要元素本身的大小大于屏幕的宽高不就可以了吗？

设置一个div，宽大于screen.width， 高大于screen.height即可，最终demo大概效果如下图所示：

	奥迪A3 2021款 Sportback 35 TFSI 进取致雅型	傲旋V-Class 2020款 V 260 L 傲旋大白鲨半隔断	爱驰U5 2020款 Lite
只看不同			
基本信息 ● 标配 ○ 选配 - 无			
官方售价	20.31万 询底价	109.8万 询底价	20.79万 询底价
厂商	一汽-大众奥迪	车驰汽车	爱驰汽车
级别	紧凑型车	中型MPV	中型SUV
发动机	1.4T 150马力 L4	2.0T 211马力 L4	-
变速箱	7挡双离合	7挡手自一体	电动车单速变速箱
长×宽×高 (mm)	4343×1815×1458	5420×1928×2060	4680×1865×1700
车身结构	5门5座两厢车	5门7座MPV	5门5座SUV
最高车速 (Km/h)	210	185	160
官方0—100Km/h加速时间(s)	9.0		10
官方油耗 (L)	-	-	-
整车保修期限	3年或10万公里	待查	4年或15万公里
车身 ● 标配 ○ 选配 - 无			

左右的确都相当的丝滑，能横着滚，也能竖着滚了，就是全方位也都能滚动，着实诧异了一会儿。

更新思维

如果那么当我朝着某个方向滚动的时候，禁掉一个方向的滚动，就可以解决了。

但是容器嵌套就出问题了，横向滚动的时候，我们领容器高度变成视口高度，overflow-y设置为hidden，横向依旧可以滚动，纵向设置为禁止滚动，在chrome里模拟了一下，的确不错，在真机上测试的时候，发现安卓端正常，ios端一旦手动设置overflow-y，发现纵向原先滚动到的位置会丢失，纵向内容直接回到最顶部，即纵向scrollTop变成了0。

设置纵向滚动，横向overflow-x为hidden时，横向原先滚动的距离也会被清除，回到最左侧。

那么通过手动记录之前的滚动位置，在设置某一方向禁止滚动时，把高度给他还原回去，就能解决了。

思路是对的，但是会出现强烈的闪屏现象，尤其在这种数据密集型的页面上（ios页面也闪，低端安卓机没做尝试，meta30存在一定延迟的闪屏）。

改变方案

双向滚动到此基本上算是失败了。

单向滚动肯定没问题，另一个方向我们可以使用模拟滚动的方式来自己写一套"滚动"效果。

首先我们判断横向滚动和纵向滚动那一方向使用原生滚动呢？

我选择了纵向，纵向数据体量大，容易有滚动的诉求，横向存在数据量不足可能不需要滚动。

效果如下图所示：

参数配置			
 仅看差异	☆ 钉在左 × 2021 2.0TSI 双离合 380TSI 豪华版	☆ 钉在左 × 2021 2.0TSI 双离合 380TSI 尊贵版	☆ 钉在左 × 2021 2.0TSI 双离合 380TSI 旗舰版
	34万	40.7万	44.6万
多媒体配置 · ● 标配 ○ 选配 - 无			
车载 CD/DVD	-	-	-
220V/230V 电源	●	●	●
行李厢12V 电源接口	-	-	-
扬声器品牌名称	-	● Dynaudio丹拿	● Dynaudio
灯光配置 · ● 标配 ○ 选配 - 无			
近光灯	● LED	● LED	● LED
远光灯	● LED	● LED	● LED
灯光特色功能	-	-	-
日间行车灯	●	●	●
自适应远近光	-	-	●

左侧第一列和顶部第一行吸顶这里使用了css的属性 position:sticky。

也有人可以考虑使用一些特殊布局，例如左侧做绝对定位，右侧内容自行滚动等。

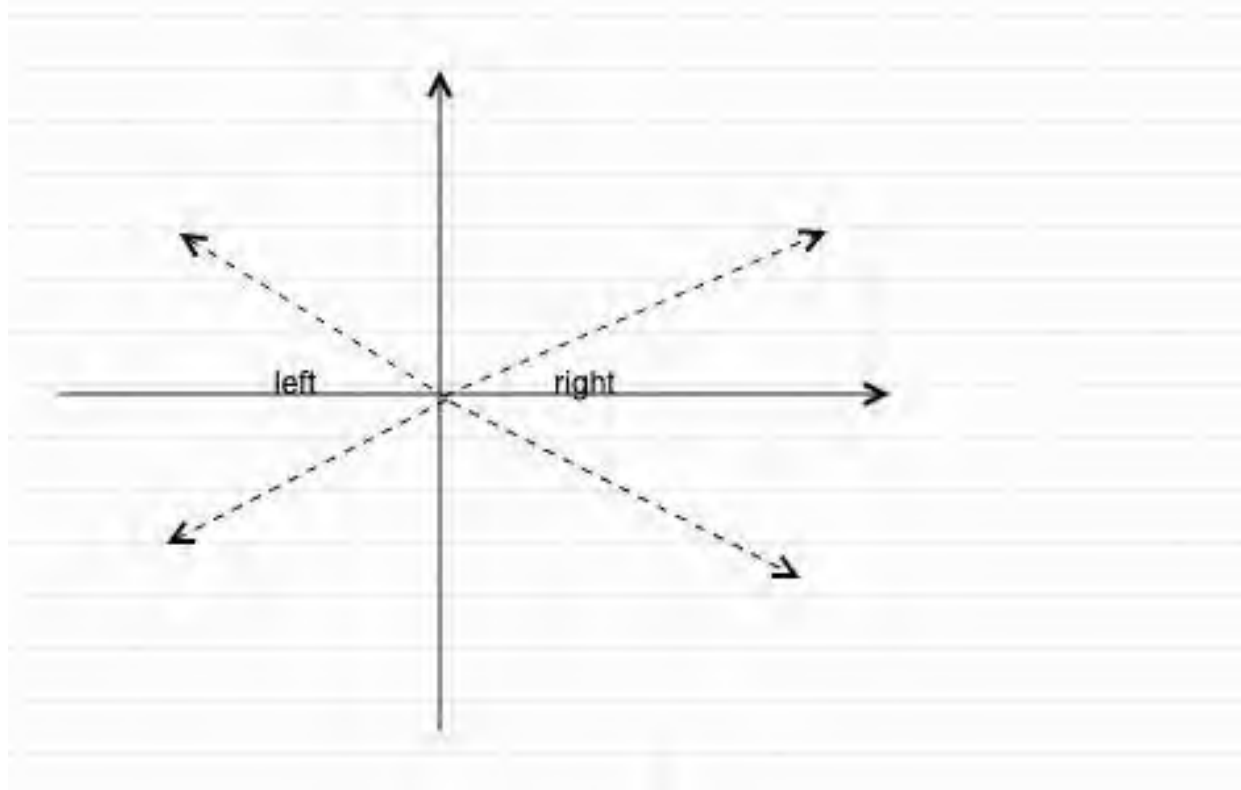
模拟滚动的方式，我们采用touch事件来仿造，通过touchStart、touchMove、touchEnd 这3个事件来模拟scroll事件。

首先在touchStart中获取手指头接触到屏幕的点的 (x,y) 坐标，然后在touchMove事件中获取屏幕出点的第二个触点的 (x, y) 坐标，来获取触摸方向。

```
1  //获得角度
2  function getAngle(angx, angy) {
3    return Math.atan2(angy, angx) * 180 / Math.PI;
4  };
5
6  function getDirection(startx, starty, endx, endy) {
7    const angx = endx - startx;
8    const angy = endy - starty;
9    let result = 0;
10
11   //如果滑动距离太短
12   if (Math.abs(angx) < 2 && Math.abs(angy) < 2) {
13     return result;
14   }
15   const angle = getAngle(angx, angy);
16   if (angle >= -150 && angle <= -30) {
17     return 'top';
18   } else if (angle > 30 && angle < 150) {
19     return 'down';
20   } else if ((angle >= 150 && angle <= 180) || (angle >= -180 && angle < -150)) {
21     return 'left';
22   } else if (angle >= -30 && angle <= 30) {
23     return 'right'
24   }
25 }
```

由于大家在使用手机时，横向滑动的行为轨迹并非是一个水平180度的横线，所以在这里设定了一定值，例如多少角度到多少角度认为是横向滚动。

方向定好，多少角度这个可以根据自己业务上的判断做修改。



连续触摸滑动的坐标轴我们已经能拿到了，通过css3的transform来使元素进行移动，通过不断修改移动的值达到元素滚动的效果。

这里有一点需要注意，并非我们触摸划过多少像素，元素就移动多少像素，这里还涉及一个速度和距离的概念。

还有我们手指头松开以后，元素应该还会滚动一部分，然后速度慢慢降下来，直到0为止。

先分析一直处于触点滑动这个阶段，我们并非划过多少像素，元素就跟着滚动多少像素，这里看情况来放大或者缩小这里的滚动值，体现给用户的感受就是滑的快还是划得慢。

我们由于横向数据量并不大，所以这里我们所有滚动距离都做了一定缩小处理。

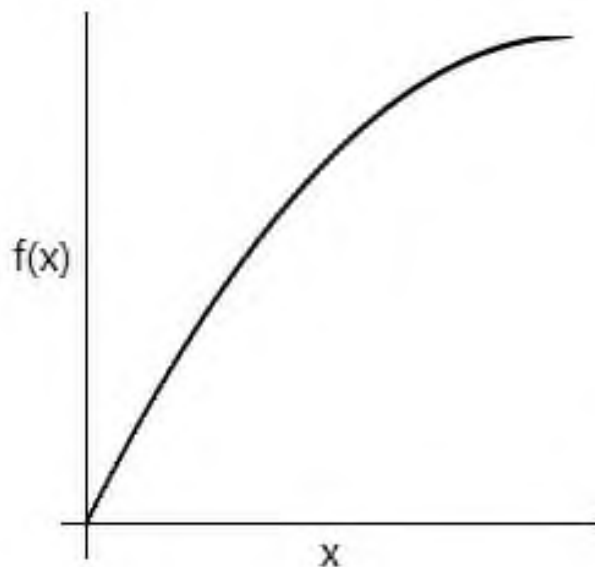
```
1 // 代码只留个框架
2 function rowScrollAction(scrollDirection, endx, isEnding = false) {
3     scrollGap = (endx - startx) * 0.85; // 每个滚动距离都乘上0.85
4     let scrollLen = preX + scrollGap; // preX为之触摸之前已经滚动到的位置
5     if (isEnding) {
6         scrollLen = endx
7     }
8     if (scrollDirection === 'right') {
9         ....
10    }
11    if (scrollDirection === 'left') {
12        ....
13    }
14 }
```

这里还需要注意的点是，如果已经滚动到横向边界了，那么就不允许在滚动了，这里需要特殊处理一下。

那么触摸结束以后，元素还应该按照惯性继续向对应方向进行滚动，速度慢慢下降，直到结束为止。

速度不能直线下降，不能是一次函数，因为： $y=ax+b$

a 代表加速度，加速的值一直不变，那么就会匀速降下去，体验上并不好，这里选择了二次函数的概念。



如图所示，慢慢靠近目标值，那么就要开口朝下的抛物线，且是对称轴左侧的这一部分函数。根据函数定义：

$$y = ax^2 + bx + c$$

通过a、b参数来产出一个对应计算函数：

```
1 // 将"当前时间"映射[0, 1]间(currentTime /= duration)
2 // 根据缓入公式计算出"当前时间"应该移动的百分比(currentTime * currentTime)
3 // 根据"总移动长度"和"移动百分比"算出应移动的具体值(changeValue * ...)
4 // 加上初始位置(+ startValue)
5 function easeOut(currentTime, startValue, changeValue, duration) {
6   currentTime /= duration;
7   return -changeValue * currentTime * (currentTime - 2) + startValue;
8 }
```

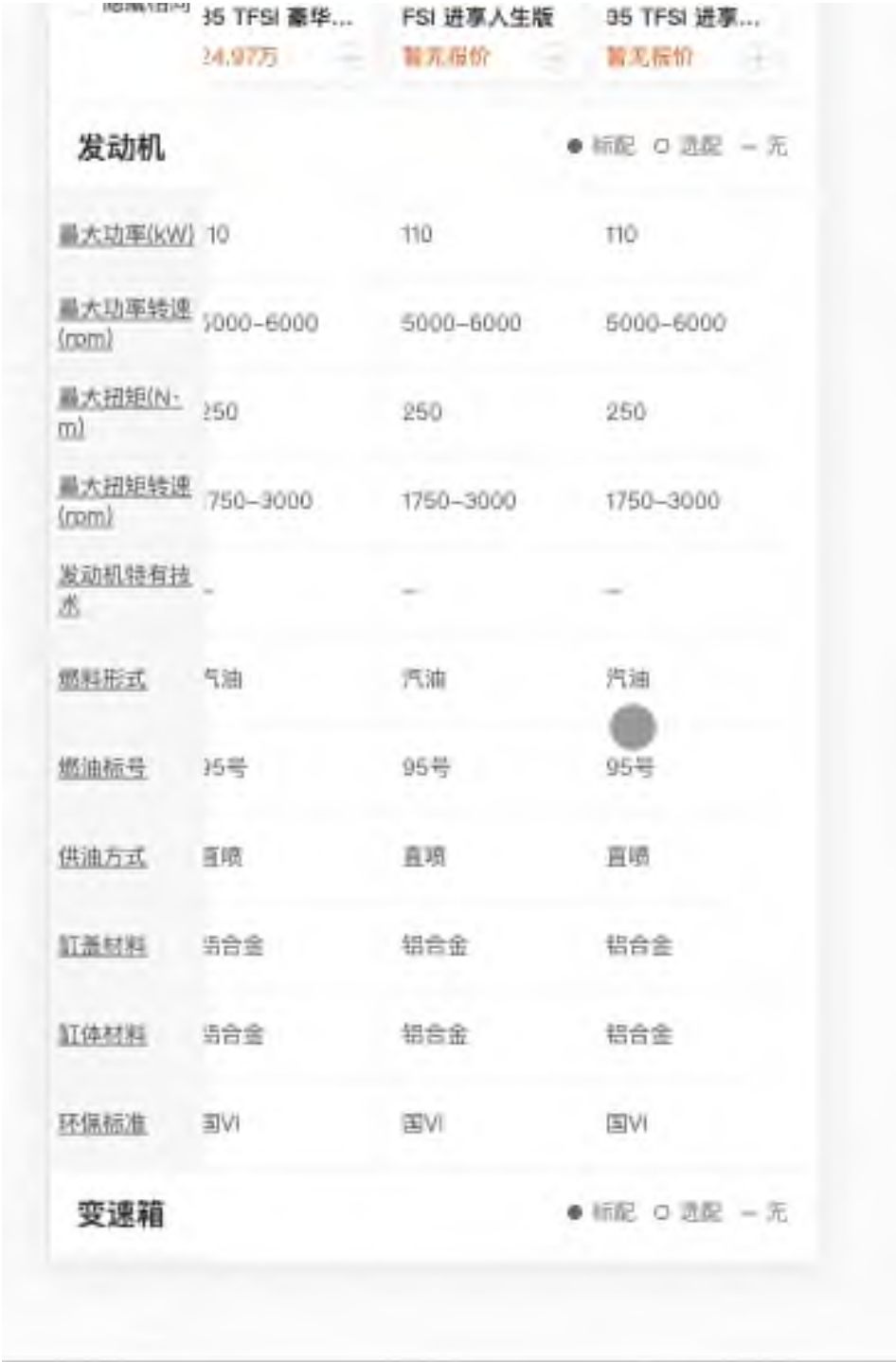
本业务场景中的实现代码如下：

```
1 /**
2  * target: 滚动最终像素值
3  */
4 const scrollAnimation = (target) => {
5   function easeOut(t, b, c, d) {
6     return -c * (t /= d) * (t - 2) + b; // 用时间t做x变量
7   }
8   var toTarget = target;
9   var endTimer = 500;
10  var startTimer = 0;
11
12  var step = function () {
13    let value = easeOut(startTimer, preX, toTarget, endTimer); // 返回一个距离值
14    // rowScrollAction(tempDirection, value, true); 执行对应元素滚动多少像素的方法
15    startTimer += 25;
16    if (startTimer <= endTimer) {
17      // 继续运动
18      requestAnimationFrame(step);
19    } else {
20      // 动画结束
21      touchTimes = 0;
22      // 动画结束后的回调
23    }
24  };
25  step();
26 }
```

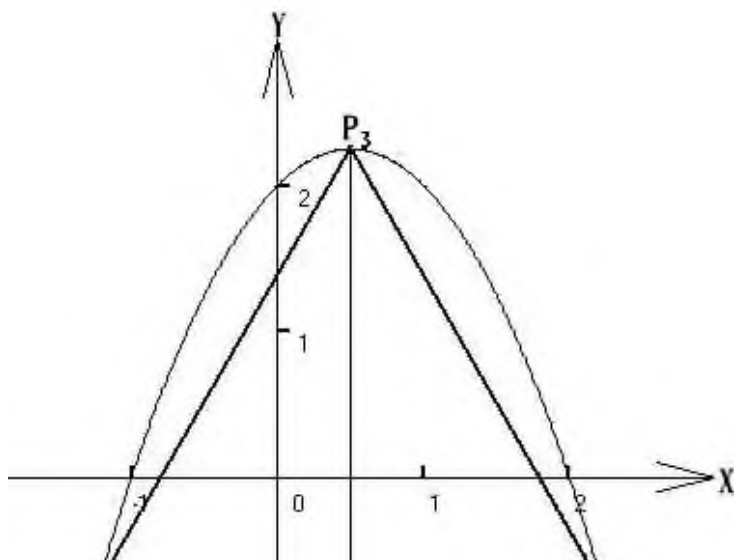
当滚动到边界时，记得清空循环计算哦

最后优化

当横向滚动到边界时，模拟ios的弹簧效果（ 安卓和ios端保持一致 ）



实现方式就是对3.3中的滚动到边界时做对应的处理。我们达到边界以后，在滚动距离上多加50~100个像素值，然后利用抛物线的原理：



利用两边的值，先慢慢靠近最高点，然后再回到对应y值上。

结论

接手一个项目，要严谨一点，如果是之前没接触过这种技术方案开发的，应该拿到项目时先写个小demo，然后再去评估开发时间。

不然脑海中出现的技术方案可以让你当场拍板，但是在实践中会发现出现预估错误，可是开发排期影响的就是整个小team的时间，你的delay很可能导致大家的排期都被打乱，不得不从拼命加班抢回开发时间。

还要学会给自己留充足的buffer能够去试错，能够应对突发情况。

• 图片上传系统在淘系中的实践

- 作者：安笺、央明
- 出品：淘系技术

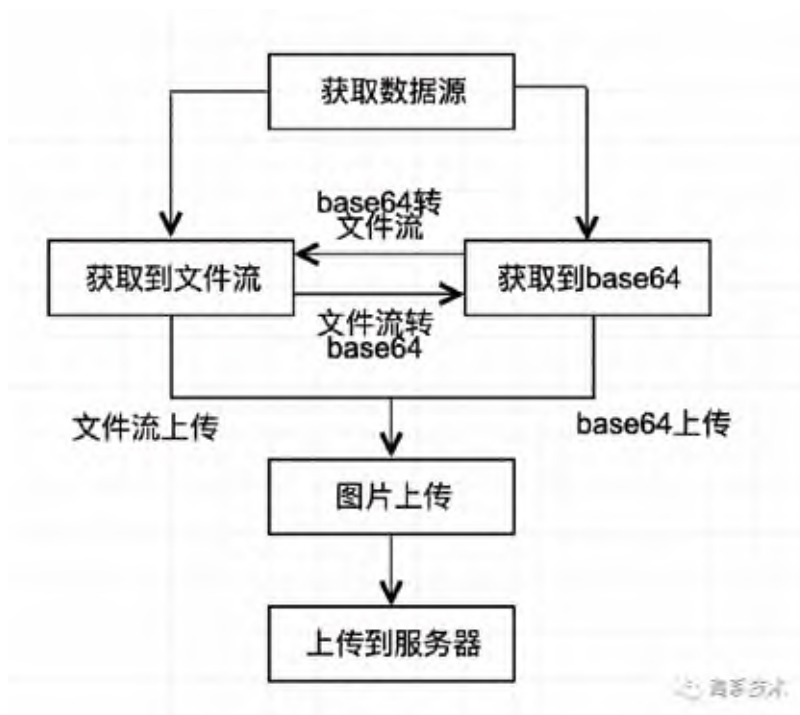
传统图片信息的上传渠道一般在pc端的后台或者客户端上直接上传，但是如何在移动端的webview上快速上传图片，并保证图片安全下载这一直是一个挑战。

名词解释

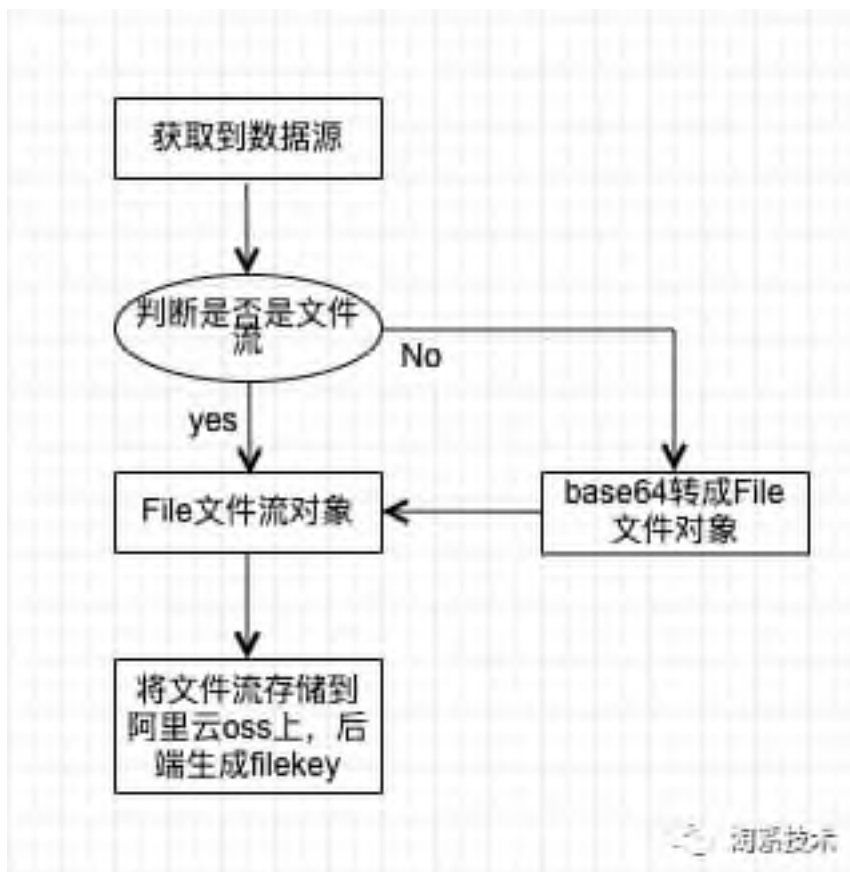
- 阿里云oss：阿里云对象存储OSS（Object Storage Service）是阿里云提供的海量、安全、低成本、高持久的云存储服务，在本文中用于存储图片。
- 文件流：二进制文件流。
- base64：图片的base64二进制。

整体技术方案

前端整体图片上传流程图

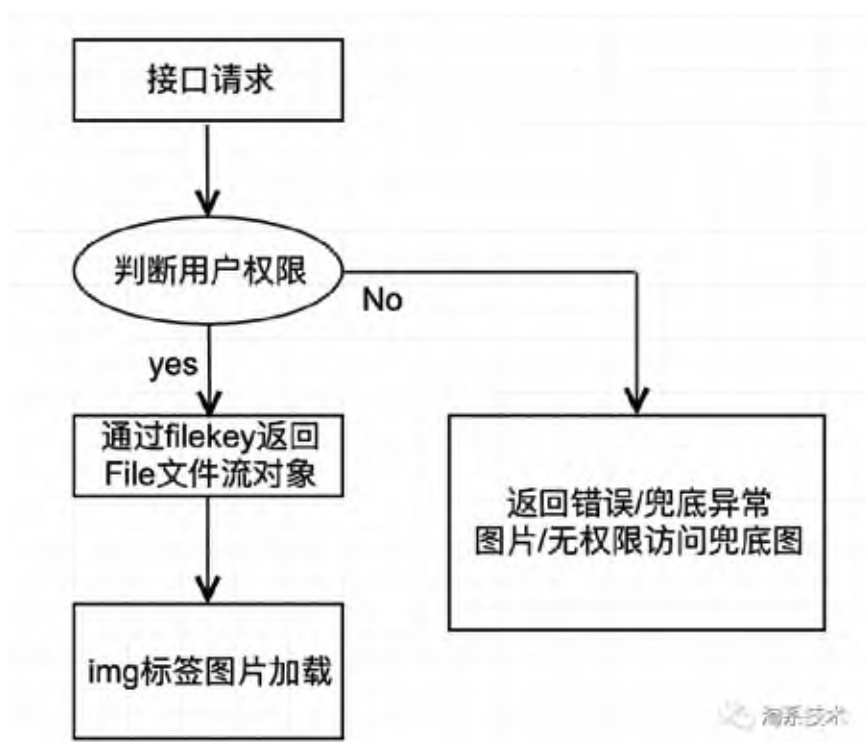


后端图片存储及图片安全返回



服务端拿到文件流存储到oss上后，oss会提供一个临时的文件访问url，服务端拿到链接就可以做非常多的操作了，比如图片识别，图片解析，图片裁剪，图片选装等等。

阿里云的oss在图片安全问题上也提供了一些能力，比如过期时间，你可以设定一张图片访问的有效时间，比如5分钟后图片访问就失效了，这种情况能解决大部分图片安全问题，但对于一些特殊需求，比如懒加载的图片，feeds流里的图片这种在不确定什么时间才加载的图片，就会出现问题。所以这里提出图片和用户绑定，对图片进行用户权限的校验：



前端方案

初期方案

经过调研和技术选型，我们初步选择使用文件流的传输方式，通过http接口传输，直接把文件流打到后端服务，然后后端去把文件流存储到阿里云的oss上，这套方案在pc端实现良好，但是到了移动端，为了更好的样式和体验，通过唤起相册或者拍照的方式，只能获取到图片的base64，而无法直接获取文件流（标准input标签也能打开相册或拍照，但是是系统级别的行为，样式和交互方式不可控）。所以只能通过将base64转成文件对象去上传，这样从理论上可以和pc端保持一致，并通过获取base64的image/**来获得数据类型和自定义后缀，自定义文件名称后传入到filename参数：

```
1  dataURLtoFile('data:image/png;base64,iVBORw0KGgoAAAANSUh.....', 'test.png');
2
3  function dataURLtoFile(base64String, filename) {
4
5      var arr = base64String.split(',');
6      mime = arr[0].match(/:(.*?);/)[1];
7      bstr = atob(arr[1]);
8      n = bstr.length;
9      u8arr = new Uint8Array(n);
```

```
10
11 while(n--){
12     u8arr[n] = bstr.charCodeAt(n);
13 }
14 let file = new File([u8arr], filename, {type:mime});
15 return file;
16 }
```

到这里一切都很完美，pc端可以通过input标签获取file对象直接上传到对应接口，移动端通过base64转换成File对象在上传到对应接口，使用一个接口就能同时打通pc端和移动端，但是在测试的时候抓包出现问题了。

由于我所使用的端内webview测试环境发现其对FormData兼容性不强，文件名经常打不上去，文件名全部变成了file，并且文件大小变成0Kb，从而导致后端无法直接拿文件流。

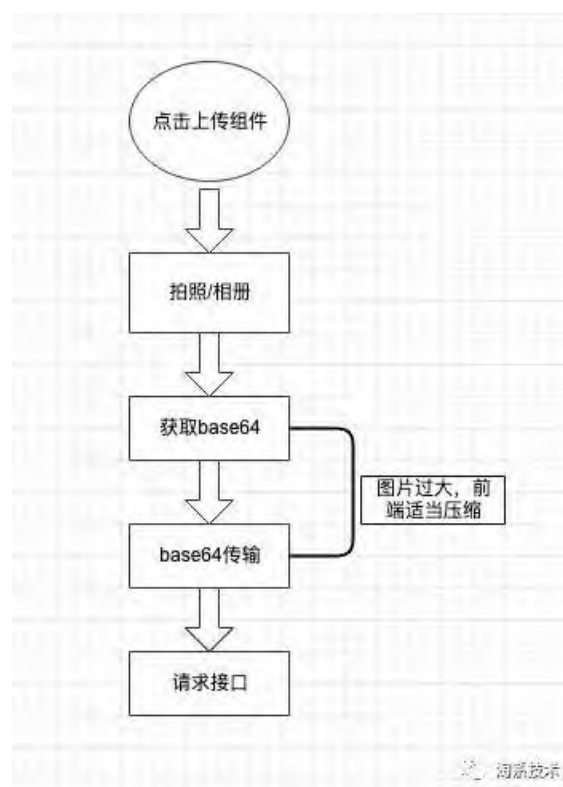
方案迭代

规避端内formData兼容问题

在接下来的文件上传相关的资料学习和写demo尝试中发现，base64本身就是个字符串，可以将其上传至服务器端，让服务器端来处理文件并上传至阿里云的oss上。

File对象后移

既然前端在移动端webview里无法正常传输js生成的File对象，那么把File对象的生成环节移到服务端去生成，不就可以了，毕竟双端环境(Android、Ios)对于base64字符串传输还是友好的，但是由于base64比文件流是要大的，所以这里会在传输过程上造成一定的时间和资源的消耗。具体方案如下：



通过客户端的能力获取相册图片的base64，然后对base64进行大小压缩的处理，demo如下：

```
1  /**
2   * 对图片进行压缩处理
3   * @param base64String 源图片base64
4   * @param maxSize 可以设置成1M = 1024 * 1024 = 1048576，这里根据自己业务调整
5   * 图片base64和文件流之间大小可以参考：https://juejin.cn/post/6913814643618086925
6   */
7  function processImage(base64String, maxSize, scale = 1) {
8    const canvas = document.createElement('canvas');
9    const context = canvas.getContext('2d');
10
11    const source = new Image();
12    source.onload = () => {
13      let sw = source.width * scale;
14      let sh = source.height * scale;
15
16      canvas.width = sw;
17      canvas.height = sh;
18
19      context.drawImage(source, 0, 0, sw, sh);
20
21      const imageBase = canvas.toDataURL('image/jpeg', 0.85);
22      if (imageBase.length > maxFileSize) {
23        processImage(imageBase, maxSize, 0.7);
24        return;
25      }
26      uploader(imageBase);
27    };
28    source.onerror = () => {
29      return base64String
30    };
31    source.src = base64String;
32  }
```

拿到处理后的base64，接下来就需要把对应的base64传递到后端，为了和浏览器端使用同一个上传接口，前端这边采用multipart/form-data上传类型的上传方式：

```
1  function uploader(base64string) {
2    const xhr = new XMLHttpRequest();
3    xhr.withCredentials = true; // 开启同源策略，避免因多级域名不同而导致cookie缺失
4    const fd = new FormData();
```

```
5     fd.append('imgBase64Str', base64string);
6     fd.append('xxx', 'xxx');
7     .
8     .
9     .
10    xhr.open('POST', 'XXXXX', true);
11    xhr.onreadystatechange = function() {
12        if (xhr.readyState == 4 && xhr.status == 200) {
13            console.log('I get it')
14        }
15    };
16    xhr.send(fd);
17 }
```

最后通过阿里云的oss查看、下载接口等方案验证后都正常。

如何同时打通pc端也适用这套上传体系呢？

后端可以新增个字段来判断当前获取到的是base64还是文件流。由于阿里云oss都是直接上传文件流，pc端可以直接获取文件流，那么可以减少后端把base64转文件流的成本，拿到就直接上传到oss上！

```
1  const reader = new FileReader();
2  reader.readAsDataURL(file); // file => input里的File对象
3  reader.onload = () => console.log(reader.result);
   reader.onerror = error => console.log(error);
```

后端实现方案（JAVA）

提供controller作为http入口

目前的测试应用框架是基于spring boot(PandoraBoot)的，所以可以直接写@Controller注解的类接收http请求。

如果跨域，针对跨域的支持

在Controller类的方法上增加：

```
@CrossOrigin(origins = {"https://xxxx.com", "https://xxx.com"})
```

就可以提供对跨域访问的支持，依次来提供不同应用的域名访问图片的诉求。origins里可以写多个域名。

上传文件使用MultipartFile的方式

方法的定义为：

```
1  @CrossOrigin(origins = {"https://test.com", "https://test1.com"})
2  @RequestMapping(value = "/pathname/result", method =
3  RequestMethod.POST)
4  @ResponseBody
5  public TestResult<FileUploadResponse> uploadOssFile(@RequestParam MultipartFile file,
6  @RequestParam(value = "test") String Test,
7  HttpServletRequest request, HttpServletResponse response){
8  }
```

其中TestResult和FileUploadResponse都是自定义的数据结构。

如果是这种方式，则获取file.getInputStream()后直接传到阿里云的oss上即可，阿里云的oss上传接口：

```
ossClient.putObject(bucketName, fileKey, inputStream);
```

不使用MultipartFile的方式上传图片

一般来说MultipartFile就能满足大多数需求了,对于PC浏览器来说MultipartFile就是标准的上传文件的方式。

但是我们的情况有些特殊，端上获取手机本地的图片文件后无法用MultipartFile传给后端，后端http接口接收到file后内容一直是空，这个问题一直没有得到解决，所以针对图片上传就使用了base64编码的方式，简单来说就是前端先把图片转成base64编码后的一个很长的字符串，然后把这个字符串作为参数传给后端。

方法的定义变成了：

```
1  @CrossOrigin(origins = {"https://test.com", "https://test1.com"})
2  @RequestMapping(value = "/pathname/result", method = RequestMethod.POST)
3  @ResponseBody
4  public AliAutoResult<FileUploadResponse> uploadOssFile(@RequestParam String imgBase64Str,
5  @RequestParam(value = "test") String Test,
6  HttpServletRequest request, HttpServletResponse response){
7  }
```

其中imgBase64Str就是图片编码后的字符串，因为长度太长了(稍微大点的图片编码后字符串长度就超过1M了)，springboot默认POST传参的大小为1M，为了突破这个限制，需要做如下配置：

```
1 # POST上传文件大小限制
2 server.tomcat.max-http-post-size = -1
3 spring.http.multipart.maxFileSize = 50Mb
4 spring.http.multipart.maxRequestSize = 100Mb
```

对imgBase64Str的处理

收到base64编码的字符串后，需要将这个字符串解码后，再作为文件流的形式存储到oss上。

先将imgBase64Str转储为本地临时文件：

```
1 //获取临时文件
2 File file = makeTempFile(imgBase64Str);
3
4 /**
5  * 将上传的文件先在本地生成临时文件
6  *
7  * @return
8  */
9 private File makeTempFile(String imgBase64Str) {
10     //解析文件类型
11     String base64Header = imgBase64Str.substring(0, imgBase64Str.indexOf(";base64,") + 8);
12     String fileType = base64Header.substring(base64Header.indexOf("/") + 1,
13 base64Header.indexOf(";"));
14     log.warn("base64Header={};fileType={}", base64Header, fileType);
15     //去掉头部
16     String imgBase64 = imgBase64Str.replace(base64Header, "");
17     //生成临时文件
18     File file = new File(FILE_NAME + fileType);
19
20     try {
21         // Base64解码
22         BASE64Decoder decoder = new BASE64Decoder();
23         byte[] bytes = decoder.decodeBuffer(imgBase64);
24         for (int i = 0; i < bytes.length; ++i) {
25             if (bytes[i] < 0) {
26                 // 调整异常数据
27                 bytes[i] += 256;
```



```
28     }
29     }
30     // 生成图片
31     OutputStream out = new FileOutputStream(file);
32     out.write(bytes);
33     out.flush();
34     out.close();
35 } catch (Exception ex) {
36     log.error("makeTempFile|Exception:", ex);
37     return null;
38 }
39
40     return file;
}
```

然后使用FileInputStream将file传到oss上:

```
1  FileInputStream fileInputStream = new FileInputStream(file);
2  ossClient.putObject(bucketName, fileKey, fileInputStream);
```

至此，上传工作已经完成。

临时文件使用完成后记得删除哦。

也有方案说直接操作stream即可，但本人在这里没做尝试，有兴趣的读者可以去尝试一下。

图片文件下载

我们要如何将图片的安全性做到一定程度，避免被他人恶意传播与利用，首先我们需要把图片分私有和共有，私有即只有特定的用户，比如上传者，后台审核人员等拥有对应业务标记身份的人可以访问，共有即不设置访问权限，所有用户都可以访问到这张图片。这样通过一些手段让非对应权限的人无法访问到这些信息流，从而达到对图片的泄露做到保护。

共有图片处理方案

如果是公有的业务背景下直接返回图片存储在服务器或cdn上的链接即可。

私有图片技术方案

如果是私有的图片，那么后端会提供一个有逻辑的http接口。例如：

`https://localhost/path?xxx=xxx&fileKey=${fileKey}`

服务器接口通过解析接口的参数fileKey，然后通过访问的用户是否具备对这个filekey所对应图片有访问权限来决定是否返回对应图片的文件流。

```
1  @CrossOrigin(origins = {"https://test.com", "https://test1.com"})
2  @RequestMapping(value = "/pathname/result", method = RequestMethod.GET)
3  @ResponseBody
4  public void downloadOssFile(@RequestParam(value = "fileKey") String fileKey,
5                             @RequestParam(value = "process", required = false,
6                             defaultValue = "") String process,
7                             HttpServletRequest request, HttpServletResponse response)
```

根据fileKey就可从oss上下载到文件，示例代码：

```
1  //获取文件访问url
2      GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest
3      (bucketName, filekey, HttpMethod.GET);
4      request.setExpiration(new Date(System.currentTimeMillis() + 1000 * expireTime));
5
6      //process是对图片的额外处理参数
7      if (StringUtils.isNotBlank(process)) {
8          request.setProcess(process);
9      }
10
11      URL url = ossClient.generatePresignedUrl(request);
```

其中process参数形如：

`image/action1,param_value/action2,param_value`

详细的oss文档=https://help.aliyun.com/document_detail/44686.html?spm=a2c4g.11186623.6.733.537b6c6bs6NzzQ

获取到oss返回的url后，如果不涉及文件的私密性，可以直接将这个url返回给前端展示。

但如果不能透出这个url(私密文件不宜返回url)，则可以读取这个url的http流，再返回HttpServletResponse的OutputStream流中。

思考

图片上传也好，视频流上传也好，思路都是一致，本文只表达了一次性将数据全打上去思路，大文件的话就不推荐了（大文件采用分片上传思路），但原理相同。图片上传方案前期webview受到客户端的各种限制，最后采用了base64这种无障碍信息上传方案。但是base64比图片本身文件流是要大出25%左右的，所以无形也增大了http带宽的开销，最好能文件流上传的话就一定优先采取文件流上传（除非受到宿主环境的困扰）。

图片目前互联网上流传的渠道基本就是图片链接，在浏览器中就可以打开，base64也是一样，所以在文件流的外面套一层http接口的壳子，就可以做到权限的校验。

当然，图片上传还可以在通过阿里云的平台做非常多的事情，比如ocr解析，图片的选择、裁剪、压缩等等，具体实现都可以去看看阿里云的文档。

• 无代码编程会是以后的趋势吗？

- 作者：凌乙
- 出品：淘系技术

近年来，关于低代码（LowCode）和无代码（NoCode）的讨论在前端社区内越来越火，简单的说低代码就是通过编写少量代码的方式完成应用的开发及上线，而无代码则更进一步，不需要编写代码通过配置的方式即可完成整个应用的开发。目前阿里集团内部的低代码平台已经有很多，比如iceluna，宜搭，乐高，云凤蝶等等，而通用的无代码搭建平台还处在探索阶段。

低代码和无代码

首先不管是低代码还是无代码，都是针对特定场景或者细分领域的，比如运营的活动页，中后台的表单，表格页面等，因为只有在这些场景下，前端交互相对收敛，能够沉淀出足够多的组件物料，从而通过可视化的方式拖拽组件就能够直接搭建出页面。

目前我所在的团队正在研究面向营销域的中后台前端解决方案。通常来说，中后台解决方案的核心目标是提效，提效包括两个方面，一方面是对研发人员的提效，另一方面是对用户的提效，提效的核心抓手在于生产关系的变更，由前端开发向后端，视觉，产品各方面参与发展，从而降低前端研发的门槛，提高生产效率。提效解决的不是20%的个性化增量需求，而是解决让“非前端”参与进来，解决80%的通用需求。中后台的提效路径大部分走的都是ProCode->LowCode->NoCode方式。

表面上看，从ProCode->LowCode->NoCode看起来好像只有很小的差别，好像只有代码量多少的问题，但整个过程已经从量变发生了质变。ProCode和LowCode主要面向的还是一些需要有前端编程能力的人，而NoCode则代表“非前端”也能够参与的前端的页面搭建中来，这里面不是说完全不需要代码，因为今天哪些算“代码”其实比较难界定，比如用户编写一个配置文件，这个文件是json格式的，那到底能不能算“代码”？所以，NoCode的意思不是说没有代码，而是说在于用户学习门槛和学习成本的降低，普通用户不需要经过艰难的学习就可以做到以前程序要编码才能实现的事情。

iceluna低代码平台的痛点问题

iceluna作为集团内优秀的低代码搭建平台，主要解决中后台页面快速搭建的场景，经过几年的探索，基本能够实现页面UI的可视化搭建，但是针对业务逻辑还是需要手动编码来实现。这对非前端开发人员的上手门槛还是比较高的。下面这张图是最近针对iceluna的用户（前端，后端和测试）做的一个调研分析，可以看到逻辑代码和数据绑定的学习成本也是用户在问卷中提的最多的。

组件使用：组件太多找不到，文档不清楚	20	31.3%
逻辑代码：手写代码成本高，分散	13	20.3%
数据绑定：学习成本高、使用困难	12	18.8%
发布速度：应用打包发布耗时很长	12	18.8%
画布能力：组件拖拽不流畅、画布利用率低	7	10.9%
总计	64	100.0%

因此在这个财年，我们尝试去用可视化逻辑编排的方式解决逻辑相关的问题，解决低代码中最后一点需要编码的部分，实现无代码化的研发模式，从而进一步降低用户的学习和使用门槛。

可视化逻辑编排

首先我们来通过一个逻辑编排的示例来看一下如果一段代码通过编排的方式呈现出来之后会带来怎么样的体感：



如上图所示，原本晦涩难懂的代码逻辑通过流程图的形式表达出来以后，产品的逻辑变得非常直观。可读性和可维护性也变得非常高。再也不用担心在接手其他人的项目时，注释不规范，文档不全的问题，逻辑编排生成的逻辑图谱就是天然的产品文档。

逻辑节点抽象

可以看出，要形成这样的逻辑图谱，本质上就是需要对不同逻辑节点进行组合和串联，真正的逻辑由封装在节点中的函数完成。那么这里就产生了两个问题，首先是如何抽象逻辑节点，抽象出的逻辑节点能不能被复用直接决定了用户编排的成本，如果需要不断的定制个性化逻辑节点可能就失去了编排的意义；其次是逻辑节点的颗粒度大小也非常关键，如果封装的逻辑颗粒度太大，大到一个功能服务，那么可能就变成了业务流程编排；如果颗粒度太小，小到一个表达式级别，那么对于有编程基础的同学来说，可能直接写代码效率反而更高。

通过对中后台营销域的部分业务代码进行梳理，发现中后台的页面大都是以表单、列表，详情为主，而其中90%的业务逻辑基本上都围绕在表单(校验，取值，赋值，提交)，对话框（显隐、提示），发送请求，消息提示，数据处理，路由跳转，条件判断等，相对比较收敛。同时iceluna作为集团内优秀的低代码搭建平台，在上层封装了很多非常好用的api，屏蔽了大部分前端语法层面的差异，比如状态赋值，页面刷新，接口调用，一些常用的工具函数（时间处理）等，为逻辑节点的抽象提供了极大的便利性。



通过分析归类，最后沉淀了10个左右的逻辑节点，如下图所示，同时每一个逻辑节点最终本质上都是对应一段执行函数，并接收一个参数作为入参，返回一个参数作为出参。



编排协议

由于是可视化编排，自然也避免不了编排的协议，为了避免重复建设最大程度的复用集团内已有的编排方案，最终计划采用LF通用可视化逻辑编排的协议来实现iceluna中的逻辑编排。


```
1 // 图
2 export interface Graph = {
3   // 节点
4   nodes: Node[],
5   // 连线
6   links: Link[],
7 }
8 export interface Node {
9   // 节点id 全图唯一
10  id: string | number;
11  // 节点名称
12  name: string;
13  // 节点ui信息
14  ui?: {
15    shape?: string;
16    // 节点位置 二维平面坐标
17    x?: number;
18    y?: number;
19    // 大小
20    width?: number;
21    height?: number;
22  };
23  // 节点数据: 配置信息、额外信息等, 由上层定义
24  data?: any;
25 }
26 export interface Link {
27   // 连线id 全图唯一
28   id?: string;
29   // 连线名字/标签
30   label?: string;
31   // 连线类型
32   type?: string;
33   // 连线起点 节点id
34   source: string;
35   // 连线终点 节点id
36   target: string;
37 }
```

技术架构图



技术难点

自动化布局

从一开始调研我们就发现大部分的编排产品，都是让用户自己进行拖拽，连线等操作去完成，但是通过前面对逻辑代码的分析，如果我们将异步回调操作使用async/await的方式转换成同步的写法，那么逻辑代码大部分都可以看作是一种串行式的执行过程，偶尔叠加一些if/else的分支判断，这样也非常符合人们常用的思维模式，理解起来非常直观。所以从编排的角度说，就是将不同的逻辑节点和分支判断节点串联起来，布局上不需要太多的灵活性，同时连线操作也显得比较多余，因此我们将拖拽连线全部改成添加节点的方式，然后自动连线即可。

采用这种自动布局的方式会大大简化用户的操作，用户只需关注核心的业务逻辑流程，按顺序添加节点即可，但

同时也带来一个问题，由于分支类型的节点会产生两个分支流，如果遇到嵌套分支的情况下，需要自动将上层分支的横坐标的位置统一向右偏移一个单位，否则处在上下层不同分支上的节点位置可能会产生重叠。为此，我设计了一自动布局算法，最终实现的效果图如下：



算法过程比较简单，采用的是DFS深度优先遍历算法，详细过程这里就不再赘述。

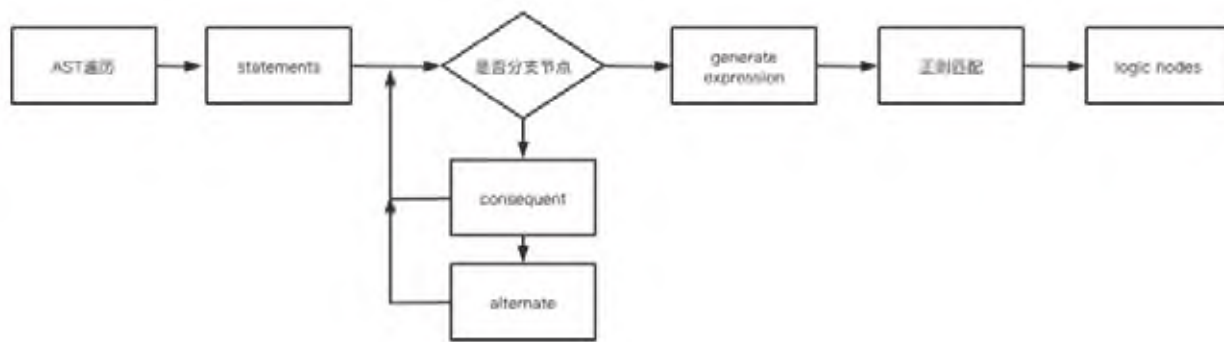
代码与schema互转

逻辑图编排完成之后，紧接着面临的问题是如何保证编排后的逻辑正确的运行，产生和源码一样的效果。一开始讨论的有两种方案，第一种方案把整个逻辑看成一个事件流，按照前面设计的逻辑编排schema，通过事件注册监听的方式完成逻辑节点的上下游串联，最后设计一套事件执行器，依次触发事件即可。这种方式实现起来比较简单，但是对原有开发流程的侵入性比较强。因为原有保存事件函数的地方都要被替换成逻辑schema，同时负责code review的前端同学看到的不再是代码diff，而是schema的diff，这无疑会大大增加了CR的风险。因此经过一番讨论之后，我们决定采用第二种方案，将逻辑编排后的schema自动转成代码，同时生成后的代码也要能够自动转回schema。

基于schema转成代码是比较容易的，因为每个逻辑节点本身就映射了一段函数片段，而将代码转回schema则稍微有些复杂。这里主要利用了Babel先对代码进行解析，得到抽象语法树AST，然后遍历AST中类型为statement的节点，最后通过正则匹配找到对应的逻辑节点，并串联好连线即可。下图就是生成好的一份代码示例：

```
"onSubmit": async function onSubmit(value, error, field) {
  return await this.utils.logicRun(async function(logic) {
    // 表单校验成功
    if (logic.callCondition('{{!ctx.error }}')) {
      // 发送请求, 保存策略
      await logic.callDataSource('createStrategy', '{{ctx.postData }}', 'res')
      // 策略保存是否成功
      if (logic.callCondition('{{ctx.res.data.success}}')) {
        // 保存成功
        logic.callMessage('success', '保存成功')
        // 关闭对话框
        await logic.callSetState({
          "createStrategyDialogVisible": "{{false}}"
        })
        // 刷新列表
        logic.callMethod('reloadListData()', [], '')
      } else {
        // 提交失败 消息提示
        logic.callMessage('warning', '提交失败')
      }
    }
  }, this, {value, error, field})
}
```

可以看出，通过schema生成后的代码与源码编写还是有一点区别，带有一些逻辑编排的特征，但是可读性更强，从代码基本也能直观的反推出逻辑流程图，反而从一定程度上降低了code review的成本。整个AST解析的过程如下：



断点调试

对于写业务逻辑来说，不可避免的需要调试功能，这对有编程能力的同学来说是件很自然的事情，但是当逻辑变成通过可视化的编排之后，如何让这些”非前端“用户也能方便的通过调试定位错误，变得也尤为关键。

调试其实本质上对用户来说，就是需要一个能够让编排后的逻辑模拟运行起来的过程，因此我们对逻辑节点的各个环节做了埋点，用户在模拟运行的过程中就能查看每个节点的数据状态、上下文参数、报错类型等，同时根据逻辑流程图的状态（绿色代表执行成功，红色代表执行失败）也能非常快速的定位问题所在，如下图所示：



目前调试功能还处在初级阶段，后面还会持续迭代优化，比如调试时增加单步执行功能，像浏览器的单步调试工具一样进行断点。

总结展望

总结

目前，可视化逻辑编排已经搭载集团内的iceluna低代码搭建平台正式上线，并已经在营销工具业务中正式使用。从低代码向无代码的研发模式升级仍处在探索阶段，过程中避免不了会遇到很多问题，但也算迈出了关键性的一步，值得期待。

展望

前面提到，从ProCode->LowCode->NoCode，通过降低研发门槛，让非技术人员参与到应用开发中来，可以大大提高生产效率，但理想很丰满，现实也很骨感，NoCode搭建平台我认为目前还只能在比较垂直的场景中才能适用，并且由于不像LowCode一样仍然能够写代码的逃离机制，通过NoCode的方式可能只能完成一个70分左右的产品。但是换一个角度看，如果可以让一个非技术人员，只用很低的门槛就完成一个70分左右的产品(最小可用产品MVP)，并能直接推广到市场去试错，如果验证可行，再通过转成LowCode或者ProCode的方式继续迭代优化。光这一点我认为就是很有价值的，是推动商业创新的核心驱动力。因此我认为未来的产品研发节奏可能是从NoCode->LowCode->ProCode，每一流程都要有对应的解决方案，并且互相之间能够相互打通，只有这样才能在竞争日益激烈的市场环境下更好的生存。

• 2021年了，前端最应该学什么？

- 作者：狼叔
- 出品：淘系技术

诚如很多大V所讲，确实是2020年，前端圈带来具有突破意义的内容或框架不多，很多人也不会有2013年到2017年间日日新的框架大战局面，也不会有Node全栈之争，也不会因为React-Native、Weex、Flutter这类跨端而欣喜若狂。

我能看到的是今天前端已趋于稳定，在深水区探索，比如蚂蚁金服的x6，在图形可视化方面做的就是非常好，比如淘宝的midway-faas，在Serverless领域确实有它独特的定位。比如语雀，钉钉文档，在线Excel等等，也都不是可以轻松可以搞定的。

我很开心的看到，混乱之后，大家都能在深水区里进行探索。2019年阿里经济体前端委员会四大技术方向：第一搭建服务，第二是 Serverless，第三是智能化，第四是 IDE。2020年阿里经济体前端委员的突破方向是互动技术、跨端技术、智能化。而中后台、数据可视化、Node.js（Serverless）、工程体系（安全生产）都变成了基础技术方向。这大概是能够代表前端技术走向的。

我个人也走过类似的路，2017年加入阿里，将PHP替换为Node.js，随后搞了开源项目egg-react-ssr，然后在2019年加入前端委员会，负责Serverless-side render方向。在2020年，转岗到淘系前端，负责前端智能化相关开发。我其实是非常看好Serverless的，Serverless这种稳步推进的必然是前端新基建，未来玩2到5年问题不大。对我而言，前端智能化的诱惑更大，能够站到产研链路是思考问题，这才是我梦寐以求的机会。

http import 会大行其道

其实就是Deno创造的方式，Deno被评为2020最佳开源贡献也是实至名归的。

```
import cheerio from "https://dev.jspm.io/npm:cheerio/index.js";
```

把cjs转esm都交给CDN类的服务来做更合适。事实上，pika.dev/skypack.dev/http://jspm.io都已经做了这件事儿。

Node.js马上跟上，相关PR早已在路上，此项必火。

逻辑编排，更加面向开发者

已收到很多imove类似项目。解决逻辑可变和不直观的问题。



以函数为粒度，继而通过运用配置类的操作，将逻辑可视化，配置化。用法极为简单。参见 <https://github.com/imgcook/imove>

智能UI精细化

首先服务端搜索瓶颈已经到了天花板，端智能和端UI的探索，一定是增量上提升业务指标的。

鲸幂智能UI对接链路



智能化 PRD 2 Code (P2C)

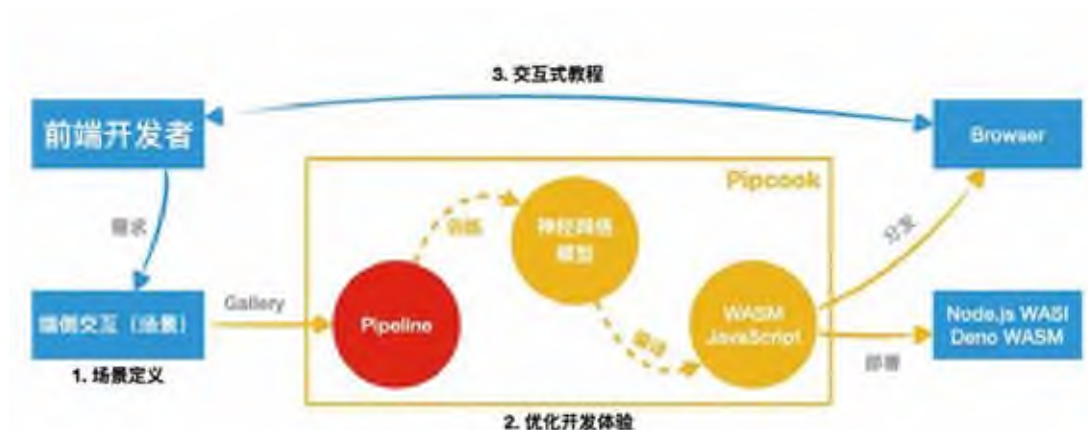
站在产研链路审视研发效率问题。站在D2C（设计稿转代码）之上，引入PD产品经理标注方式提升出码，进一步做到无人工，真正智能化。（招人做此项目）

下面2020年D2前端大会上ppt分享的一页，PD标注业务含义讲的还是比较清楚的。



不会 Python，前端也能搞机器学习

github.com/alibaba/pipcook, 基于pipeline思路抽象的AI基础框架, 让AI落地更简单。



一些我关注的开源项目

midway-hooks

最好用最潮的Serverless同构框架，没有之一。开源地址：

<https://github.com/midwayjs/hooks>

Midway Serverless 云端一体特性

云原生微服务大会
Microservices Summit

1 函数式研发
Functional R&D Solutions

函数即接口，我们通过函数来统一后台端的体验，减少不必要的样板代码，加速应用研发。

2 一体化调用
Api not required

从此不再手动调用 Api。在跑框侧中，你可以直接从服务端导入函数并调用，就像调用普通函数一样简单。

3 Hooks

Using Node.js like React Hooks

通过 Hooks 开发 Node.js 应用，是全新的体验，也是你熟悉的语法，更为函数式开发带去了完善的能力支持。

4 渐进式开发
Progressive development

简单和复杂场景适配，通过对于 IsC 的支持，我们可以复用阿里 Node.js 复杂应用最佳实践，支撑企业级应用开发。

Midway Serverless 的云端一体方案

云原生微服务大会
Microservices Summit

定义：云+端一同开发、无缝融合

易于开发

易于维护

易于部署

```
graph LR
    subgraph "File Structure"
        public
        src
        subgraph "apis"
            hooks
            lambda
        end
        subgraph "pages"
            about_tsx[about.tsx]
            index_tsx[index.tsx]
        end
        subgraph "test"
            component_ts[component.ts]
            lambda_ts[lambda.ts]
        end
        gitignore[.gitignore]
        readme[README.md]
    end

    use_hydrate_ts[useHydrate.ts] --> CustomHook[Custom Hook]
    about_ts[about.ts] --> Api[Api]
    index_ts[index.ts] --> Api
    about_tsx --> Webpage[Webpage]
    index_tsx --> Webpage
    component_ts --> Test[Test]
    lambda_ts --> Test
```

imove

逻辑编排工具，开发是有快感的。基于阿里开源的x6和formrender，简单易用。

已开源，<https://github.com/imgcook/imove>

ykfe/ssr

基于Serverless的端渲染方案。支持多个Faas环境。同时支持csr和ssr无缝降级的方案。基于之前成熟的egg-react-ssr，去掉Eggjs，改成midway-faas，天然一套支持跨运营商。

使用本框架开发 Serverless SSR 应用开发流程



相比于传统服务端应用开发，我们将细节在底层统一抹平。前端开发者只需要关注业务逻辑，无需感知服务器的运行状况。成本和心智负担大大降低，只需要申请一个域名即可将应用发布到公网让所有用户可以访问。

🚀 表示已经实现的功能

里程碑	状态
最小而美的实现 React 服务端渲染功能	🚀
约定式前端路由	🚀
All in JSX，抛弃传统模版引擎，所有部分包括 layout 布局皆使用 JSX 来编写生成	🚀
渲染模式切换：服务端渲染一键降级为客户端渲染	🚀
统一服务端客户端的数据获取方式	🚀
类型友好，全面拥抱 TS	🚀
支持无缝接入 antd 无需修改任何配置	🚀
支持使用 less 作为 css 预处理器	🚀
实现 SSR 场景下最优秀的代码分割方案	🚀
接入 useContext + useReducer 实现极简的 数据管理 ，摒弃传统的 redux/dva 等数据管理方案	🚀
支持在阿里云 云平台 创建使用	🚀
ssr deploy 一键部署到 阿里云平台	🚀
ssr deploy --tencent 无需修改任何配置一键部署到 腾讯云平台	🚀

已开源，开源地址 <https://github.com/ykfe/ssr>

airpack

阿里内部的支持http import和cjs转esm的高效构建工具，据说已经在筹备开源工作了。看到一个性能压测，airpack大约是webpack5的20倍左右。

Benchmark

webpack5
38.7s

```
jeason@admindMacBook-Pro-2: ~/projects/gitlab.alibaba-inc.com/airpack/airpack-bundless-demo
airpack-bundless-demo @ |> master [??] is v1.0.0 via v10.20.1 took 5s
> time on run benchmark:webpack

✓ webpack
Compiled bundlessBundle.js (4.34.40s)

assets by status 54 bytes [cached] 1 asset
asset index.js 1.07 MiB [emitted] [minimized] [big] (name: index) 2 related assets
asset index.html 334 bytes [emitted]
webpack 5.11.8 compiled successfully in 16998 ms
dn run benchmark:webpack 38.72s user 3.04s system 192% cpu 21.691 total
```

airpack
1.8s

```
jeason@admindMacBook-Pro-2: ~/projects/gitlab.alibaba-inc.com/airpack/airpack-bundless-demo
airpack-bundless-demo @ |> master [??] is v1.0.0 via v10.20.1
> time on run benchmark:airpack
[10:34:40] Babel transform starting...
[10:34:40] Transform to esm for src/App.tsx
[10:34:41] Transform to esm for src/index.tsx
[10:34:41] Babel transform finished.
dn run benchmark:airpack 1.88s user 0.41s system 74% cpu 2.971 total

airpack-bundless-demo @ |> master [??] is v1.0.0 via v10.20.1 took 1s
> |
```

不确定的是 react server component
仍需要时间去了解。

总结

2020年各大厂应该都在困惑，老项目为提升业务指标发愁，新项目在为研发提效发愁。很多既得利益者，吃着所剩不多的红利，一方面担心被替换，舍不得放弃，一方面又不敢做改变。我的观点是服务端算法（包括搜索推荐）今天已经触及了天花板，再提升一个点都会比之前更难。传统前端基建也面临一样的问题，比如node，搭建，ui框架，对于下一代升级想法，大概也是缺少想法和目标的。

创新是需要勇气的，眼界不够的人不能做到，能力不足的人不能做到。前端和AI结合的跨技术融合项目是存在非常大的机遇和挑战的。甄子以一人之力，扛起前端智能化大旗是非常不容易的。目前imgcook在设计稿转代码领域已经取得阶段性结果，但我们还有如下探索。

- 智能UI，目前已经能够看到增量的点。头条也做了类似的事儿。
- P2C，围绕业务标注，实现产研提效，已验证。
- Pipcook会进一步简化ai开发，只要有数据就能训练模型，真的是有手就行。
- Design+，设计资产管理。
- 像airpack、http import、imove、midway-hooks、ykfe/ssr这些其实都会成为前端新基建。

总结一下，笔者认为前端智能化是2021年最有前途的方向。

很多人都以为前端智能化对ai和算法要求极高，其实这个看法是片面的。在前端智能化团队里有3种事儿可以做：1）业务，2）工程，3）算法。其中工程和业务是不需要算法的，对于新人也是会给缓冲期的，可以先做擅长的事儿，同时跟着团队向ai算法方面学习。

我经历过的阶段：

- 熟悉imgcook，这是d2c领域。覆盖了 2020 年双 11 会场 90%+ 的模块开发，出码可用率达到 79.26%，且需求吞吐量提升 1.5~2 倍，给前端研发带来实质性的提效。因为我是双十一PM，这个我是相当知道。
- 接手P2C，建立pd标注体系。这个过程是很困难的，但也是很有成长的。其实，更多的我是一个产品的角色。站在D2C的肩膀上，站在用户视角（PD），保证项目方向不歪。对于pd能做什么，该怎么做，如何快速落地业务，我是有很多思考和成长的。
- 搞定API，以前都是先选数据源然后确定字段，这是很麻烦的，PD是无法接受的，如果api有100个，每个api有10个字段，pd就疯了。我们的做法是先选字段，然后再确定具体接口，这种逆向思维，在这种项目里是非常适用的。
- 继续深化C端解决方案，站在淘系业务和技术都很成熟的前提下，提升业务数据，又能兼顾技术创新，大概不会有比这还胆大包天且令人激动的目标了。

• 如何看待 svelte 这个前端框架？

- 作者：Rich Harris
- 出品：淘系技术

我看排名靠前的答案在利弊上总结都挺到位的了，分享一个让你的react代码跑在svelte引擎上的具体实践。

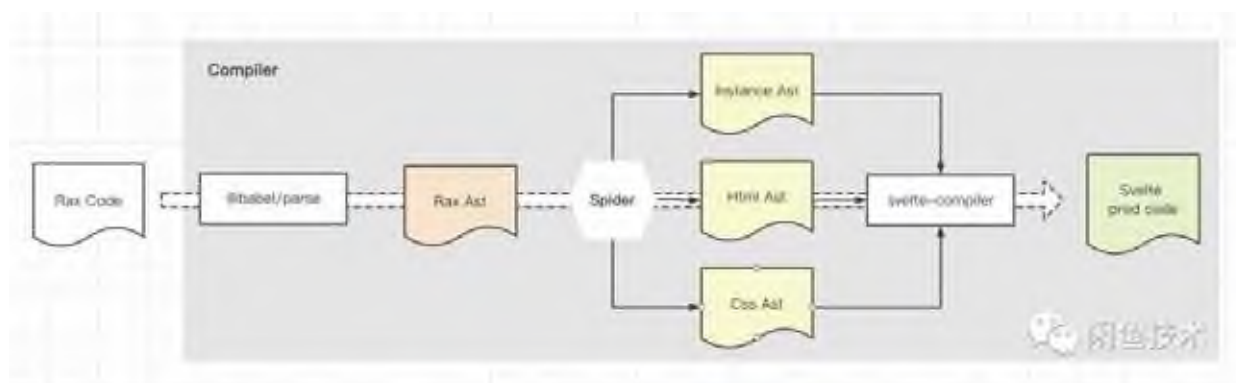
我自己的看法还是围绕着运行前编译这个idea，一方面通过编译魔法可以让响应式的写法这么自然原生（当然还可能可以开启更多魔法，只要有想法）；另一方面我比较印象深刻的是它为了实现runtime按需打包可以把一个UI框架定义的这么零件化，我们也是借助这个编译+零件化重新设计了现有的一些UI组件，然UI组件也能做到按真实使用情况来打包。

Svelte UI框架是一款类似react、vue一样的UI框架，有兴趣的同学可以自行查阅。Svelte有着很多不一样的特质，其中我们最为关注的是它的运行前编译。像更简洁灵活的响应式写法、更小的运行包体积这些都得益于运行前编译；但同样也存在它的局限性，Svelte无法像React一样做到高度灵活的模板嵌套、封装以及独立上下文。早期在小项目中我们也体验过，在包体积上收益颇大；但目前整个淘系前端还是有着自己统一的一套基于React的Rax framework，如果说需要将Svelte的收益快速落地最好的方式肯定不是重构所有的业务与生态。

下面详细分享下我的实践思路。

主体思路

要实现dsl层无感换引擎也就只有在ast层做转换，所以我们从开始的思路就是React ast -> Svelte ast，为了保证这一思路的清晰我们暂且把框架的特性（hook、函数jsx）丢掉，保留最干净的三部分jsx、js、css进行转换。主体思路如下



JSX转换

jsx可以说是这两个UI框架最大差异的体现；React的jsx是运行时，原汁原味保留着原生js运行时的灵活性，而Svelte的html是运行前编译，意味着运行前就会敲定html中的逻辑与依赖变量。我暂且将jsx本身拆分为三部分：element节点、attribute属性、mustache模板系统、函数JSX

element与attribute

这部分主要是常规的节点属性映射能力



mustache模板系统

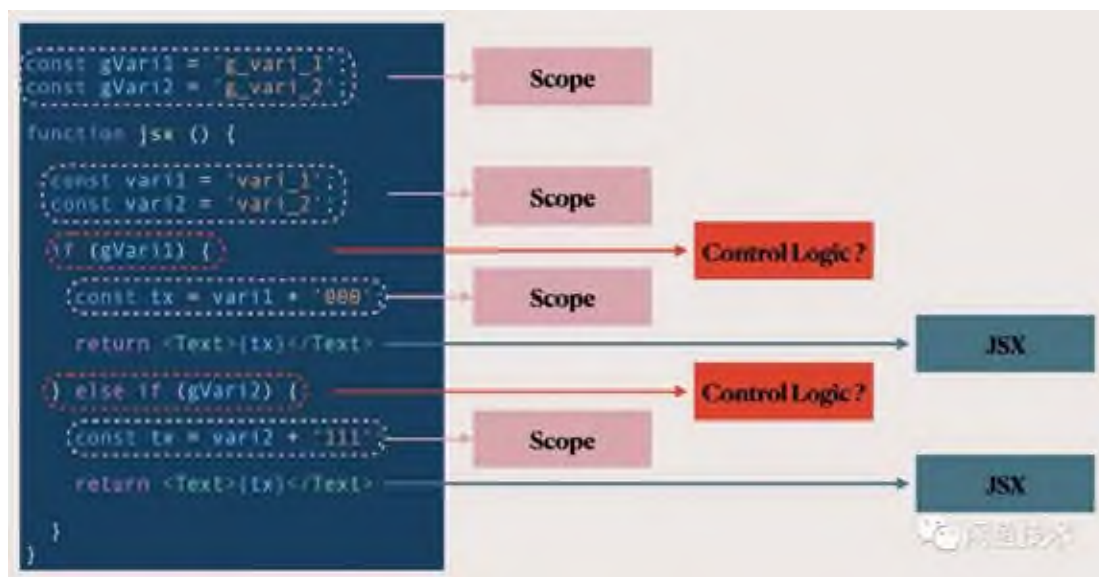
在React的模板系统中可以有以下几种可能：1、变量输出 { xxx } 2、与运算jsx输出 { xx && <view> } 3、三元运算jsx输出 { xx ? <n> : <m> } 4、函数调用jsx输出 { jsxFn() } 其中第1种情况不需要处理，Svelte的模板系统也是能够支持常规的变量输出；而第4种属于函数jsx；第2、3种可以转换成Svelte提供的{#if ...}语法糖。



至此，简单jsx的ast就能编译成svelte html的ast。

函数JSX

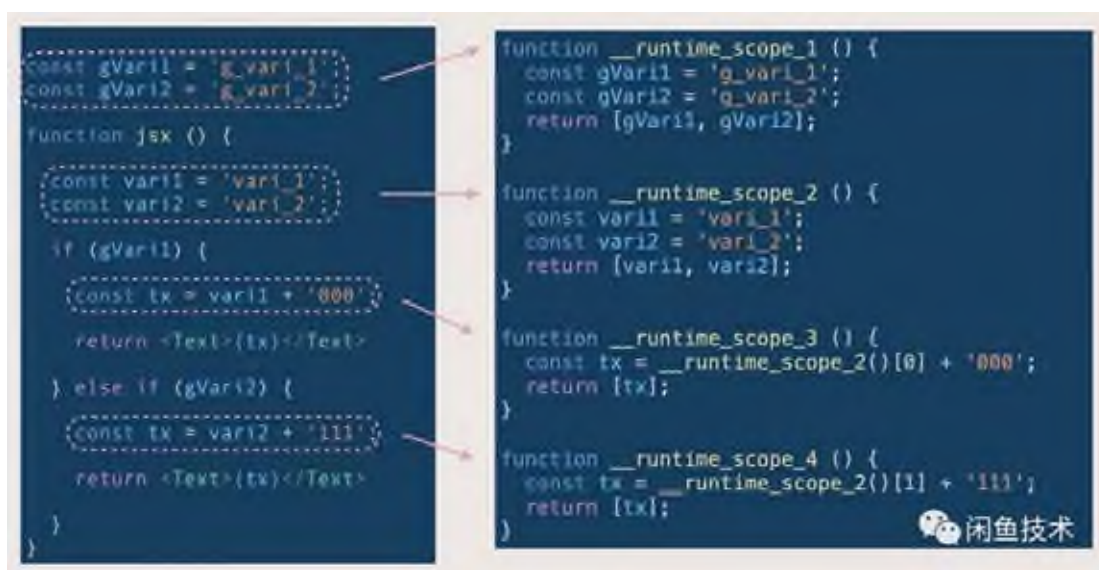
函数jsx背后折射的是js运行时，意味着可以封装独立的运行上下文(scope)，可以做到环境变量隔离。先来看一个常规的函数jsx，再进一步抽象成三部分：上下文、逻辑控制、JSX。



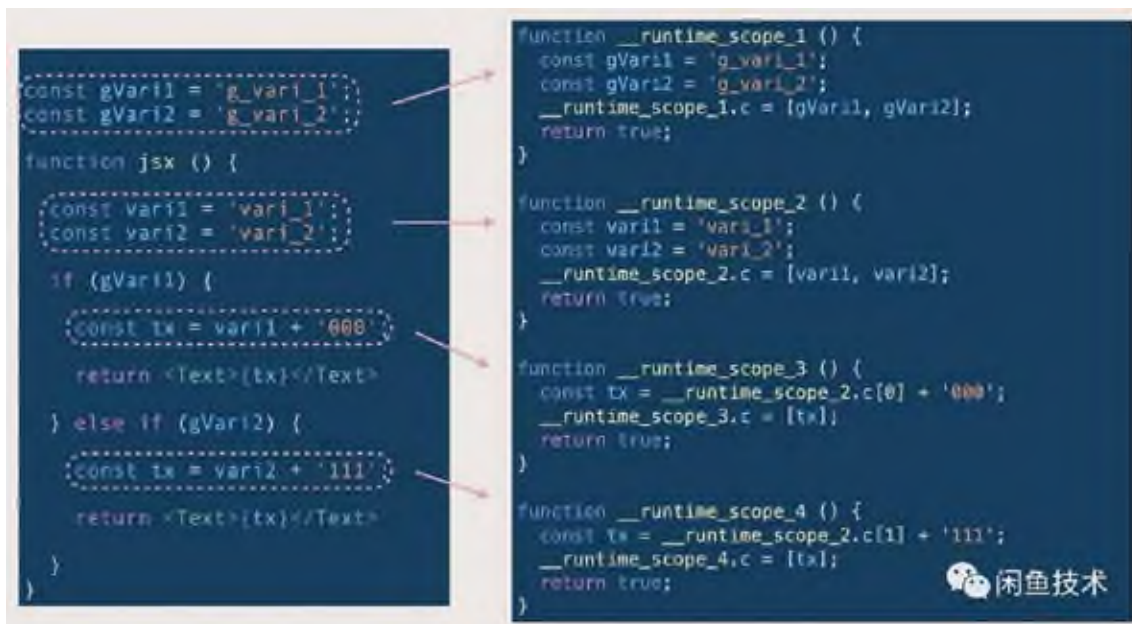
其中JSX部分的转换可以复用上文讲到的element与attribute部分。我们重点来看上下文与逻辑控制的转换。

上下文(scope)

尝试使用函数模拟出完整的执行上下文



优化调用次数



逻辑控制

这里我指的逻辑控制是控制jsx输出的逻辑（非jsx输出的控制逻辑不需要关心），在一个jsx函数中，逻辑控制语句可有、可无、可组合、可嵌套，在js中大致常用的逻辑控制语句大致有以下几种：1、if else2、switch case3、for/while/do while4、return is ? <viewa> : <viewb>5、return is && <view> 根据Svelte提供的可编译html逻辑控制语法糖来做个映射：

React	Svelte
if else	{# if }
switch case	{# if }
for/while	{# each }
return is ? <viewa> : <viewb>	{# if }
return is && <view>	{# if }

```
function jsx () {
  const vari1 = 'vari_1';
  const vari2 = 'vari_2';

  if (!gVari3) return;
  if (!gVari4) return jsxFn();

  if (gVari1) {
    const tx = vari1 + '000';
    return <Text>{tx}</Text>
  } else if (gVari2) {
    const tx = vari2 + '111';
    return <Text>{tx}</Text>
  }
}
```

[illegible]

看着产物有些恐怖，极其膨胀，但这也给后面优化留够了空间。

JS转换

js层面的差异主要来自两方面：hook体系（不考虑类）、ecma ast差异

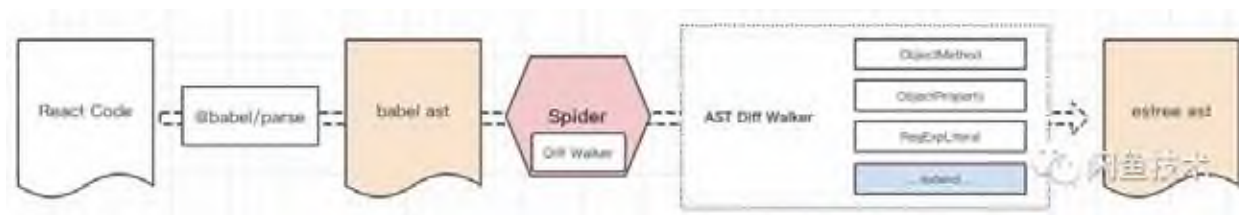
hook体系

hook体系的api更多是为了纯函数组件注入状态与生命周期的，在这两点上Svelte给出的方案是截然不同的，得益于运行前编译，Svelte编译器扫描了所有与UI相关的状态并注入黑魔法，使得状态的使用就是变量申明赋值那么简单，基本上开发者不需要太关心所谓的副作用；所以有一些hook接口在Svelte框架上显得是有些多余的。但鉴于hook接口较多，我们在转换过程选择了内置svelte-hooks来简化转换逻辑，svelte-hooks是基于svelte且对标react hook来实现的一套hook接口，在使用上基本保持一致。



ecma ast差异

babel提供的parse是基于estree，但同时在基础上细化出一些类型，具体可以差异可以参考这里 babel-parser[1]，细化的这些数据类型在我们做转换推断有一定帮助，所以我们并没有使用babel提供estree插件，而且在转换之后的ast再进行一次差异抹平。



CSS转换

css转换相比上面两部分的转换就要简单很多，React的样式是标准的css，Svelte编写的样式也是标准css，不过会在基础上增加了一定的编译能力，可以理解是标准css的超集，可以直接使用。不过为了抹平jsx与Svelte html在自定义组件的类选择器上的差异，我们还是在编译阶段做了一些转换，这里就不展开了。大致流程如下

结果

一个重构过的业务（只含UI相关逻辑）

基于Rax DSL

- o rax构建
 - o 压缩混淆：116.71k
- o 编译转换
 - o 压缩混淆：63.18k ↓46%



细化后的体积差异

	Rax构建	Spider编译
编译产物	34.13k (含样式)	34.5k (不含样式)
framework	32.52k	16.3k
依赖的UI Component	47.6k	10.83k

客观对比：编译产物是要比正常打包要大的，从前面的函数jsx编译思路来看是要膨胀的，在预期内；拉开差距的是framework与依赖的UI Component，Rax是集团内部基于react进行简化后的framework，如果使用React framework做比较这个差距会进一步的拉大；而UI Component这部分的差异是得益于我们参照运行前编译的思路重新设计了一套可在编译阶段根据具体情况来按需打包的轻量UI组件（本文没有具体展开）。

后续

- 1.将运行时的jsx百分百准确编译成静态的html是不可能的，弱类型语言的变量追踪是不可靠的，非原生逻辑控制语法也无法一一在编译器中枚举；目前在转换工作中还遗留很多编译相关问题，但这些问题可以通过一些插件来补充以致逐渐完善。
- 2.在大型项目中包体积的现状并不乐观，Svelte借助着运行前编译将整个framework按需打包可以有效减少包体积，但编译产物本身是没有优势的，当一个页面UI交互越复杂，编译产物会越大，加上对framework依赖越多，整体包体积的优势就荡然无存了；除此之外，我们转换器为了抹平差异又给编译增加了一定复杂度，所以在编译产物体积的优化上还存在很大的空间。
- 3.无性能不前端，在性能方面的数据我们还是缺失的，但也从一些三方文章了解到整体Svelte性能并不是瓶颈，而且从理论上来说借助编译来实现数据驱动DOM是简洁高效的，脱离Virtual DOM理论上也让内存表现更优异；但我们还是会单独去看性能一块的具体表现。
- 4.运行前编译的思路不单止应用于framework，component同样受用，而且收益颇大。

客户端篇

• iOS 性能检测新方式 ——AnimationHitches

- 作者：岚遥
- 出品：淘系技术

AnimationHitches 的运行原理

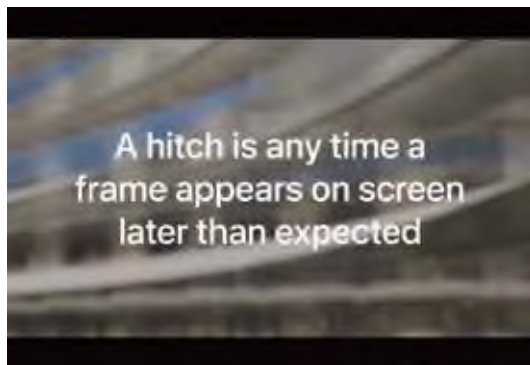
背景

在 Xcode12 中, Instrument 新增 AnimationHitches 检测类型用以检测卡顿, 并去除 CoreAnimation 检测方式。在支持 PromotionDisplay 的设备上帧率可调整至 120 帧, 并且会根据当前用户手势和设备状态进行动态调整。此时再继续使用帧率来判断性能的好坏及流畅度将会是一个错误的选择。所以 AnimationHitches 主要用于代替帧率检测, 并且提出 卡顿时间比(Hitch Time Ratio) 的概念用于替代 FPS。由于目前关于 Hitch 相关的资料很少, 而在 iPhone13Pro 之前 iPhone 屏幕最高刷新频率仍为 60 HZ, 所以很多同学都还未关注到该能力。所以本篇将主要介绍 Hitch(卡顿) 的概念、RenderLoop(渲染循环) 的整体流程, 卡顿类型及如何避免卡顿。

什么是卡顿?

概念

任何时候屏幕上出现晚于预计的帧都属于卡顿。

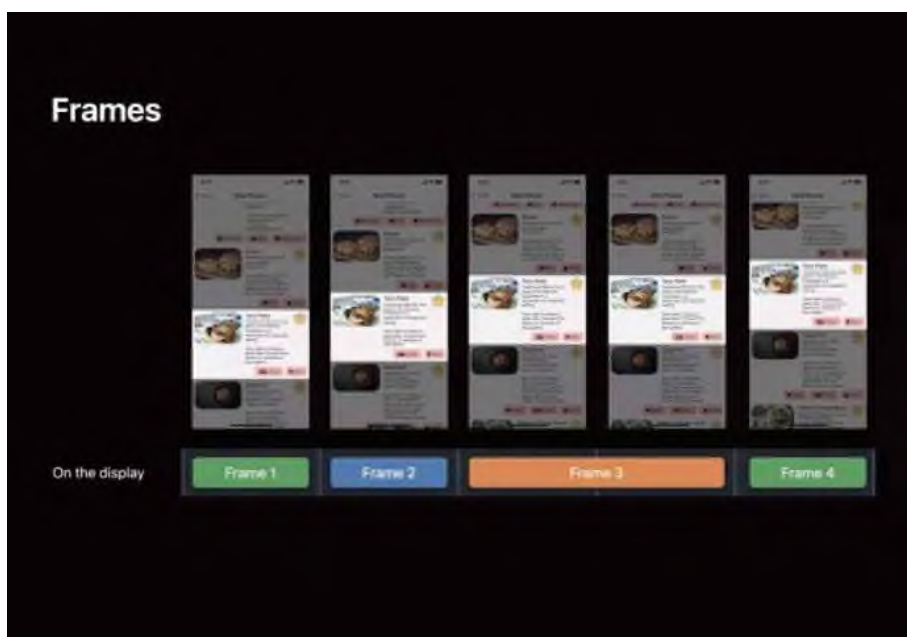


实例

例如 滚动动画(Scroll)、点击动画(Animation)、转场动画(Transition)，这些流畅的动画构建了一种用户和屏幕内容的视觉连接感，而如果动画卡顿会导致动画画面跳跃，打破这种连接感，用户体验会变得很差。



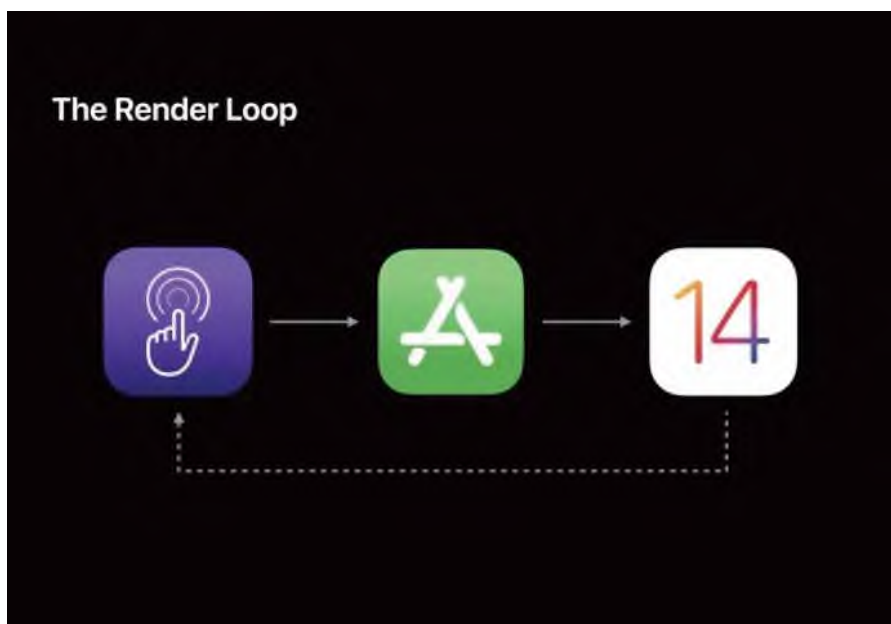
一个常见的例子，当用户在进行一个滚动视图上下滚动时，发生了卡顿，这是因为第四帧的延迟导致了第三帧占用了两帧的时间，给用户看到的的就是卡顿掉帧的现象。



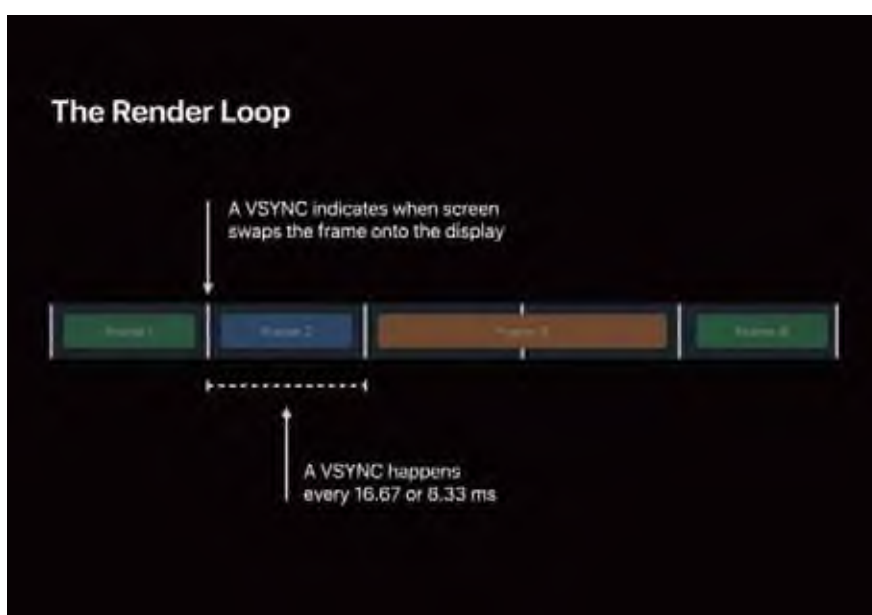
RenderLoop

概念

RenderLoop 是一个连续的过程，通过用户手势等将事件传给 App，接着 App 向操作系统传递事件并最终响应事件，再将响应传递给用户的过程。



RenderLoop 的时间随着设备刷新频率，在 iPhone13 Pro(Max) 以下的 iPhone 设备最大均为 60 帧，而 iPhone13 Pro(Max) 及 iPadPro 则最高支持 120 帧，也就是最短仅需每 8.33 毫秒就可以显示一个新帧。



帧准备阶段

在准备每一帧的过程中，可以总体分为三个阶段。App、RenderServer 和 Display。其中 App 中主要进行一些用户事件的处理，而 RenderServer 会进行真正的用户界面绘制，这两个阶段都需要在下一个 VSYNC 到来前完成。最终到 Display 阶段会将缓冲的帧展示出来。对这一帧进行双帧处理我们把这称之为双缓冲，由于显示器是逐行扫描进行画面显示，双缓冲和垂直同步机制避免了屏幕撕裂的现象。



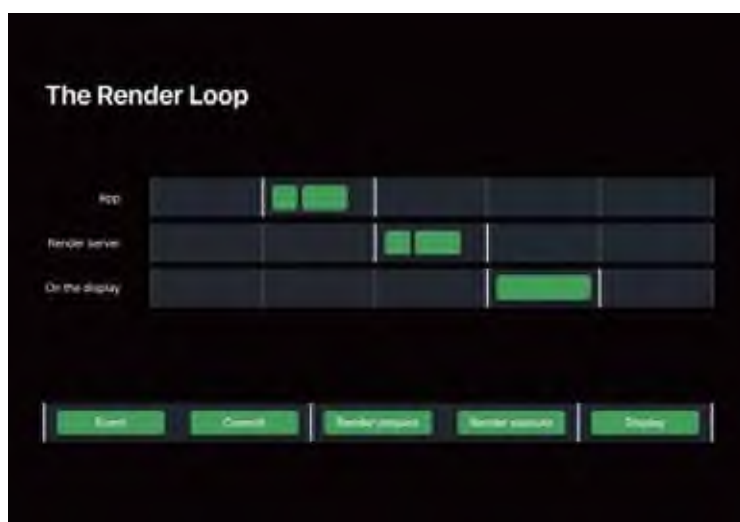
当然，系统也提供备用的三缓冲机制，为 RenderServer 提供额外的一帧进行渲染，该机制通常情况下不会开启。



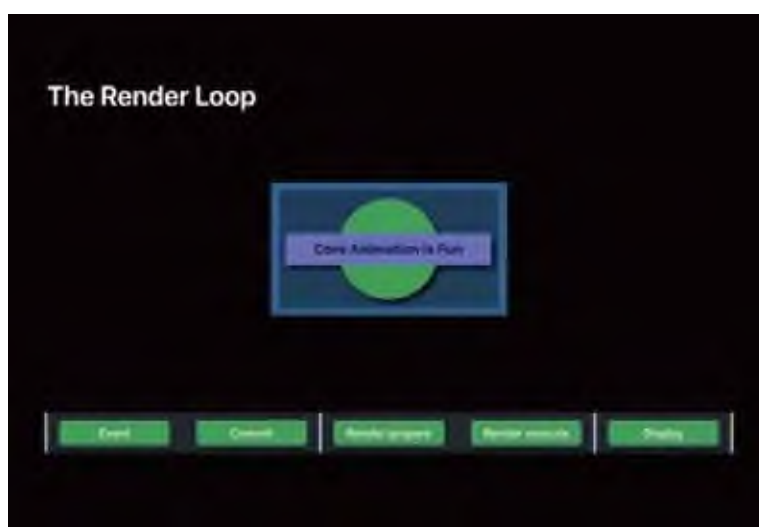
阶段细节

整个渲染循环可细分为 5 个阶段，其中在我们 App 中的为 Event，Commit 阶段，而 Commit 阶段可进一步细分为 Layout、Display、Prepare 和 Commit。

- 1.在事件阶段通过 touch，timer 等事件决定用户界面是否需要改变；
- 2.而在 Commit 阶段 App 会向渲染服务器 RenderServer 提交渲染命令；
- 3.RenderServer 中的 Prepare，Execute 阶段，在 Prepare 阶段会为 GPU 的绘制做好准备，而在 Execute 阶段会由 GPU 将用户界面的图像绘制出来；
- 4.最后的 Display 阶段会将缓冲帧交换到屏幕上显示。



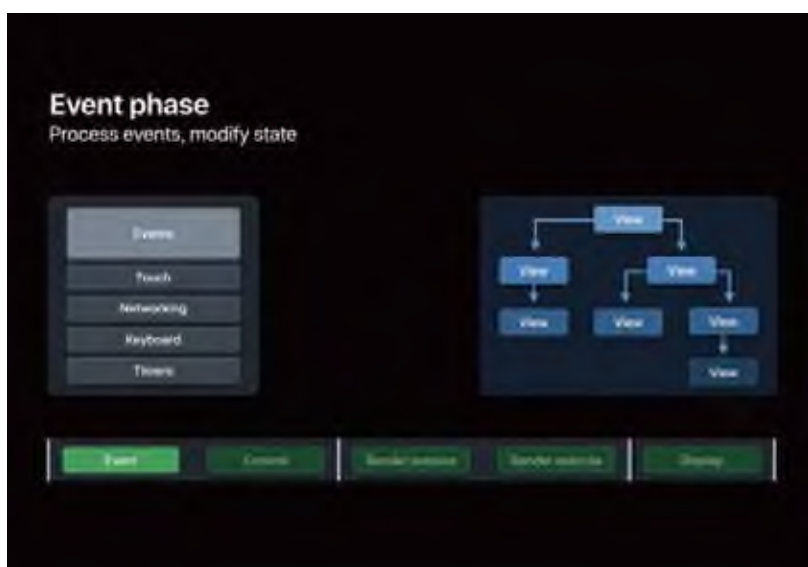
以一个带有阴影的渲染图形为例，观察下 RenderLoop 中每一帧所做的工作



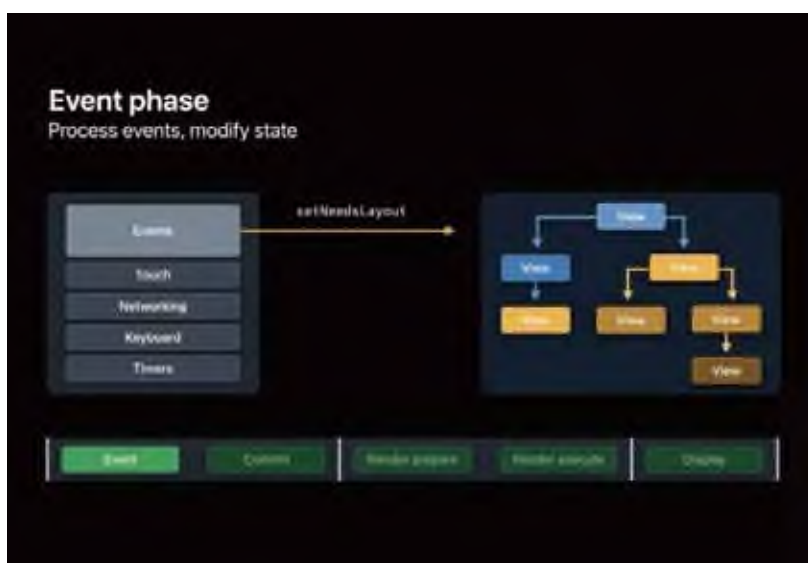
App

Event

在该阶段表示 App 接收到了事件，比如 touch 事件、网络请求回调、键盘和 Timer 。一个 App 可以通过改变其层级结构或是用任何其他方式响应这些事件。



例如 App 能改变图层的背景颜色，甚至能改变图层的大小和位置。当 App 更新了图层的限制范围时，CoreAnimation 会同时会调用 `setNeedsLayout`。它能够分辨哪些图层必须要重新计算布局，系统会合并这些需要布局的请求并在 Commit 阶段按顺序执行，用以减少重复工作。



Commit

在一次事务的提交中共涉及四个不同的阶段：布局阶段、显示阶段、准备阶段和最后的提交阶段。

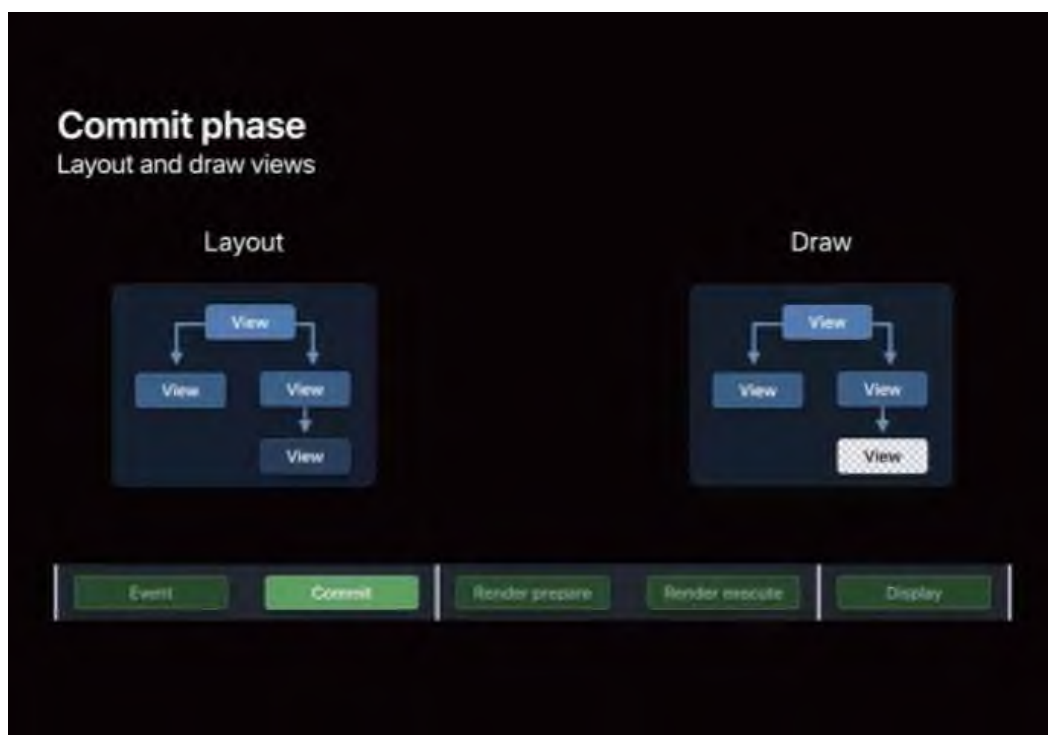


布局阶段

在布局阶段，`layoutSubviews` 会被所有需要布局的 View 调用。比如布局视图（`frame`、`bounds`、`transform`），增加或移除视图，亦或是直接调用 `setNeedsLayout`。注意这些布局操作并非立即执行，系统会合并这些布局请求，在 `RunLoop` 休眠前统一执行这些操作。

显示阶段

在显示阶段，`drawRect` 会被每个需要被更新的 View 调用。比如 `UILabel`、`UIImageView` 或者只是任何重写 `drawRect` 方法的类。他们必须调用 `setNeedsDisplay` 用以支持 View 的更新。在绘制时每个自定义的绘图图层都会接收到带纹理的 `CoreGraphics` 的背景。他们将利用 `CoreAnimation` 进行绘制，这些图层就变成了图片。所以如果没有必要则不要重写 `drawRect` 方法，其不仅会额外开辟一块内存用以存储 `bitmap`，还会在 CPU 上进行绘制，增加了整体主线程时间占用，当自定义 `drawRect` 视图较多时，对整体的内存压力也比较大。



准备阶段

在 Prepare 阶段还没有解码的图像将会在这一步进行解码，也就是我们需要优化的常见的图片主线程解码操作。

对于每个被解码的图像，App 可能会持续存在大量的内存分配。这种内存分配与输入图像的大小成正比，而与 FrameBuffer 中实际渲染的图像视图的大小没有必然联系。当 App 占用越来越多的内存时，操作系统将会开始压缩物理内存(physical memory)。整个过程都需要 CPU 的参与，所以除了我们自己的 App 对 CPU 的使用外，还可能会增加无法控制的全局 CPU 使用率。最终，我们的 App 可能会消耗更多的物理内存，以至于操作系统需要启动终止进程，它将从低优先级的后台进程开始。如果我们的 App 对内存的消耗达到了特定数量，可能会被终止，这也就是为什么经常会因为大图的原因产生 OOM。

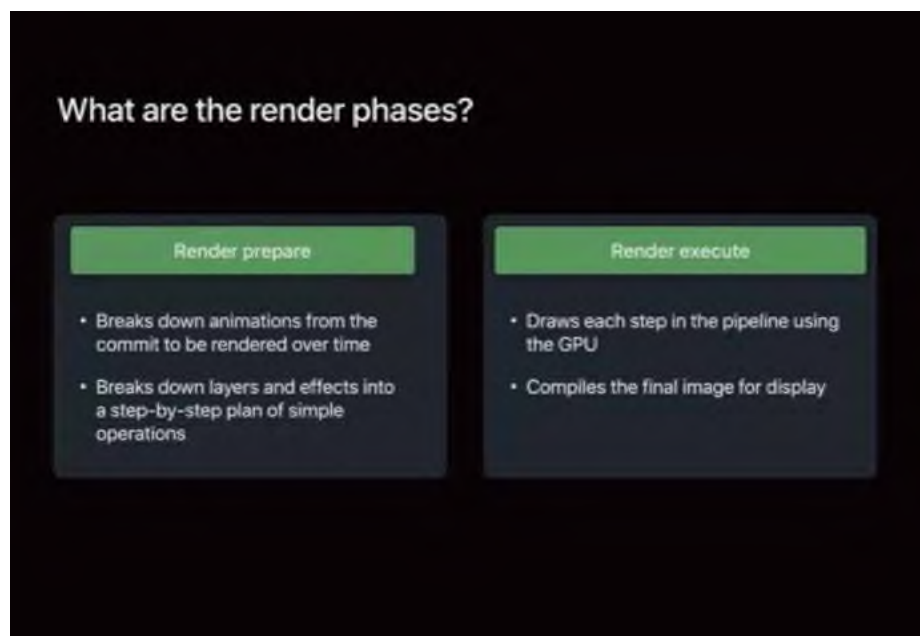
若某个图像的颜色格式 GPU 无法直接使用，也会在这一步进行格式转换。这就要求对该图像进行 copy 操作，而不是直接使用指针，这样会耗时更长及占用更多的内存。

提交阶段

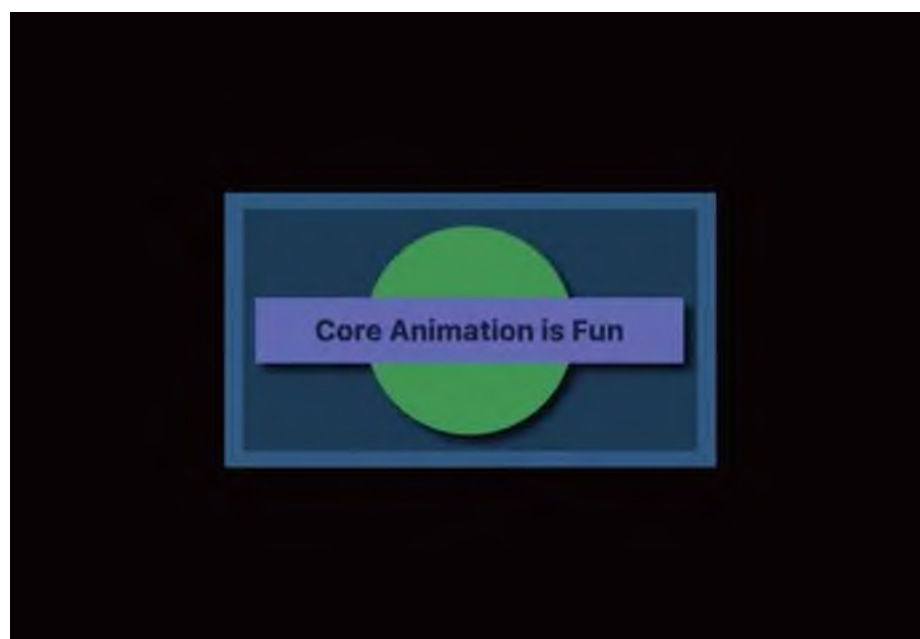
在提交阶段中，视图树将会被递归打包并发送到 RenderServer 中，所以当视图层级较为复杂时，这个过程耗费的时间也会更长一些，所以需要尽量减轻视图层级结构。

RenderServer

RenderServer 负责将我们的图层树转换为真正可显示的图像。RenderServer 有两个阶段：Prepare 和 Execute。在 Prepare 阶段我们的图层树被编译成一系列简单的指令，供 GPU 执行，帧动画也在此处进行处理。在渲染执行阶段 GPU 将 App 的图层绘制成最终图像。



下面来一个渲染实例。在下面这个实例中，圆形和长条周围都有阴影。



Prepare

在准备阶段，RenderServer 会广度优先遍历 App 的图层树，准备一个线性管线，这样 GPU 就能按照顺序执行命令进行绘制。从根图层开始逐层遍历，最终才有了 GPU 可以在下一个执行阶段执行的整个管线。



Execute

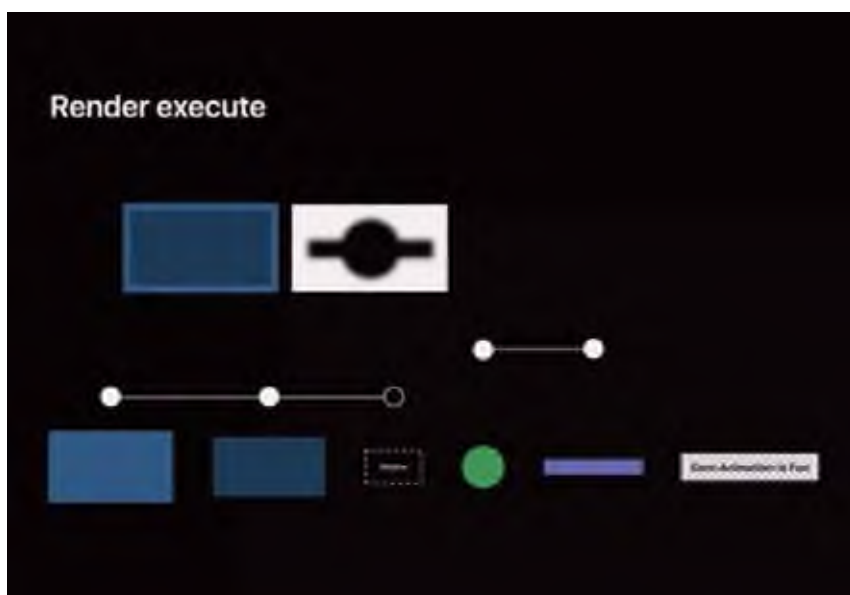
执行阶段主要是由 GPU 根据前面 prepare 阶段准备好的图层树进行顶点着色、形状装配、几何着色、光栅化、片段着色与图层混合。一旦 GPU 执行完会将渲染好的图像放入帧缓存区中等待下一个 VSYNC 的到来并交换到屏幕上进行显示。



在该例中，GPU 的工作就是利用该管线将每一步都绘制成纹理并最终合成，最终在显示阶段会在屏幕上显示该纹理。

从第一个蓝色的图层开始，它在指定的边界内绘制颜色。然后深蓝色被绘制在其边界内，但是当前圆形和矩形中都有阴影，所以现在 GPU 必须先绘制阴影。而阴影的形状由还未绘制的两层定义，所以需要先绘制圆形和矩形，为了避免这两图层被阴影遮挡，所以需要切换到不同的纹理先绘制阴影，对于这种情况我们称之为“离屏渲

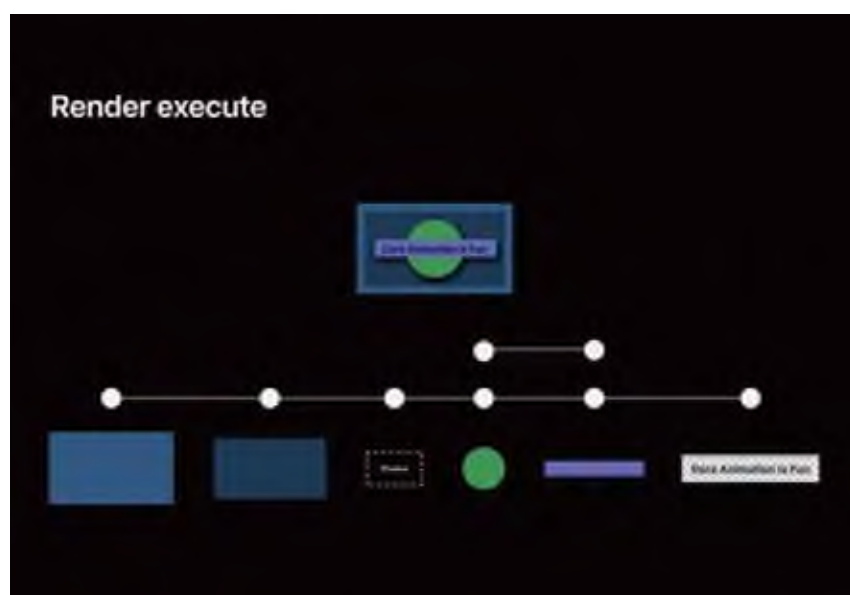
染”。在这里需要额外开辟一块内存用以绘制圆形和矩形，然后将该图层变为黑色并且模糊来实现阴影的效果。



然后 GPU 可以将阴影的离屏渲染纹理复制到最终的纹理中。阴影图层就完成了，下一步是再次绘制圆形和矩形。可以注意到的是，这里不仅开辟了一块额外的存储空间用以渲染阴影，圆形和矩形也被渲染了两次，对性能损害极大。

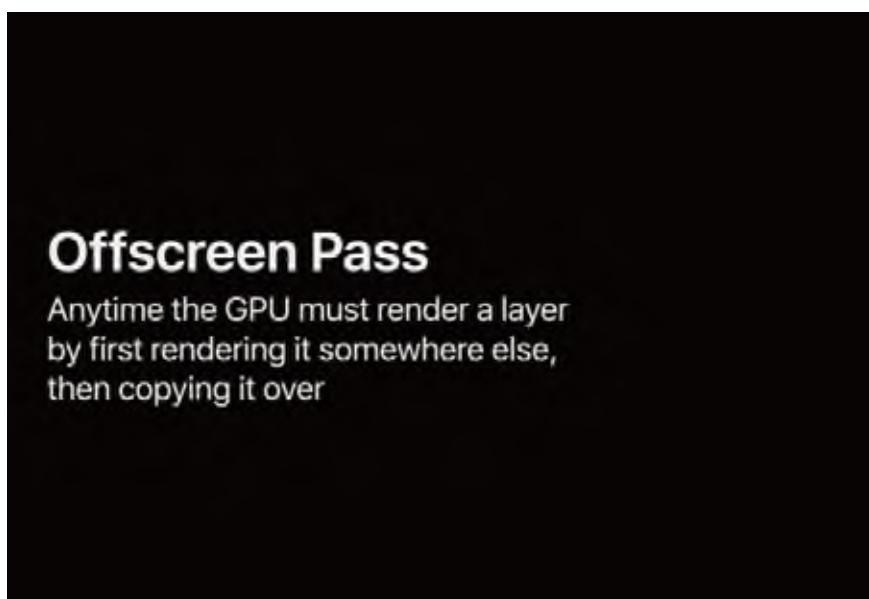
而最后的文本是在 CPU 上完成绘制的，GPU 会通过复制 CPU 绘制的文本图像来完成。完成上述流程后，帧已经准备好进行显示了。

需要注意在这个过程中我们不得不用离屏渲染来渲染阴影，导致渲染需要更长的时间。



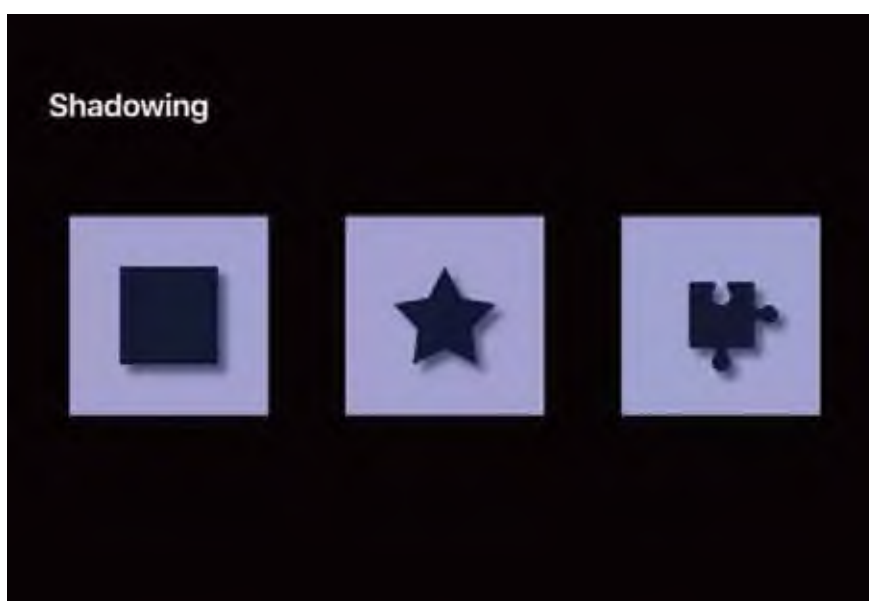
离屏渲染

离屏渲染通道指的是 GPU 必须先在其它地方开辟一块内存进行图层渲染，然后再将其复制回来。就阴影而言，它必须绘制图层，以确定最终形状。

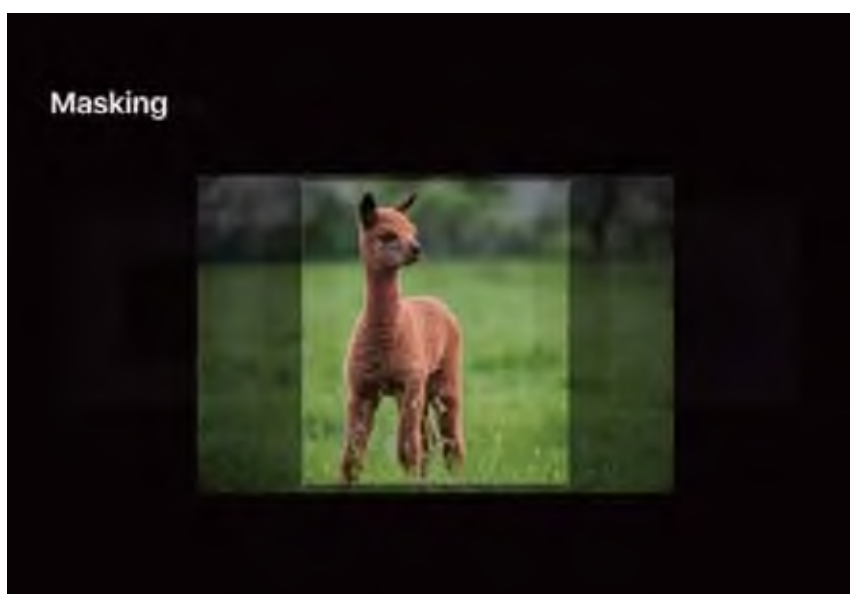


偶尔的离屏渲染对性能影响并不大，但离屏通道可能会积少成多，导致渲染出现卡顿。因此需要在 App 中监控并尽量避免。主要有四种主要类型的离屏通道可以优化：阴影、蒙版、圆角和毛玻璃。

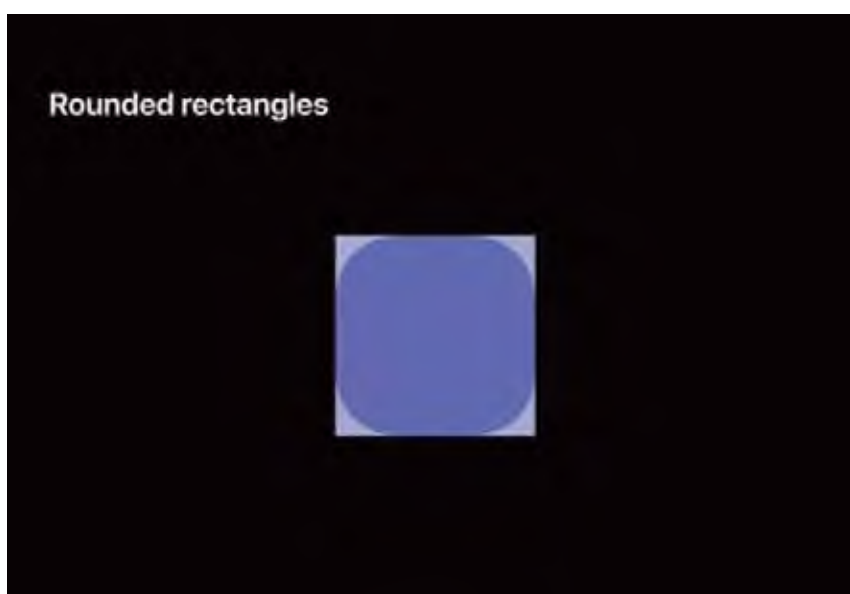
Shadow：比如在实例中，如果不先绘制附加到图形上面的阴影，GPU 就没有足够的信息来绘制阴影。



Mask: 当图层或图层树需要被遮蔽时, GPU 需要渲染被遮蔽的子树, 它也需要避免覆盖被遮蔽形状外的像素。因此它只会把最终需要显示的像素复制回最终纹理, 由于最终结果可能由多层渲染结果叠加, 所以必须要利用额外的内存空间对中间的渲染结果进行缓存, 因此系统会默认来触发离屏渲染, 这种离屏渲染可能会导致渲染了许多用户永远不会看到的像素。



CornerRadius: 由于 GPU 绘制时会先从根节点开始绘制, 所以如果根节点上设置了圆角, 并且设置了 maskToBounds 裁剪属性, 那就会需要一个额外的离屏渲染 buffer 用以缓存中间的裁剪结果, 并最终将圆角内的像素复制回来, 组透明度等属性都可能会触发离屏渲染。



iOS8 中开始支持 `UIBlurEffectView` 控件用以支持模糊化和鲜亮化，要应用这些效果，GPU 必须用离屏通道将内容复制到另一个纹理中，然后对其进行模糊、缩放叠加等操作并将最终结果复制回来。

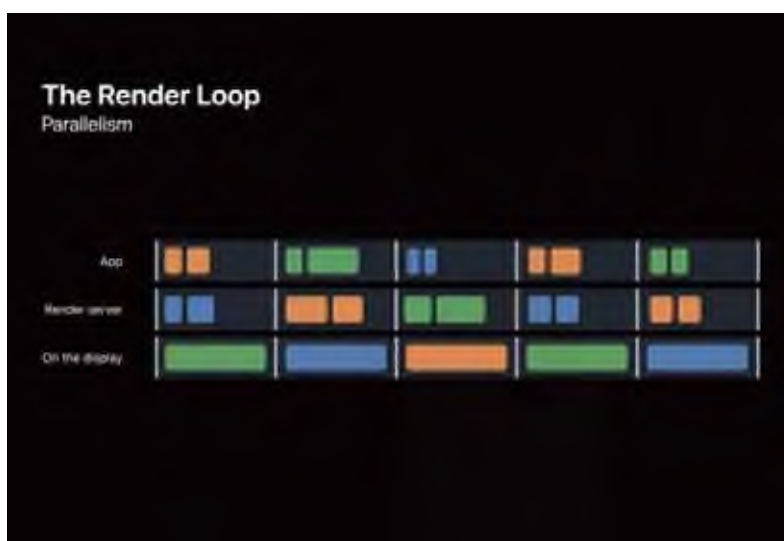


Display

Display 的过程实际上就是将帧缓存区中的内容交换到显示器上进行最终显示，这一过程我们参与不多。

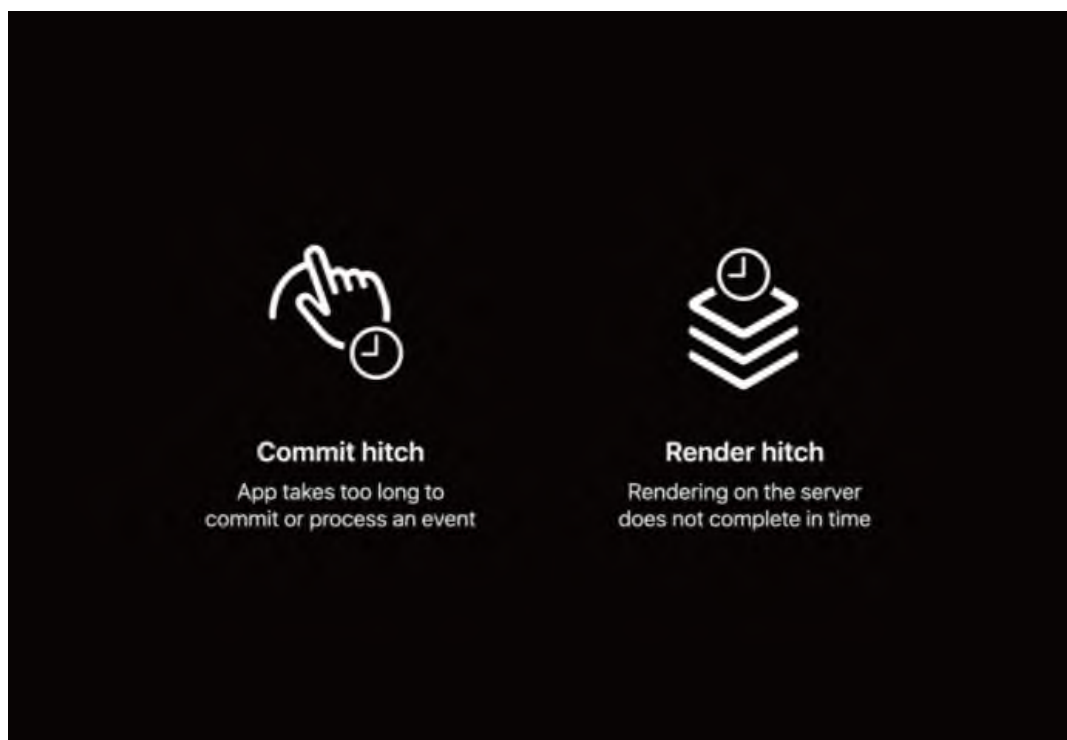
总结

为了达到目标帧速率并且保持低输入延迟，RenderLoop 的整个过程实际上是在每一帧中并行进行的，这样管线就成了并行的。在系统渲染前一帧的同时 CPU 可以准备一个新帧，所以每帧的截止期都很重要。



卡顿类型

上面已经描述了 RenderLoop 的整个工作流程，实际上主要是在 App 和 RenderServer 中进行，所以总共有两种主要类型的卡顿：提交卡顿（发生在 App 中），渲染卡顿（发生在 RenderServer 中）。



提交卡顿

概念

提交卡顿指的是 App 花费过长时间来处理或提交事件。

在提交中用了太长的时间而错过了截止期，所以在下一个 VSYNC 中 RenderServer 没有事情可以处理，必须等待下一个 VSYNC 到来后才能开始渲染。现在已经把帧传送的时间推迟了一帧，以毫秒计时这将是 iPhone(60hz) 或 iPad 上的 16.67 毫秒。这个延迟时间就被称为“卡顿时间(Hitch Time)”。如果提交工作花了更长的时间，比如通过了下一个 VSYNC 的起始时间，那么这一帧就晚了两帧或者说是 33.34 毫秒，在这 33.34 毫秒中用户都无法得到顺畅的滚动。



如何避免卡顿

保持视图的轻量

1. 为了保持视图的轻量尽可能地利用CALayer 上 GPU 加速的可用属性，如非必要需要避免使用 CPU 进行自定义绘制。
2. 若非必要情况下不要重写 drawRect 属性，因为其会开辟额外的内存空间进行 CPU 绘制，并且在 CPU 上绘制会耗费更多的时间主线程。针对于文本、图片等原本就在 CPU 上进行绘制的系统控件，我们可以尝试使用其更底层线程安全的 CoreGraphics 能力，比如 TextKit、CoreText 等搭配多线程异步绘制减轻主线程压力。
3. 尽量复用视图而不是频繁的添加或移除视图。
4. 如果要把某一视图从某一动画中移除，尽量使用 hidden 属性。
5. 对于 Prepare 阶段，当我们的 UIImage 容器视图的大小小于图片本身时，我们通常可以使用下采样技术 (downsampling) 来进行缩略图的创建以节省部分内存空间。

避免复杂布局

1. 减少代价过高且重复的布局，在需要更新布局时尽量只使用 setNeedsLayout。layoutIfNeeded 会消耗当前事务的生命周期也会造成卡顿，大多数时候你可以等到下一次 Runloop 执行时再更新你的布局。
2. 尝试使用最少的约束来完成布局。
3. 视图应该只能使自己或自己的子视图无效，而不能使其同级视图或父视图无效，避免递归布局。
4. 避免非必要的视图层级创建，复杂的视图层级会增加提交阶段的整体耗时

如何避免卡顿

合理多线程能力

- 1.学会利用 GCD 的多线程能力，充分利用 CPU 多核优势，提前在子线程进行布局等 UI 无关操作，避免主线程挂起(hang)。
- 2.避免主线程 IO 等磁盘相关操作。
- 3.而针对于常见的主线程解码操作，在 iOS15 之前，我们通常都是自己封装或是利用最常见的第三方库 SDWebImage 替我们在子线程进行解码操作。而在 iOS15 中，Apple 终于提供了官方的解决方案以解决该问题：UIImage 的 prepareThumbnailOfSize:completionHandler: 等新接口。
- 4.针对于必须在 CPU 上进行绘制的组件，尝试结合多线程使用异步绘制能力减轻主线程压力。

渲染卡顿

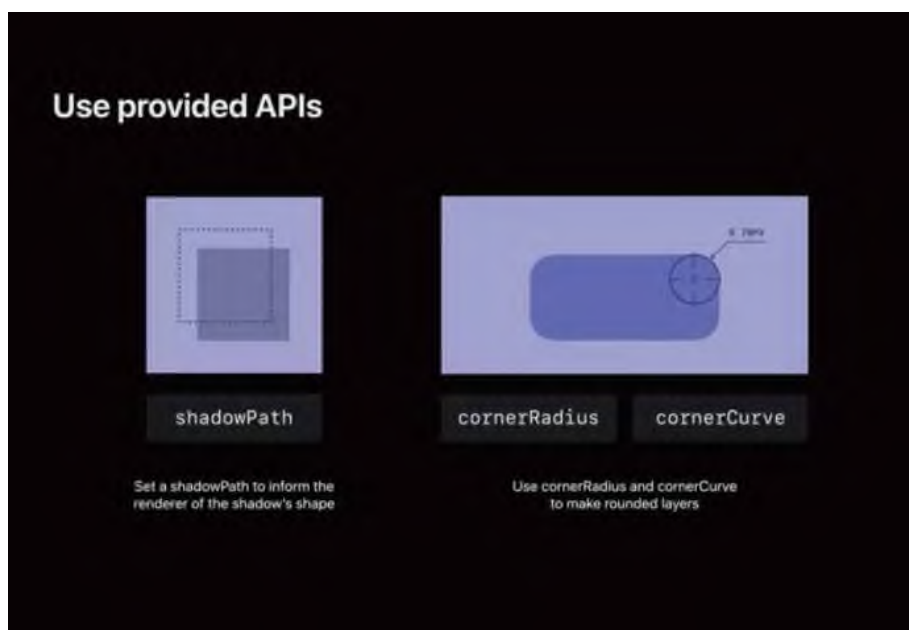
概念

渲染卡顿会在渲染服务器无法按时准备或者执行图层树时出现。这里显然 Execute 的时长超过了 VSYNC 的界限，因此这一帧无法按时准备好。绿色的画面比预期的晚了一帧于是有了 16 毫秒的卡顿

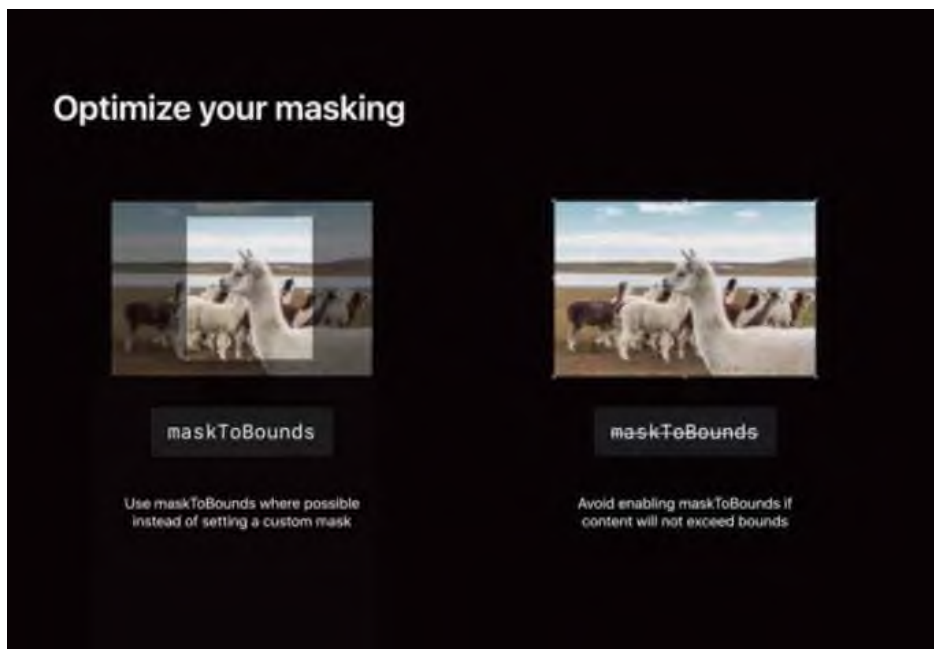


如何避免卡顿

准备阶段我们影响较少，通常主要影响在于执行阶段的离屏渲染。对于阴影来说，在设置阴影时，确保设置 shadowPath 以减少大量离屏通道。在圆化矩形时，使用 cornerRadius 和 cornerCurve 属性避免用蒙版或角内容来构成圆角矩形。



优化整个 App 的 Mask。使用 `masksToBounds` 遮蔽为矩形圆角矩形或椭圆形的性能比自定义蒙版图层好得多。重要的是用 Instruments 来对 App 进行分析并检查图层树以获得重要的技巧从而降低整体离屏计数。。



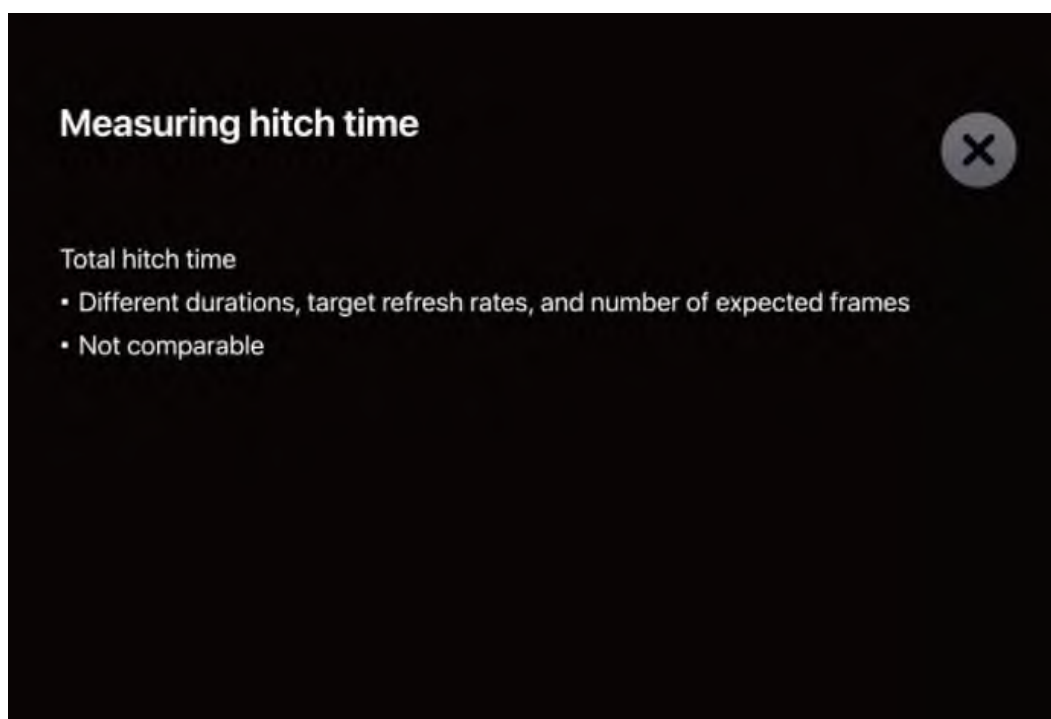
合理并谨慎的使用 `shouldRasterize` 属性，它会对一块图层进行光栅化操作并进行缓存。若针对于需要频繁刷新的图层使用该属性反而对性能有着负面影响。

尽量使用非透明的图层以尽量减少图层混合。

检测卡顿

当只着眼于一个卡顿或几个卡顿时，卡顿时间是很有用的，但在像在滚动、动画或者是转场等时长更长的事件时会变得很难处理。除非每次滚动或者动画用的都是完全相同的时间，这样就会有相同的帧数。并且 iOS 设备并不总是更新屏幕，如果没有事务发送到 RenderServer 上，新的一帧就不会被提交。通过测试来比较卡顿时间就更难了。所以 Apple 提供了一种叫“卡顿时间比(Hitch Time Ratio)”的指标来衡量一段时间内的卡顿情况。

卡顿时间比就是一个区间内的总卡顿时间除以它的持续时间。因为它标准化为总时间，我们就能在不同的实践中交叉比较。它是由每秒中的卡顿毫秒时间来测定的。所以代表着设备在每秒内出现卡顿的毫秒数。



一个实例如下，在一台 iPhone(60HZ) 上这是半秒的工作量，每一帧都在 VSYNC 到来前准备好了，所以用户看不到卡顿，卡顿时间为 0，卡顿时间比也为 0。

No hitches

Duration: 30 frames at 60 FPS = 0.5 s

Hitch time: 0 ms

Hitch time ratio: $0/0.5 = 0 \text{ ms/s}$



第二个例子如下，在该例中有时是在提交阶段的卡顿，有时是在 RenderServer 中造成了卡顿。将卡顿时间加起来结果就是 100.02 ms 半秒。我们就得到了每秒 200.04 ms 的卡顿时间比。

Many hitches

Duration: 30 frames at 60 FPS = 0.5 s

Hitch time: 100.02 ms

Hitch time ratio: $100.02/0.5 = 200.04 \text{ ms/s}$



以下是苹果建议的卡顿时间比目标。目标是 5 ms/s 以下的卡顿，是最不易被用户察觉到的。5~10 ms/s 的卡顿用户就会察觉到一些中断。超过 10 ms/s 就会严重影响用户体验。



总结

本篇主要讨论了 RenderLoop 以及新的一帧展现给用户的整个流程，并且着眼于什么是卡顿，以及它的两种类型：提交卡顿以及渲染卡顿。并最终定义了卡顿时间比用以测量当前 App 的卡顿程度和性能。相信大家对整个渲染循环和卡顿类型有了更清晰的认识，在日常编码中也可以尽量避免这些问题。

本篇主要介绍了一些原理相关的概念，那么具体的卡顿应该如何测量？下一篇将会通过实践结合 Instrument 的 AnimationHitches 能力分析 DXSDK 作为卡片层面在日常信息流的使用过程中在性能方面存在的一些问题，以及 DXSDK 上半年做的一些性能优化改进。

● 淘宝客户端诊断体系升级实践

- 作者： 伝逸
- 出品： 淘系技术

淘宝作为一个航母级的应用，每天都有数亿的用户在使用。保证客户端的稳定性是我们的首要目标。为此，我们也提出了5-15-60的目标。即问题告警时，5分钟响应，15分钟定位，60分钟恢复。

但是现有的排查体系并不能很好的达到这个目标，分析下来主要原因是：

监控阶段

- 1.通过Crash堆栈、异常信息进行聚合统计，不够精细准确，不够灵敏；
- 2.监控到异常后，端侧行为比较单一，只上报异常信息，无法提供更多有用数据；
- 3.手淘大部分问题都和线上变更有关，但是缺少对变更质量的监控。

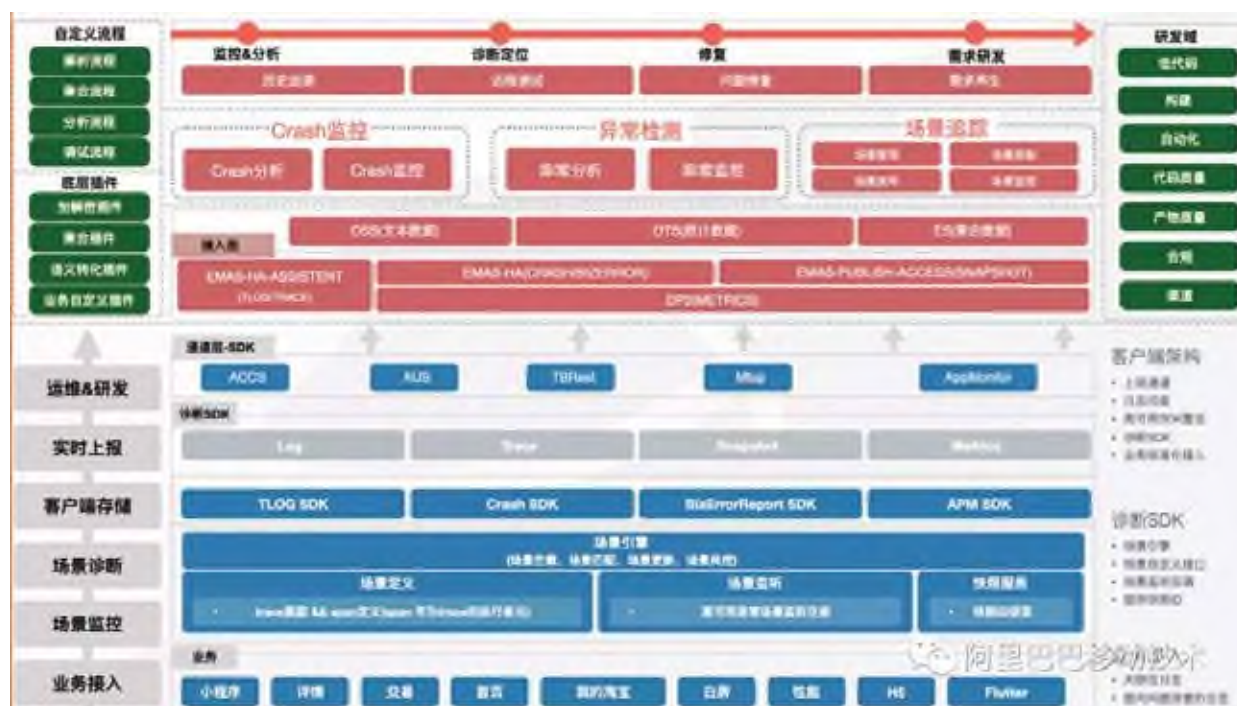
排查阶段

- 1.监控上报的异常信息不足，依赖日志进行问题排查；
- 2.Crash或异常时不会主动上传日志，需要手动捞取，用户不在线获取不到日志；
- 3.获取日志之后：
 - a.缺少分类，缺乏标准，日志杂乱，看不懂其他模块的日志；
 - b.缺少场景信息，无法完整的重现异常时用户的操作场景；
 - c.缺少整个生命周期相关的事件信息，无法掌握app的运行情况；
 - d.各个模块上下游的日志信息无法有效关联形成完整链路；
 - e.现有日志可视化工具功能较弱，无法提高排查效率；
- 4.问题排查靠人工分析，效率低下，相同问题缺少沉淀；
- 5.各个系统间的数据缺少关联，需要到多个平台获取数据。

诊断体系升级思路

针对以上现有问题，我们重新设计了整个无线运维排查诊断体系的架构。在新的架构中，我们引入了场景的概念。以往端上发生的异常都是一个个独立的事件，没有办法针对不同的异常做更精细的处理和数据收集。而引入

场景概念后，一个场景可能是一个异常和多种条件的组合，针对不同的场景可以做配置，使得异常信息的收集更加丰富，更加精准。



同时我们重新定义了端侧异常数据，主要包括标准的Log日志数据、记录调用链路的Trace全链路数据、运行时相关的Metric指标数据以及发生异常时的现场快照数据。平台侧可以利用异常数据进行监控和告警。也可以对这些数据进行可视化的解析，针对业务的差异，平台提供了插件化的能力来解析数据。利用这些语义化后的信息，平台可以进行初步的问题诊断。

所以接下来要实现的目标是：

- 1.实现端侧场景化的监控运维；
- 2.升级日志体系，实现LOG、TRACE、METRIC数据整合，提供更加丰富和精准的排查信息；
- 3.完成高可用体系数据整合，提供面向问题排查的标准化接口和平台；
- 4.插件化支撑平台赋能，业务自定义诊断插件，完成诊断体系平台间对接；
- 5.平台依据诊断信息，给出诊断结果，能做到自动化、智能化；
- 6.依据诊断结果给出解决方案或提出整改需求，形成从需求->研发->发布->监控->排查->诊断->修复->需求的闭环。

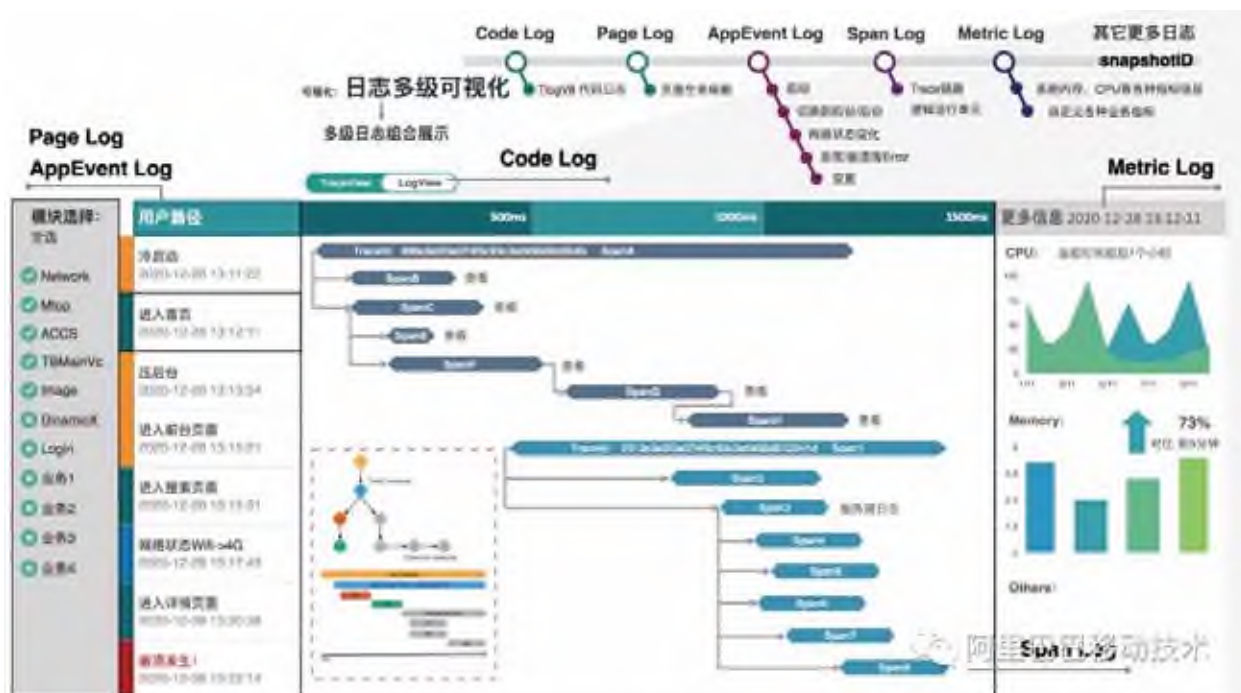
日志体系升级

目前分析运行日志还是端侧排查问题的主要手段，前面也提到我们的日志本身存在一些问题。所以我们第一步是对日志体系进行了升级。（在此之前我们完善了日志自身的基础能力，比如提升写入性能、提升日志压缩率、提升上传成功率、建立日志的数据大盘等等）



为了提高日志的排查效率，我们从日志内容着手，重新制定了端侧的标准日志协议。标准化的日志协议可以为我们在平台侧进行日志可视化建设、自动化日志分析提供帮助。我们从Log、Trace、Metric角度出发，依照手淘的实际情况将现有的日志分为以下几类：

- 1.CodeLog: 兼容原有的日志，这些日志相对杂乱；
- 2.PageLog: 记录端上页面跳转情况，排查问题时候可以从页面维度对日志进行划分；
- 3.EventLog: 记录端侧各种事件，比如前后台切换、网络状态、配置变更、异常、点击事件等等；
- 4.MetricLog: 记录端侧运行时的各种指标数据，比如内存、CPU、业务的各种指标数据；
- 5.SpanLog: 全链路日志数据。把各个独立的点串联起来，定义统一的性能度量、异常检测的标准。基于OpenTrace的方案和服务端打通，形成端到端的全链路排查机制。



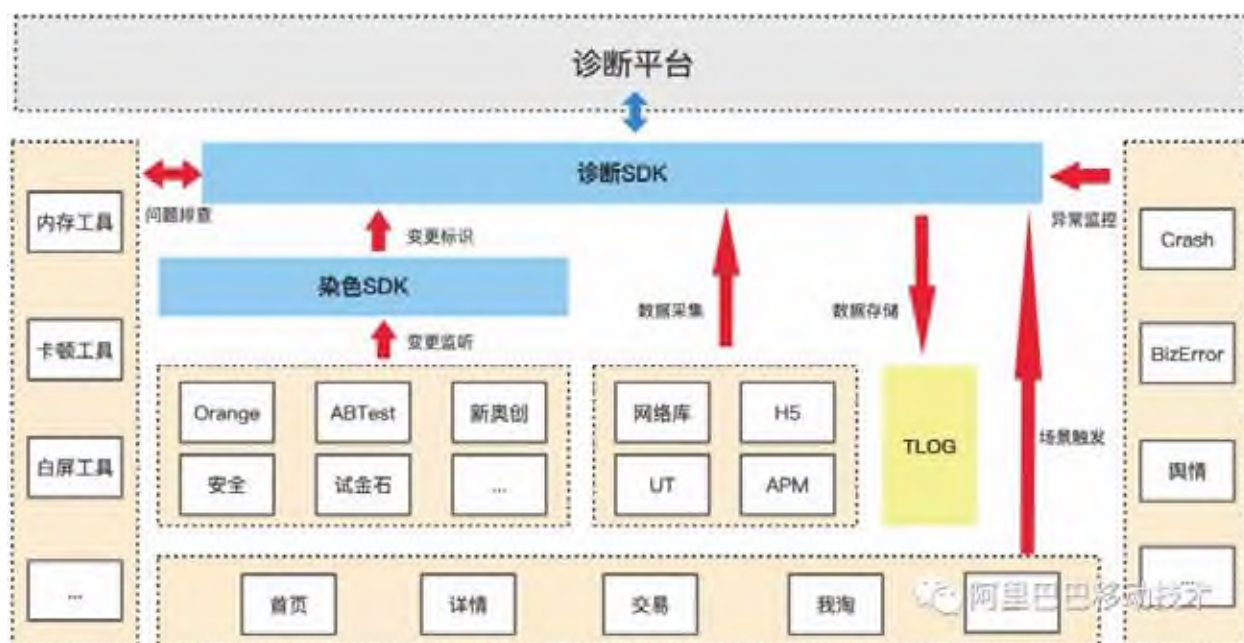
有了这些数据之后。在平台侧配合日志可视化平台，可以对端侧的行为进行快速回放。得益于日志的标准化，可以从平台侧对日志进行分析，快速展示存在异常的节点。

端侧诊断升级

客户端是整个诊断体系的源头，一切的异常数据，运行信息都是由客户端上的各种工具进行收集上报的。目前端侧主要工具有：

- 1.APM：收集端侧的各种运行、性能等信息，数据上报到服务端；
- 2.TLOG：收集端侧运行时的日志信息，日志文件存放在本地，需要时服务端下发指令进行捞取；
- 3.UT：端侧埋点工具，很多业务异常信息都会通过UT进行上报告警；
- 4.异常监控：这个以Crash SDK为代表，主要收集端侧的Crash信息，还有收集异常、用户舆情等相关的SDK；
- 5.排查工具：内存检测、卡顿检测、白屏检查等工具。这些没有直接归类在APM中，因为这些工具是采样运行，有的甚至默认是关闭状态。

可以看到端侧其实已经有不少成熟的工具在使用，但是在排查问题时候经常发现数据缺失等问题，主要原因在于一方面在平台侧这些数据比较分散，没有一个统一的接口进行查询；另一方面客户端没有对这些工具的数据做整合，发生异常时这些工具间少有交互，造成了数据缺失。

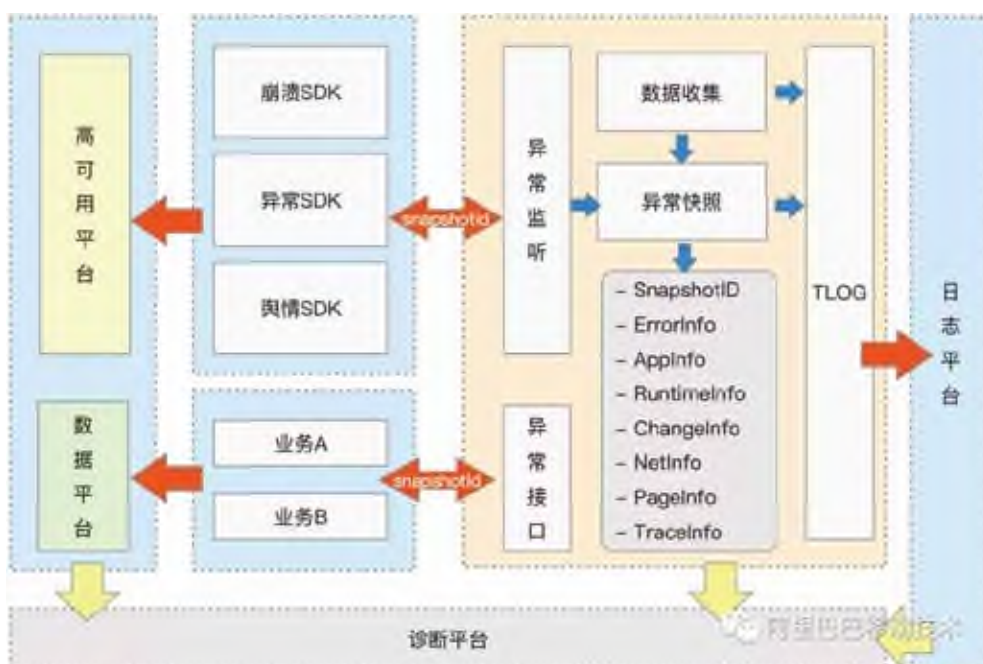


为了解决这些问题，我们在端侧引入了全新的诊断SDK和染色SDK。它们的主要功能有：

1. 对接现有工具，收集端侧运行数据。把这些数据按照标准的日志协议写入到TLOG日志中；
2. 监听端侧的变更信息，生成变更的染色标识；
3. 监听端侧异常，异常发生时生成快照信息（包括运行信息、变更信息），上报服务端；
4. 场景化诊断数据上报，根据服务端配置的规则，在特定场景进行数据收集和上报；
5. 支持定向诊断，根据服务端下发的配置，调用对应的排查工具进行数据收集；
6. 支持实时日志上传，针对特定用户和设备进行在线调试。

异常快照

端侧的异常主要包括Crash、业务异常、舆情等。当异常发生时，上报的数据格式、内容、通道、平台都不一样。若想要增加一些数据，端侧和相应平台都需要进行改造。所以我们对端侧异常监控相关的SDK进行了监听，对于业务提供了异常通知接口。当诊断SDK接收到异常时，会利用当前收集到的运行信息生成一份快照数据。每个快照数据会有一个唯一的snapshotID。我们只需要把这个ID传递给对应的SDK，这样对现有的SDK改动是最小的。



快照数据随着端侧能力的加强也会越来越丰富。收集到的快照信息会上传到诊断平台，平台之间可以利用 snapshotID 进行数据关联。对于诊断平台来说，可以根据快照信息、日志信息对异常进行分析，给出初步的诊断结果。

变更监控

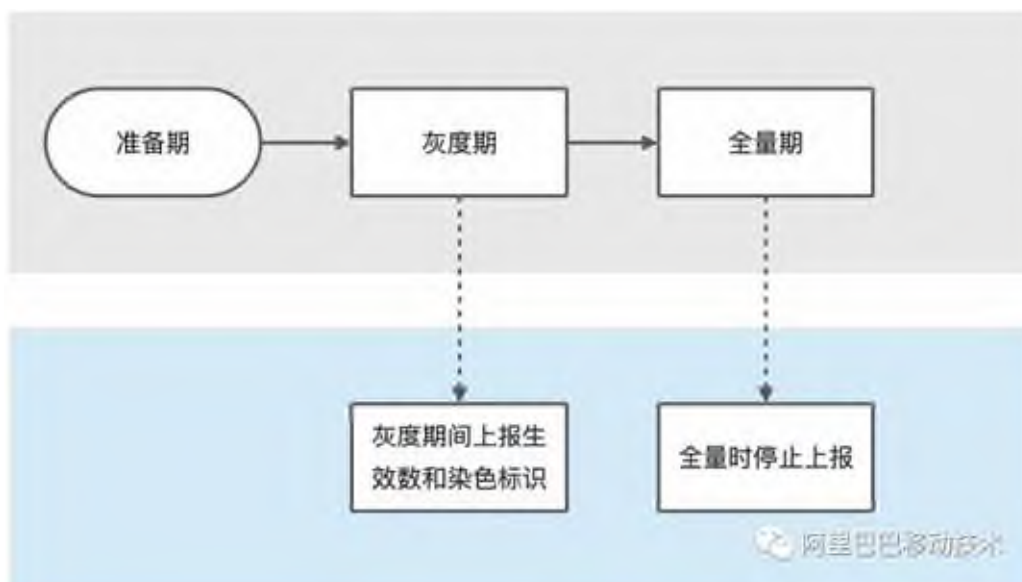
手淘中大部分的问题是因为线上变更导致的。现有监控排查体系并没有专门对变更进行监控，主要还是依赖异常数量、异常趋势进行告警。这就有一定的滞后性，导致很难在放量阶段快速的发现问题。同时对于变更发布也没有一个统一的管控标准。所以我们在端侧引入了染色SDK来收集变更数据，配合无线运维的变更诊断平台，对变更发布进行监控，做到可灰度、可观测、可回滚。



- 2.配置类型: 一个变更类型下可能有多个配置, 比如orange变更, 每个配置有一个namespace, 用来区分配置的类型;
- 3.版本信息: 代表了一个配置的一次具体的发布。不一定所有配置变更都有明确的版本信息, 可以把某个发布或配置的唯一标识作为版本信息。比如每个orange配置发布都有一个version, 每个ABTest发布都有一个publishID。

有了这些信息我们就可以给每一个变更生成一个唯一的染色标识。通过上报染色标识我们可以统计出变更的生效数; 通过在快照信息中携带染色标识, 可以计算有变更标识的crash率、舆情数量; 通过和高可用大盘数据进行比较监控变更的质量。业务也可以在网络请求中带上染色标识, 用于统计相关的接口是否存异常。

灰度定义



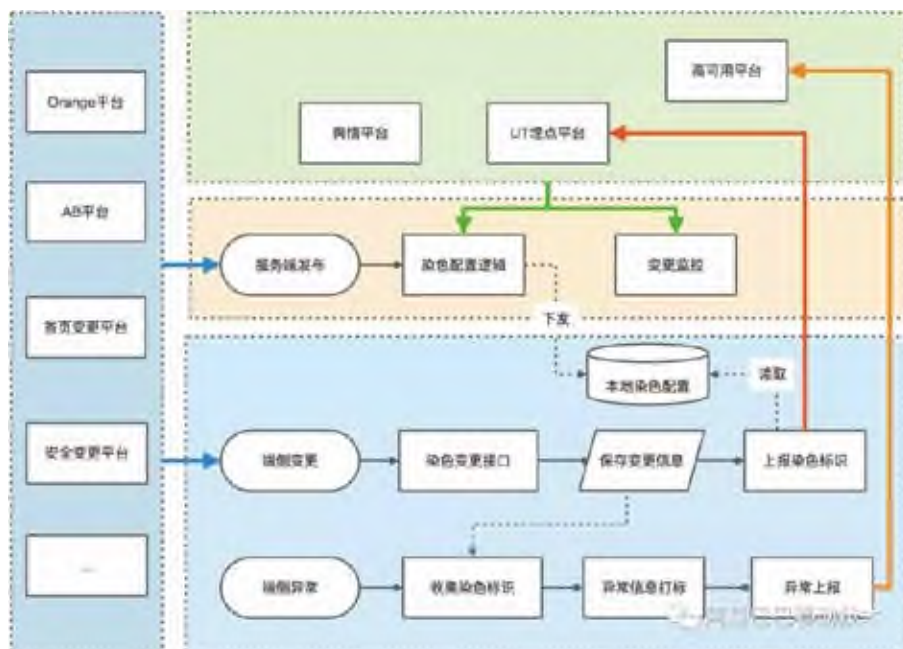
对于染色SDK我们是希望在灰度期间可观测, 提早发现问题, 不把变更导致的问题带到线上全量环境中, 所以我们把发布过程定位为三个阶段:

- 1.准备期: 准备变更数据, 预发验证、提交审批、线上发布。这个阶段可以确定发布变更的类型、配置、版本;
- 2.灰度期: 对部分用户下发变更配置。我们的染色监控也主要是在这个阶段运行, 主要为上报染色标识和在异常快照中携带染色标识。平台侧在这个阶段对数据进行处理, 生成灰度相关数据;
- 3.全量期: 当灰度达标之后, 进入全量期。这个时候向所有满足条件的用户推送配置。

数据上报控制

因为手淘用户量太大，变更全量之后如果还继续上报生效数据，对服务端的压力会很大。同时异常信息中如果继续打标意义也不大了。所以我们必须有一个机制来管控端侧的染色数据上报。

针对orange、ABTest这些通用的变更，我们进行了单独适配，可以根据实验号、配置namespace进行黑白名单控制、采样控制或发布状态来控制。但对于自定义变更来说，可控制的条件千差万别，如果要做到精细的控制就必须理解这个特定的变更。所以我们定义了一些通用的条件：灰度标识、采样率、过期时间来控制。



这些信息是通过配置文件的形式下发到端上。端侧无需关注这些条件设置的逻辑，而是在平台侧进行设置，平台侧对接发布平台、高可用平台，并根据上报数据进行决策。目前主要还是依据配置中的灰度标识+超时时间来决定是否上报。

发布门禁

端侧上报了生效数、异常染色等数据之后，服务端就可以根据这些数据来监控变更。根据相关Crash数量、舆情数量、灰度时间等来判定当前变更是否达到了全量发布的标准。



举个例子，按照现有的排查方式，当线上发生异常告警时，我们一般先通过上报的异常信息来排查，但受限于现有信息不全，往往还需要通过拉取TLOG来做进一步排查，然而TLOG捞取又依赖用户在线，这就导致整个排查定位时间非常长。



引入场景化概念之后，当平台检测到异常数量快要达到阈值的时候，可以自动生成一份场景规则配置，圈选一批用户下发到端上。当端上发生了相同异常的时候，场景引擎会负责收集和上报所需要的数据，这样当告警达到阈值时平台上就已经有足够的数据进行分析定位。

场景规则

场景引擎主要用来执行服务端下发的场景规则，而一个场景规则主要由三部分构成：

触发(Trigger)

可以是端上的一个行为或一个事件。相对于以前只有在Crash、业务异常时才上报数据，我们对异常触发时机进行了扩充。

- 1.崩溃异常
- 2.用户截屏反馈
- 3.网络异常（ mtop错误、network错误等）
- 4.页面异常（白屏、显示异常）
- 5.系统异常（内存占用过高、cpu占用过高、耗电过快、发热、卡顿）
- 6.业务异常(业务错误码、逻辑错误等)
- 7.启动（一般用于定向诊断）

条件(Condition)

条件判定是整个场景化上传的核心。增加了场景条件之后，我们可以从多个维度更精准的去划分异常类型。条件大致上可以分为：

- 1.基础条件：从设备信息、用户信息、版本信息等维度去进行匹配和筛选；
- 2.状态条件：主要包括网络状态、当前页面、内存水位等运行时的信息；
- 3.特定条件：不同场景需要判定的条件是不同的。比如发生Crash时，可以根据Exception类型、堆栈等信息进行匹配。而业务异常时可能根据错误码和网络错误来进行匹配。

行为(Action)

当端上某个规则被触发，并且满足设定的所有条件时，就会触发指定的行为，这样就可以根据不同的场景收集不同的数据。目前端侧主要行为有：

- 1.上传TLOG日志
- 2.上传快照信息
- 3.上传内存信息
- 4.协同其他排查工具根据下发的参数进行数据收集上报

场景下发



平台侧建立了一套新的场景管理平台，可以方便的配置各种场景和条件。并且也有标准的发布、审核、灰度流程。通过PUSH + PULL的方式，客户端可以及时的获取到场景规则。



平台和端侧也都支持定向下发配置的能力，通过指定一些基础的条件，可以针对不同设备，不同用户进行有针对性的场景下发。

数据流量管控

端上匹配了对应的场景规则后，就会上报各种异常数据。但是如果数据量过大，会对服务端存储产生较大压力。所以我们针对场景上报的数据进行了流量管控。



从排查问题的角度出发，对于同一个问题我们可能只需要几份相关的日志和数据。所以我们在创建一个场景时会指定一个数据上报的阈值。当平台收集到足够的数据之后，这个场景就会停止，并通知客户端规则下线。

同时端上为了保证用户不因为频繁上传诊断数据而对正常使用造成影响，端侧也有自己的限流策略。比如必须是wifi环境才可以上报、限制每天执行规则的数量、限制每天上传的数据量、设定数据的上传间隔时间等等。

自定义场景

目前我们的触发都是一些通用的场景，从高可用工具中获取数据上报。但是业务可能有一套自己的异常监控体系。所以我们也提供了相应的接口给业务来调用，利用我们场景下发、规则表达式执行、获取运行数据等能力，来帮助业务进行问题诊断。业务可以定义自己的触发时机、触发条件。后续我们也可以加入自定义行为的能力，让业务也可以根据场景来上报相应的数据。

定向诊断

端上目前除了TLOG，还有内存工具、性能工具、卡顿工具等其他一些排查工具。这些工具在排查特定问题时比较有用。但目前在线上都是通过配置采样开启或默认关闭的。而现有的配置还无法针对设备，用户进行定向下发。这就会导致排查问题时拿不到完整有效的信息。所以我们也是利用场景化定向下发能力，对现有工具进行了简单的改造，协同这些工具进行异常数据的收集和上报。

未来展望

目前端侧的诊断能力还在持续的建设中，我们也针对现有排查过程中存在的痛点进行迭代，比如实时日志、远程调试、完善全链路数据、异常数据等等。此外，我们也需要看到，端侧诊断不再是简单的堆砌工具，收集数据，我们未来还需要更有针对性、更精细化的去收集和处理数据。



而随着诊断数据的完善，下个挑战将是平台如何利用数据进行问题根因定位、分析问题影响面、对诊断结果进行沉淀。对于端侧而言，也不仅仅只定位在数据收集，还可以依赖平台的诊断知识沉淀，结合端智能的能力，在端侧实现问题诊断的能力。同时可以配合端侧的降级能力、动态修复等能力，在发生异常后实现自愈。

• 闲鱼的统一跨端 API 方案 —— Uni API

- 作者：叶遥
- 出品：淘系技术

本文将介绍跨端 API 解决方案应该解决什么问题，提供什么能力。

背景

随着 C 端流量红利的逐渐消失，端外投放已成为业务寻求增长的重要抓手之一。而不同 App 上存在不同应用场景、不同时期诞生的前端容器：

- 基于浏览器的 Webview 容器（h5）
- 基于客户端渲染衍生的 React Native、Weex 容器
- 基于自绘渲染的 flutter 容器
- 平台开放能力的小程序容器

业务在端外进行跨端时，就需要前端同学对投放目标环境逐一评估和适配。在初期阶段，前端侧通常通过同一业务针对不同目标环境维护多套实现的方式进行支持，这使得工作量成倍增加。效率提升空间下，催生了跨端方案。

问题定义

在跨端业务的前端代码中，通常存在大量的跨端 JS API 调用的重复逻辑：

```
1  if (isH5) {  
2    if (isXianyu) { // 闲鱼  
3      webXianyuToast();  
4    } else if (isTaobao) { // 淘宝  
5      webTaobaoToast();  
6    } else if (isAlipay) { // 支付宝  
7      webAlipayToast();  
8    }  
9    // ...  
10 } else if (isWeex)  
11 // ...
```

这些调用从开发到上线通常需要经历的路径：

- 和业务同学确定需要投放的目标环境。如 H5（支付宝、淘宝、天猫等）
- 向目标容器维护者询问 API 调用方式
- 在不同环境下调试、开发
- 测试同学使用不同机型在不同 App 上适配

每一步都是比较耗时的体力活。假想，能够满足可用性、易用性、丰富性、可扩展性，使得业务直接开发如下代码，正常测试后即可上线。跨端 API 解决方案应该解决什么问题，提供什么能力呢？

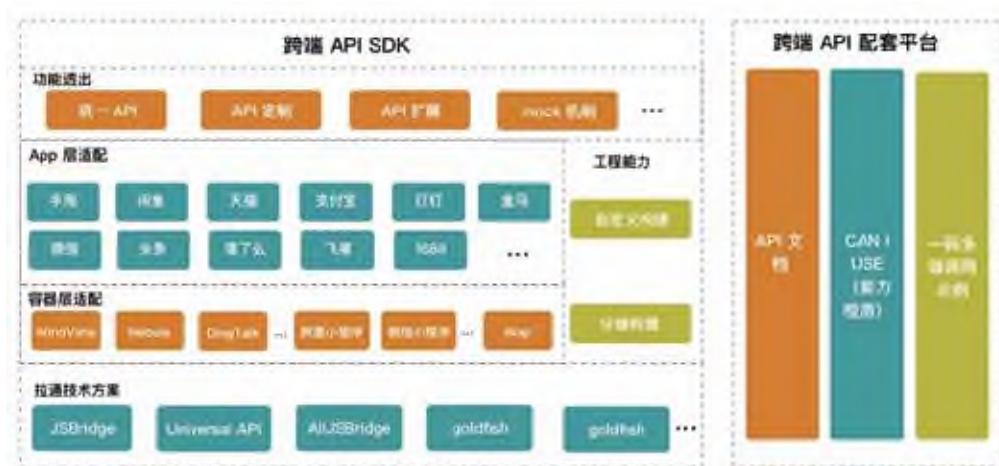
```
1 import toast from 'uni-toast'  
2  
3 toast()
```

1. 可用性：适配业务常投放的 Webview、小程序、Wap 等前端容器和 App。并最大化保障 API 能力使用，调用 APP 定制能力 -> 调用容器通用能力 -> Wap 环境模拟能力 -> 返回支持度信息，避免调用异常
2. 易用性：让不同环境的 API 调用有可靠、可快速调试支持度的统一入口
3. 丰富性：所支持的目标环境、API 集合足够多，满足绝大部分业务诉求
4. 可扩展性：随着业务的发展，能够支持更多 App、API；随着前端技术的发展，能够支持更多形态的容器

方案设计&实现

整体设计

在项目起步阶段，了解到不止是闲鱼，整个阿里经济体前端支撑跨端业务也有相同的问题。于是，跨端 API 项目站在经济体的视角，以共建的方式进行推进。针对上述问题的方案设计：





我们将跨端 API 整体方案定义为规范、SDK 和配套设施三个部分：

- 规范作为跨端 API 抹平标准，使得对上层业务只感知一套成为可能。
- SDK 遵循规范，对不同环境调用底层 API 进行抹平。通过自定义构建、分端构建等工程能力，透出可定制、可扩展的跨端 API 产物；
- 配套设施通过 API 文档、CANIUSE 工具提供快速检索能力，一码多端调用示例提供快速试用能力；

促成规范

跨端 API 规范的意义是：

向上为 SDK 层 API 设计提供参考标准，提高业务侧使用时的确定性、合理性；

向下为 Native 层 JSAPI 设计提供参考标准，减缓底层分化趋势；

规范能否普遍推广和保持生命力取决于 自身合理性 **和 **上下游认可度，为保障以上两点，邀请了经济体各现有跨端方案作者成立了跨端 API 规范小组，从以下四个方面制定了跨端 API 规范：





确定范围

什么样的 API 应该算作跨端 API。

跨端 API 应该具备 跨端属性和高频属性：跨端属性可通过各容器支持度客观反映，高频属性一方面通过各方案的调用统计作为依据，一方面通过跨端规范小组成员逐个投票判断

环境探针

跨端 API 核心在于根据不同环境调用相应实现，精确的环境感知尤为重要。

环境判断涉及到经济体内、外的标准容器、特殊容器，在环境探针规范中通过 容器识别协议、端识别协议、系统识别协议、识别次序约定 的组合机制覆盖了全场景；

标准 API 定义

标准 API 模型是真实 API 的 interface，也是所有 API 的骨骼。

通过标准 API 合集进行分析，API 之间的差异主要体现在：方法命名、调用方式、出入参结构、返回错误码 四个部分，这四部分加上 出入参扩展机制 就定义了标准 API 模型；

扩展机制

标准 API 合集未覆盖的场景，通过定制、扩展能力支持。

基于标准 API 扩展新的端容器实现，或者扩展一个全新的 API；

SDK实现

有了规范作为实现依据和指导之后，我们开始进行进行 SDK 实现。在整个过程中，主要面临了以下挑战：

【实现】巨大的维护工作量

55+ API 在 8 容器、30+ APP 上的适配和长期维护，且 API 和环境数量均无收敛趋势

【工程】多场景产物输出

多场景使用 -> 多形态产物。如日常业务开发期望的使用方式是 `window.uni.toast()`；跨端基础物料开发期望的是 `import toast from 'uni-toast'`。多场景的使用使得产物需要多形态输出

【工程】方案的可定制性

站在经济体视角，场景不只是面向闲鱼自身业务。业务侧场景通常只需要投放方案所支持容器环境中的一部分，使用整个方案会引入不必要的冗余。所以方案需要支持从全集中定义构建出子集的能力

应对上述挑战的关键解法是：分层按端适配 API。API 差异分布在容器层、APP 层，SDK 设计时，也按照分层进行抹平。容器层抹平了通用 API 差异，APP 层基于容器层进行定制。按端适配的设计带来了天然的扩展性，在支持一个新端时，只需实现对应的 API 适配集合，其余环境判断、挂载 API、多维度输出包的部分就交给工程和 uniapi-core 完成了。



开放方案扩展能力，降低中心化维护工作量。官方优先保障高频 APP、容器的 API 维护，当业务跨端投放目标环境未适配时，可通过共建的方式按照规范进行适配；另外，建立 APP 适配维护认领机制，使得维护多端适配具备长效性。

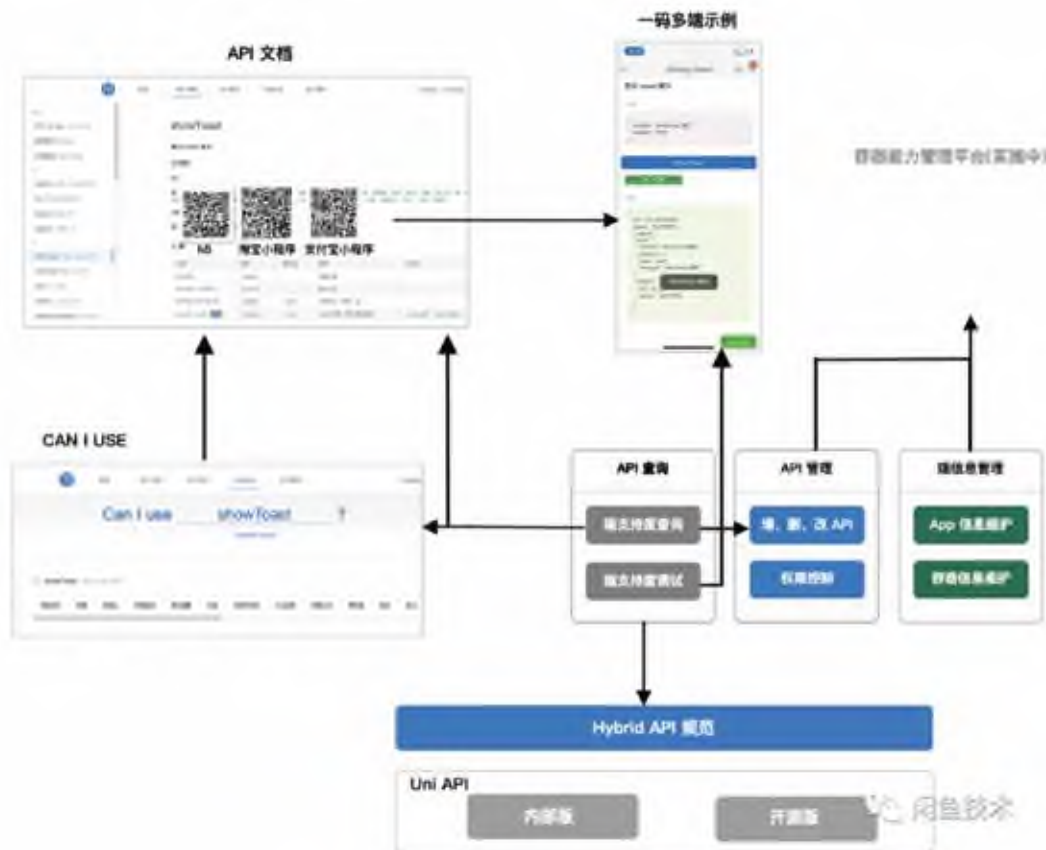
自定义构建能力。原子化 API 的设计 + 组合机制可以使整个方案具备了自定义构建能力。原子化 API：将指定容器、APP 上的 API 适配作为最小单元，进行无副作用的实现；组合机制：通过配置提取所需 API 及目标投放环境，以代码模板的方式自动组合 API 进行构建、发布；



拥有全环境的 API 原子化实现, 便能构建出任意支持度的跨端 API 方案。目前已输出数个 BU 级定制包。

配套设施建设

规范和 SDK 满足了可用性、丰富性、可扩展性诉求。在易用性, 跨端 API 提供复杂 API 的查询、调试能力, 建设了内部、开源站点: 文档(支持度信息细致到参数、APP 粒度)、CAN I USE 工具、一码多端调用示例等。



方案设计&实现

在接入跨端 API 方案后，跨端业务的工作流有了以下优化：

- 1.中心化的容器能力信息维护，使得产品、开发同学不再逐个环境询问能力，而是通过统一入口快速查询评估
- 2.统一多端适配 API 的 SDK，使得开发同学不再逐个环境拼凑、调试、开发 API，而是像标准 API 一样直接使用
- 3.统一维护 SDK，免去测试同学针对不同机型、不同环境下容器能力使用的工作量

从 SDK 1.0 release 开始，闲鱼会玩社区、端外分享承接页等业务就开始陆续进行试点和落地。此外，方案 SDK Uni API 已在阿里经济体内 10+ BU 应用，逐渐成为经济体前端开发的基础设施。

展望

抹平不是终点，上层适配应对分化永远是中间态方案，一套底层标准 API 才是最优解。「跨端 API 调用规范」为「容器 API 标准」提供来自上层使用侧的输入，使得新容器 API 在设计时能够有所参考，避免不必要的分化开源社区版本（[https://universal-api.js.org\[1\]](https://universal-api.js.org[1])）建设（由跨端 API 小组、Rax 等团队共同建设）。目前 Uni API 开源版本已支持阿里系、微信、字节系、百度等小程序和 h5 容器，预期后续持续扩展 API 和支持容器等丰富度

所以在端技术仍在日益发展的背景下，下一代的跨端方案，是由底层的 Fuchsia OS、HarmonyOS 等多端操作系统统一，还是依旧通过上层适配实现呢，你怎么认为呢？

Flutter企业级应用开发在闲鱼的落地

- 作者：闲鱼技术
- 出品：淘系技术

本文回顾闲鱼引进应用Flutter的过程，并介绍在使用过程中，我们遇到的一系列问题与困难。

闲鱼在 2017 年便引入了 Flutter，是国内第一个引进Flutter的团队。

当时的 Flutter 还远未成熟，行业内也没有把 Flutter 放入已有工程体系进行开发的先例。

这两年，Flutter 也逐渐在其他企业里落地，但同时也不断有质疑的声音发出。甚至有传言表示“闲鱼的新业务已经放弃 Flutter”、“相信闲鱼遇到了很大的难题”……

那么，作为 Flutter 先驱和探路者，闲鱼在过去几年的摸索过程中遇到过什么阻力？踩过什么“坑”？对于相关技术开源的态度是怎样的？对于以上的一些传言又是怎么看待的呢？

针对这些疑问，闲鱼代表团队分享了一些开发者背后不为人知的小故事。希望能让大家从背后人的维度去看闲鱼团队，理解他们的初心，以及这个过程中遇到的挑战、挣扎、付出的努力、得到的教训。

在新技术落地过程中，遇到过什么阻力，有什么避免”踩坑“的经验？

闲鱼从2017 年开始接触Flutter，最初是一个自上而下的尝试。

一方面，从创新团队的角度来看，需要以战养战，为团队储备更多的人才；
另一方面，希望切实解决当时团队人难招、产能低的问题。

调研过程

在前期调研过程中，最初是希望将Flutter 作为闲鱼国际化版本的主要技术选型；由于后续业务上的决策，导致该项目夭折，因此希望将Flutter 引入主App 中进行探索。

在尝试之初，从组织、配套工具，到架构侧都没有准备得特别好，项目落地面临很多困难。在团队抽调的突击小组中，有大量的研发人员非常不看好这项技术，大部分研发人员并不认同Flutter，没人知道这件事情该怎么做。

当然，从2017 年开始，闲鱼在多个技术方向都进行了类似的尝试，实践证明，创新的确比想象难得多。

时间来到2020 年，回首过去尝试的多个方向，目前只有Flutter 得到了阶段性的胜利，并作为闲鱼的技术品牌为大家广泛熟知。这里面所经历的艰难险阻只有当时参与过从0 到1 过程的研发人员才能体会。

由于Flutter 落地过程的艰辛，其间也有不少同事离开了闲鱼，幸好后续有更多希望这件事情成功的研发人员加入，大家一起努力至今，才有了今天看到的成果。我们也非常感谢在此过程中付出过努力的每一位同事，不管今天他们是否还在闲鱼，是每一位同事的努力让美好终于发生。

从Flutter 落地的过程中总结出以下观点，供大家参考。

从布道者的角度来看

需解决以下两方面的问题：向上，需要得到更多决策者的支持；向下，需要减少开发者进入的成本。

关于决策者链路

Flutter 在闲鱼落地的过程中没有上层的阻力，因为开始是一个自上而下的决策，但在后续的推广和落地过程中，从更大的组织视角去看，比如构建AliFlutter 组织、构建Flutter China 组织，这些是一个自下而上的行为。

在这个过程中，闲鱼选择的是“农村包围城市”的理念。

先从社区出发，发现同路人，找到不同公司、不同业务的价值，再回归内部，影响决策者。前期不管是构建集团Flutter 的兴趣小组，还是正式有了AliFlutter 的虚拟组织，都是找到同路人，探讨和争论其技术价值的过程，这个过程可以请各团队的关键决策者躬身入局。通过讨论又能帮助团队进行更深入的思考，并为Flutter 提供一个阶段性的明确定位——面向研发效能提升，构建新的研发模式，快速支持小前台业务落地。将该目标与阿里巴巴集团大的研发效能目标进行连接和推动，即可顺利地推进该项目。

关于开发者成本

首先是自己团队的开发者成本。在任何新技术落地的前期，都有一定的成本，需要度过这个阶段才能得到收益。因此，在开始阶段，必须尽量降低落地成本。

在项目之初，最好以小规模团队的形式进行尝试，同时新技术在前期架构落地中需要重点考虑过渡阶段的效率问题，不要断崖式地迁移。闲鱼在初期没有充分考虑这个问题，导致工程侧开始的架构并不合理。后来，闲鱼重

点围绕混合架构，以Flutter 开发小组与原生开发小组的协作模式进行设计和落地后，该问题得到了有效解决。

例如，前期闲鱼就重点解决了原生项目无法独立打包的问题，极大地解决了当时研发效率的阻塞问题。混合栈（Flutter Boost）的重新设计，也解决了当时由于Flutter和原生生命周期不一致带来的一系列复杂问题，保证了开发调试的稳定性。如果团队仍有余力，可以考虑对开源社区进行一些力所能及的输出。闲鱼先通过发布文章，再通过部分开源项目以及内部开源的形式，给Flutter 社区和阿里巴巴集团提供了一些帮助。闲鱼相信“日拱一卒，功不唐捐”，虽然依然还有很多问题需要解决，但是随着更多开发者的进入，大量的问题终究会被解决。

在闲鱼对开源项目的态度是什么？未来会把大量文章中提到的技术开源出来吗？

闲鱼在2018 年完成Flutter 主链路的部分改造后，发现在落地过程中有很多问题，而且Flutter 社区并没有给出很好的解决方案。

从帮助Flutter 社区其他开发者的角度出发，闲鱼内部开始讨论开源事宜，这是闲鱼第一次尝试开源，所有人都没有经验。大家开开心心地希望能将自己做的一些事情同社区进行交流。

当然，作为第一次开源，很多事情做得也非常糟糕，糟糕的文档、糟糕的Demo、没有测试用例、问题维护不及时、没有好的社区共建方式，于是引发了社区开发者的大量抱怨。

从自己真实的感受来说，要维护好社区项目，需要付出大量的精力和时间，这些甚至需要不少同事全职去做，这对一个业务团队来说有些勉为其难。

对开源的反思是要敬畏开源，敬畏社区，在反馈中不断地成长，完善自身项目。例如，Flutter Boost 前期的设计在兼容性上侵入性太强，导致后续的升级成本较高。在未来的3.0 版本中，闲鱼会主动解决这一问题，减少关键的几个类的继承和重写，简化概念，降低接入成本。

目前，这个阶段会重点关注已经开源的项目，尤其是Flutter Boost 这种在阿里巴巴集团内外都属于开发者刚需的项目，需要持续地维护，也非常欢迎更多的社区开发者参与进来。

关于文章中提到的其他Flutter 相关技术，闲鱼会尽量考虑使用Demo 等形式将部分原理输出，帮助开发者更好地理解文章及其背后的思想。

一方面保证了闲鱼对Flutter 社区的启发和帮助，另一方面也缓解了闲鱼目前面临的无法保证全职人员从事开源工作的问题。但无论如何，闲鱼希望通过文章及代码，持续地帮助Flutter 社区，尽我们的一份绵薄之力。

也许我们做得并不完美，甚至不够优秀，但随着闲鱼有更多优秀研发人员的加入，在创新和分享文化方面进行不断的自我革新，相信总有一天能伴随这项技术成为真正的国民级别应用的贡献者。

听说闲鱼放弃了Flutter，作为当事人，闲鱼的态度是什么？

在各类社交平台匿名区和公众号中，都有人阶段性地、煞有介事地散播闲鱼放弃了Flutter 以及未来不会再投入的消息，作为当事人，我先说结论——这些都是谣言！

《Flutter企业级开发应用实战：闲鱼技术演进与创新》一书的出版是对这些谣言最有力的反驳，也感谢出版社帮我们辟谣。

移动互联网发展的十年，对于跨平台技术来说，从最早的PhoneGap 到ReactNative、Weex，再到目前的Flutter，形形色色的技术不断迭代和演进。

从当前阶段来看，Flutter 具备跨平台技术的先驱——QT 的大部分优势，且配套了更现代的语言、更合理的分层架构，并有Google 公司支持，是新一代跨平台技术的集大成者。

在下一个十年，面向万物互联的新操作系统，如华为鸿蒙及Google Fuchsia 的出现，使得跨平台技术的价值进一步凸显。

因此，学习这项开源技术对未来的开发者来说是一个不错的选择。我们无法预测Flutter 是否一定为下一个十年的主流技术，但这套代码和其背后的思想一定能给未来的开发者以启发，就像我们的前辈用QT 开发了KDE，KDE 又提供了一个排版引擎KHTML，而苹果公司Fork 了这个项目，从而产生了WebKit，而Google 又Fork 了WebKit，从而产生了Blink。

这些令人尊敬的公司和开发者站在巨人的肩膀上不断地努力，最终打造了令人耳熟能详的技术，从而改变世界。

今天，Flutter 代码中有很多概念也存在浏览器内核的一些影子。作为客户端的从业者，希望开发者都能客观地评价这些技术，根据实际情况进行学习并将其落地。

对于外界传出的类似闲鱼放弃Flutter 的谣言，大部分时间我们选择付之一笑。

作为管理者，我不太希望团队过于关注外部的评价和过度回应不实传闻，希望闲鱼的开发者可以做正确的事情，并保持长期的耐心。

未来的三年，闲鱼更希望探索先进的研发模式，帮助整个阿里巴巴集团在商业竞争中保持技术的先进性和成本优势。当然，我们希望这种研发模式是基于我们现有的技术沉淀基础之上的，因此会依然持续地投入Flutter 之中。

同时，作为一个创新团队，闲鱼也一定会持续地关注行业的最新动态，未来也一定会像在Flutter 领域一样，在其他领域有所建树。

• 一个 Crash 引发的血案

- 作者：九渝
- 出品：淘系技术

背景

手淘 9.13.0 版本上线后，突然出现了一个首现 crash，废话不多说上堆栈

```
java.lang.IllegalStateException: The specified child already has a parent. You must call removeView() on the child's parent first.
    at android.view.ViewGroup.addViewInner(ViewGroup.java:5328)
    at android.view.ViewGroup.addView(ViewGroup.java:5155)
    at android.view.ViewGroup.addView(ViewGroup.java:5095)
    at android.view.ViewGroup.addView(ViewGroup.java:5068)
    at android.support.v4.app.FragmentManagerImpl.moveToState(FragmentManagerImpl.java:1434)
    at android.support.v4.app.FragmentManagerImpl.moveToFragmentToExpectedState(FragmentManagerImpl.java:1759)
    at android.support.v4.app.FragmentManagerImpl.moveToState(FragmentManagerImpl.java:1627)
    at android.support.v4.app.BackStackRecord.executeOps(BackStackRecord.java:797)
    at android.support.v4.app.FragmentManagerImpl.executeOps(FragmentManagerImpl.java:2596)
    at android.support.v4.app.FragmentManagerImpl.executeOpsTogether(FragmentManagerImpl.java:2383)
    at android.support.v4.app.FragmentManagerImpl.removeRedundantOperationsAndExecute(FragmentManagerImpl.java:2338)
    at android.support.v4.app.FragmentManagerImpl.executePendingActions(FragmentManagerImpl.java:2245)
    at android.support.v4.app.FragmentManagerImpl.executePendingTransactions(FragmentManagerImpl.java:784)
    at com.taobao.taop.navigation.TBFragmentTabHost.onAttachedToWindow(TBFragmentTabHost.java:409)
    at android.view.View.dispatchAttachedToWindow(View.java:19759)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3596)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewRootImpl.performTraversals(ViewRootImpl.java:2229)
    at android.view.ViewRootImpl.doTraversal(ViewRootImpl.java:1879)
    at android.view.ViewRootImpl$TraversalRunnable.run(ViewRootImpl.java:6178)
    at android.view.Choreographer$CallbackRecord.run(Choreographer.java:1057)
    at android.view.Choreographer.doCallbacks(Choreographer.java:875)
    at android.view.Choreographer.doFrame(Choreographer.java:776)
    at android.view.Choreographer$FrameDisplayEventReceiver.run(Choreographer.java:1042)
    at android.os.Handler.handleCallback(Handler.java:900)
    at android.os.Handler.dispatchMessage(Handler.java:103)
    at android.os.Looper.loop(Looper.java:219)
    at android.app.ActivityThread.main(ActivityThread.java:8347)
    at java.lang.reflect.Method.invoke(Native Method)
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:513)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1055)
Back traces end.
```

看起来很普通的一个 crash，报在手淘首页启动的瞬间，但是后续引发了一个很奇怪的问题，这篇文章就对这个问题的记录做一个详细记录。

第一次排查

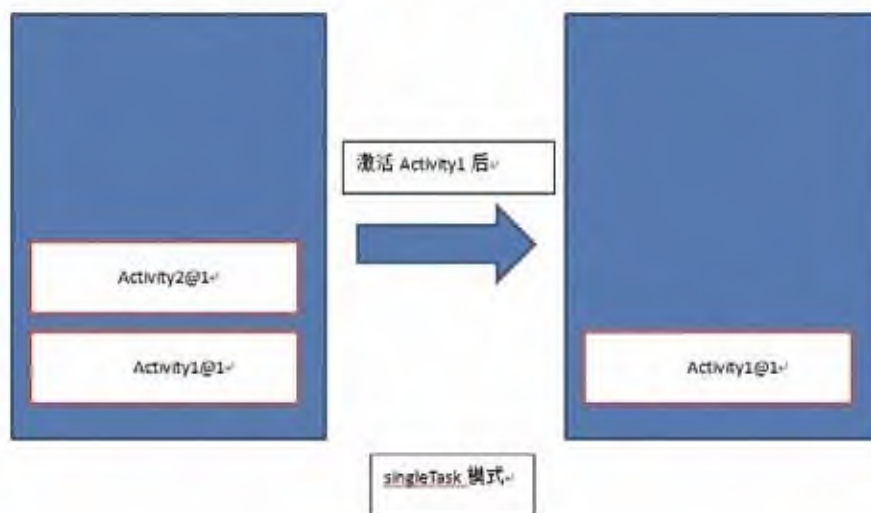
发现问题后，找到日志做了初步分析，想看到到底是什么场景下会引发这个 crash，先看日志。

```
E2020-10-12 00:11:01.009:23471750217714apnActivityLifecycle->onActivityCreated->SearchDoorActivity
E2020-10-12 00:11:01.110:23631750217790apnActivityLifecycle->onActivityResumed->SearchDoorActivity
E2020-10-12 00:11:01.576:24831750217790apnActivityLifecycle->onActivityStopped->MainSearchResultActivity
E2020-10-12 00:11:01.584:26141750217714apnActivityLifecycle->onActivityDestroyed->MainSearchResultActivity
E2020-10-12 00:11:04.862:143721750217790apnActivityLifecycle->onActivityPaused->SearchDoorActivity
E2020-10-12 00:11:04.895:145821750217714apnActivityLifecycle->onActivityCreated->DrawerActivity
E2020-10-12 00:11:04.948:146581750217790apnActivityLifecycle->onActivityCreated->DrawerActivity
E2020-10-12 00:11:04.951:146581750217790apnActivityLifecycle->onActivityResumed->DrawerActivity
E2020-10-12 00:11:05.204:147481750217790apnActivityLifecycle->onActivityStopped->SearchDoorActivity
E2020-10-12 00:11:05.237:147521750217714apnActivityLifecycle->onActivityDestroyed->SearchDoorActivity
E2020-10-12 00:11:09.085:151681750217714apnActivityLifecycle->onActivityPaused->DrawerActivity
E2020-10-12 00:11:09.085:151681750217790apnActivityLifecycle->onActivityCreated->TbMainActivity
E2020-10-12 00:11:09.087:151971750217790apnActivityLifecycle->onActivityCreated->TbMainActivity
E2020-10-12 00:11:09.152:152741750217714apnActivityLifecycle->onActivityResumed->TbMainActivity
E2020-10-12 00:11:09.753:161881750217790apnActivityLifecycle->onActivityStopped->DrawerActivity
E2020-10-12 00:11:09.757:161311750217714apnActivityLifecycle->onActivityDestroyed->DrawerActivity
E2020-10-12 00:11:17.544:171803750217790apnActivityLifecycle->onActivityPaused->TbMainActivity
E2020-10-12 00:11:17.588:171921750217790apnActivityLifecycle->onActivityCreated->Welcome
E2020-10-12 00:11:17.707:172191750217714apnActivityLifecycle->onActivityCreated->Welcome
E2020-10-12 00:11:17.708:172221750217790apnActivityLifecycle->onActivityResumed->Welcome
E2020-10-12 00:11:17.741:172261750217790apnActivityLifecycle->onActivityPaused->Welcome
E2020-10-12 00:11:17.772:172351750217790apnActivityLifecycle->onActivityCreated->TbMainActivity
E2020-10-12 00:11:17.784:172341750217790apnActivityLifecycle->onActivityCreated->TbMainActivity
E2020-10-12 00:11:17.785:172561750217714apnActivityLifecycle->onActivityResumed->TbMainActivity
E2020-10-12 00:11:17.831:173071750217790apnActivityLifecycle->onActivityResumed->TbMainActivity
at android.view.ViewGroup.addViewInner(ViewGroup, Layout$1$8) at android.view.ViewGroup$
```

上面日志中打印出来了出问题之前的手淘生命周期日志和最后一行发生 crash 时候的堆栈，其中 Welcome 是手淘的欢迎页面，TbMainActivity 是手淘首页（其他的 activity 不是重点，暂时不关注），发现了一个非常奇怪的事情，TbMainActivity 执行了两次 onCreate 生命周期，但是两次生命周期之间却没有执行 onDestroy，而 TbMainActivity 的启动模式是 singleTask 的。作为 Android 开发同学来说，Activity 的启动模式再熟悉不过了，当启动一个 Activity 是 singleTask 模式的时候，如果这个 Activity 已经存在栈中，那么行为应该是把该 Activity 之上所有 Activity 都直接清除掉，然后直接复用 Activity，借用网上一张图说明：

singleTask

与 singleTop 的区别是 singleTask 会扫描整个任务栈并制定策略。上效果图：



看到这个日志的第一反应是线上存在一种特殊情况，手淘会在栈中同时存在两个首页！！

但是抛出结论的时候让否了，因为首页的启动模式很清楚，不可能在一个栈里面存在两个实例，大家都说这里的两个实例应该是在两个栈里面，我也觉得应该是这样，就暂时放弃了这个方向的排查，把排查的重心放在拉起两个首页 crash，而不是为什么拉起了两个首页。

以为自己破案了

连续拉取好几份日志观察之后发现一个共同点，每次发生 crash 之前都是在首页的时候拉起了 welcome 页面，判断是在首页点击 back 按键的时候快速 finish 自己，瞬间用户又点击了 icon 打开了手淘，导致第一个首页还没有 finish 然后又拉起了一个新的，于是在首页 finish 的时候做了保护逻辑，进行了第一次灰度。

过几天观察效果，发现 crash 数量大幅减少，从占比 20% 多降到 6%，终于松了一口气，以为自己破案了。

但是一个周六突然接到了告警电话，这个 crash 疯长，过几分钟之后又回落到正常水位，看来还没有完全破案。事情还得查。

这次真的破案了

上次灰度失败，我反思了一下，还是要回到 crash 本身。

回到 crash 本身，堆栈中我们自己的代码出现在图中红框中的代码：

```
java.lang.IllegalStateException: The specified child already has a parent. You must call removeView() on the child's parent first.
    at android.view.ViewGroup.addViewInner(ViewGroup.java:5326)
    at android.view.ViewGroup.addView(ViewGroup.java:5155)
    at android.view.ViewGroup.addView(ViewGroup.java:5095)
    at android.view.ViewGroup.addView(ViewGroup.java:5069)
    at android.support.v4.app.FragmentManagerImpl.moveToState(FragmentManagerImpl.java:1434)
    at android.support.v4.app.FragmentManagerImpl.moveToState(FragmentManagerImpl.java:1759)
    at android.support.v4.app.FragmentManagerImpl.moveToState(FragmentManagerImpl.java:1827)
    at android.support.v4.app.BackStackRecord.executeOps(BackStackRecord.java:797)
    at android.support.v4.app.FragmentManagerImpl.executeOps(FragmentManagerImpl.java:2590)
    at android.support.v4.app.FragmentManagerImpl.executeOpsTogether(FragmentManagerImpl.java:2383)
    at android.support.v4.app.FragmentManagerImpl.removeRedundantOperationsAndExecute(FragmentManagerImpl.java:2338)
    at android.support.v4.app.FragmentManagerImpl.execPendingActions(FragmentManagerImpl.java:2145)
    at android.support.v4.app.FragmentManagerImpl.executePendingTransactions(FragmentManagerImpl.java:1784)
    at com.taobao.taobao.navigation.TabFragmentTabHost.onAttachedToWindow(TabFragmentTabHost.java:409)
    at android.view.View.dispatchAttachedToWindow(View.java:19759)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3596)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewGroup.dispatchAttachedToWindow(ViewGroup.java:3603)
    at android.view.ViewRootImpl.performTraversals(ViewRootImpl.java:2229)
    at android.view.ViewRootImpl.doTraverse(ViewRootImpl.java:1879)
    at android.view.ViewRootImpl$TraverseRunnable.run(ViewRootImpl.java:8179)
    at android.view.Choreographer$CallbackRecord.run(Choreographer.java:1857)
    at android.view.Choreographer.doCallbacks(Choreographer.java:875)
    at android.view.Choreographer.doFrame(Choreographer.java:776)
    at android.view.Choreographer$FrameDisplayEventReceiver.run(Choreographer.java:1842)
    at android.os.Handler.handleCallback(Handler.java:906)
    at android.os.Handler.dispatchMessage(Handler.java:103)
    at android.os.Looper.loop(Looper.java:219)
    at android.app.ActivityThread.main(ActivityThread.java:8347)
    at java.lang.reflect.Method.invoke(Native Method)
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:511)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1055)
Back Traces end.
```

FragmentTabHost 是手淘首页中的 Fragment 管理器，但是这块代码确实好几个版本都没有变动了。代码中看到 TabHost 在 onAttachedToWindow 函数中，最终调用了 FragmentManager 类中的 executePendingTransactions 方法提交了 Fragment 的切换请求。中间代码就是一路按照堆栈调用，调用到 FragmentManagerImpl 的 moveToState 方法。系统源码如下：

```
1 void moveToState(Fragment f, int newState, int transit, int transitionStyle,
2     boolean keepActive) {
3     .....
4     case Fragment.CREATED:
5         if (newState > Fragment.CREATED) {
6             if (DEBUG) Log.v(TAG, "moveto ACTIVITY_CREATED: " + f);
7             if (!f.mFromLayout) {
8                 ViewGroup container = null;
9                 if (f.mContainerId != 0) {
10                     container = (ViewGroup)mContainer.onFindViewById(f.mContainerId);
11                     if (container == null && !f.mRestored) {
12                         throwException(new IllegalArgumentException(
13                             "No view found for id 0x"
14                             + Integer.toHexString(f.mContainerId) + " ("
15                             + f.getResources().getResourceName(f.mContainerId)
16                             + ") for fragment " + f));
17                     }
18                 }
19                 f.mContainer = container;
20                 f.mView = f.performCreateView(f.getLayoutInflater(
21                     f.mSavedFragmentState), container, f.mSavedFragmentState);
22                 if (f.mView != null) {
23                     f.mInnerview = f.mView;
24                     if (Build.VERSION.SDK_INT >= 11) {
25                         ViewCompat.setSaveFromParentEnabled(f.mView, false);
26                     } else {
27                         f.mView = NoSaveStateFrameLayout.wrap(f.mView);
28                     }
29                     if (container != null) {
30                         Animation anim = loadAnimation(f, transit, true,
31                             transitionStyle);
32                         if (anim != null) {
33                             setHwLayerAnimListenerIfAlpha(f.mView, anim);
34                             f.mView.startAnimation(anim);
35                         }
36                         //在这里调用了 addView
37                         container.addView(f.mView);
38                     }
39                     if (f.mHidden) f.mView.setVisibility(View.GONE);
40                     f.onViewCreated(f.mView, f.mSavedFragmentState);
41                 } else {
42                     f.mInnerview = null;
43                 }
44             }
45         }
46     }
```

```
44         }
45
46         f.performActivityCreated(f.mSavedFragmentState);
47         if (f.mView != null) {
48             f.restoreViewState(f.mSavedFragmentState);
49         }
50         f.mSavedFragmentState = null;
51     }
52     .....
53 }
54
```

在上面代码中，调用了 container 的 addView 方法，container 是一个 ViewGroup，而这个 addView 的参数 f.View 是在 performCreateView 中进行赋值的，系统源码如下：

```
1 View performCreateView(LayoutInflater inflater, ViewGroup container,
2     Bundle savedInstanceState) {
3     if (mChildFragmentManager != null) {
4         mChildFragmentManager.noteStateNotSaved();
5     }
6     return onCreateView(inflater, container, savedInstanceState);
7 }
```

而 performCreateView 函数最终调用到 Fragment 的 onCreateView 中，这个函数是首页 Fragment 自己实现的，找负责同学要来代码继续看。

```
1 public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
2     .....
3     if (mRootView == null) {
4         View view = workflow.onCreateView(inflater, container, savedInstanceState);
5         return view;
6     }
7     .....
8     return mRootView;
9 }
10
```

看起来逻辑是首页 Fragment 做了自己的缓存处理，如果 mRootView 为 null，则从 coldStartUpWorkflow 中调用 onCreateView 创建一个 view 并返回，如果 mRootView 不为 null，则去掉 view 的父节点，一点毛病没有。那问题只能出现在 coldStartUpWorkflow 中创建的 view 可能有问题。


```
1  @Override
2      public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
3          super.onCreateView(inflater, container, savedInstanceState);
4          return create(inflater,container);
5      }
6
7      private View create(LayoutInflater inflater, ViewGroup container){
8          View rootView = null;
9          .....
10
11          rootView = RevisionReuseViewController.getInstance().getHomePageView(inflater, container);
12
13          .....
14          return rootView;
15      }
```

发现这个函数名字虽然叫 onCreateView，但是其中的 view 对象却是通过一个 get 开头的函数获取来的，不出意外问题肯定是在这里了。

```
1  public class RevisionReuseViewController {
2
3      private WeakReference<View> homePageViewWR;
4
5      private RevisionReuseViewController(){}
6      private static class SingleHandler {
7          private static RevisionReuseViewController instance = new RevisionReuseViewController();
8      }
9
10     public static RevisionReuseViewController getInstance() {
11         return RevisionReuseViewController.SingleHandler.instance;
12     }
13
14     public View getHomePageView(LayoutInflater inflater, ViewGroup container) {
15         View view = homePageViewWR == null ? null : homePageViewWR.get();
16         if (view == null) {
17             homePageViewWR = new WeakReference<>(inflater.inflate(R.layout.activity_homepage, container, false));
18         }
19         return homePageViewWR.get();
20     }
21
22 }
```

RevisionReuseViewController 这个单例里面竟然自己做了缓存逻辑，而触发这个逻辑的时候并没有把缓存的 view 的父节点 remove 掉。

确定了问题所在，马上第二次灰度，果不其然 crash 不再出现了。

但是还遗留下来一个问题，为什么会有两个首页呢？

两个首页

crash 解决了，但是更棘手的问题还在后面，crash 只是把这个问题暴露出来了，也就是说两个首页有可能引起这个 crash，但是不是所有两个首页都引发这个 crash，我们还不知道这个问题在线上到底有多么严重，会不会出现很多意想不到的问题。

可能很多同学会说，出现两个 singleTask 的 activity，他们有可能在两个栈里面。但是第一次灰度的时候我们已经确定了，他们是在一个栈里面，这回线索彻底断了。

天无绝人之路，在首页同学把修复代码集成进手淘的前一天，突然接到了一个钉钉消息，一个同学告诉我，他们有个 crash，要我确认，我看到堆栈之后发现跟前面解决的问题堆栈一致，就告诉他，这个版本修复，低概率的话暂时不用管。

但是他告诉我，我这边必现这个 crash。

按耐住激动的心，让他告诉我复现路径，果然必现。要来代码看到了一个崩快的原因：原来有同学在代码里面调用 startActivityForResult 启动了首页。

想了一下也没有什么问题，因为 startActivity 内部调用就是 startActivityForResult 来实现的，但是直觉告诉我就是这里有问题，于是写个 demo 验证一下我的想法。

demo 很简单，两个 activity，第一个 activity 是 singleTask 模式，其中一个按钮，点击正常调用到第二个 activity，第二个 activity 是标准模式，其中两个按钮，第一个按钮调用 startActivity 拉起第一个 activity，第二个按钮通过 startActivityForResult 拉起第一个 activity，启动 requestCode 传递的是 1。具体代码就不贴了，很简单一个 demo。当我点击第二个 activity 的第一个按钮的时候，看到栈里面只剩下一个 activity，符合预期。但是当我点击第二个按钮的时候，神奇的一幕出现了。很明显一个 TaskRecord 里面有两个 MainActivity，而且我确认这个 MainActivity 是 singleTask 的。奇怪的是 startActivity 里面调用的也是 startActivityForResult，只是传递的值是 -1，难道真的是这个值决定了这个行为么？

真相大白

看源码吧（以下代码来自 Android 9.0）

```
1  public void startActivity(Intent intent) {
2      this.startActivity(intent, null);
3  }
4  @Override
5  public void startActivity(Intent intent, @Nullable Bundle options) {
6      if (options != null) {
7          startActivityForResult(intent, -1, options);
8      } else {
9          // Note we want to go through this call for compatibility with
10         // applications that may have overridden the method.
11         startActivityForResult(intent, -1);
12     }
13 }
14
15 public void startActivityForResult(@RequiresPermission Intent intent, int requestCode) {
16     startActivityForResult(intent, requestCode, null);
17 }
18 public void startActivityForResult(@RequiresPermission Intent intent, int requestCode,
19     @Nullable Bundle options) {
20     if (mParent == null) {
21         options = transferSpringboardActivityOptions(options);
22         Instrumentation.ActivityResult ar =
23             mInstrumentation.execStartActivity(
24                 this, mMainThread.getApplicationThread(), mToken, this,
25                 intent, requestCode, options);
26         if (ar != null) {
27             mMainThread.sendActivityResult(
28                 mToken, mEmbeddedID, requestCode, ar.getResultCode(),
29                 ar.getResultData());
30         }
31         if (requestCode >= 0) {
32             // If this start is requesting a result, we can avoid making
33             // the activity visible until the result is received. Setting
34             // this code during onCreate(Bundle savedInstanceState) or onResume() will keep the
35             // activity hidden during this time, to avoid flickering.
36             // This can only be done when a result is requested because
37             // that guarantees we will get information back when the
38             // activity is finished, no matter what happens to it.
39             mStartedActivity = true;
```

```
40     }
41
42     cancelInputsAndStartExitTransition(options);
43     // TODO Consider clearing/flushing other event sources and events for child windows.
44 } else {
45     if (options != null) {
46         mParent.startActivityFromChild(this, intent, requestCode, options);
47     } else {
48         // Note we want to go through this method for compatibility with
49         // existing applications that may have overridden it.
50         mParent.startActivityFromChild(this, intent, requestCode);
51     }
52 }
53 }
54 可以看到 startActivity 调用到 startActivityForResult，startActivityForResult 调用了 Instrumentation 中的 execStartActivity 执行后续流程（下面 else 中的 mParent.startActivityFromChild 的流程最终也调用到了此处）。
55 public ActivityResult execStartActivity(
56     Context who, IBinder contextThread, IBinder token, Activity target,
57     Intent intent, int requestCode, Bundle options) {
58     .....
59     try {
60         intent.migrateExtraStreamToClipData();
61         intent.prepareToLeaveProcess(who);
62         int result = ActivityTaskManager.getService()
63             .startActivity(whoThread, who.getBasePackageName(), intent,
64                 intent.resolveTypeIfNeeded(who.getContentResolver()),
65                 token, target != null ? target.mEmbeddedID : null,
66                 requestCode, 0, null, options);
67         checkStartActivityResult(result, intent);
68     } catch (RemoteException e) {
69         throw new RuntimeException("Failure from system", e);
70     }
71     return null;
72 }
```

可以看到startActivity 调用到 startActivityForResult，startActivityForResult 调用了 Instrumentation 中的 execStartActivity 执行后续流程（下面 else 中的 mParent.startActivityFromChild 的流程最终也调用到了此处）。

```
1 public ActivityResult execStartActivity(  
2     Context who, IBinder contextThread, IBinder token, Activity target,  
3     Intent intent, int requestCode, Bundle options) {  
4     .....  
5     try {  
6         intent.migrateExtraStreamToClipData();  
7         intent.prepareToLeaveProcess(who);  
8         int result = ActivityTaskManager.getService()  
9             .startActivity(whoThread, who.getBasePackageName(), intent,  
10                intent.resolveTypeIfNeeded(who.getContentResolver()),  
11                token, target != null ? target.mEmbeddedID : null,  
12                requestCode, 0, null, options);  
13         checkStartActivityResult(result, intent);  
14     } catch (RemoteException e) {  
15         throw new RuntimeException("Failure from system", e);  
16     }  
17     return null;  
18 }
```

里面删除一些无关逻辑，最终调用到 ActivityTaskManagerService 中的 startActivity 函数继续流程。

```
1 int startActivityAsUser(IApplicationThread caller, String callingPackage,  
2     Intent intent, String resolvedType, IBinder resultTo, String resultWho, int requestCode,  
3     int startFlags, ProfilerInfo profilerInfo, Bundle bOptions, int userId,  
4     boolean validateIncomingUser) {  
5     enforceNotIsolatedCaller("startActivityAsUser");  
6  
7     userId = getActivityStartController().checkTargetUser(userId, validateIncomingUser,  
8         Binder.getCallingPid(), Binder.getCallingUid(), "startActivityAsUser");  
9  
10    // TODO: Switch to user app stacks here.  
11    return getActivityStartController().obtainStarter(intent, "startActivityAsUser")  
12        .setCaller(caller)  
13        .setCallingPackage(callingPackage)  
14        .setResolvedType(resolvedType)  
15        .setResultTo(resultTo)  
16        .setResultWho(resultWho)  
17        .setRequestCode(requestCode)  
18        .setStartFlags(startFlags)  
19        .setProfilerInfo(profilerInfo)  
20        .setActivityOptions(bOptions)  
21        .setMayWait(userId)
```

```
22         .execute();
23
24     }
```

ActivityTaskManagerService 的 startActivity 方法只是转换了一下控制权，最终调用到 startActivityAsUser 中，把控制权交给 ActivityStarter，传递参数，并最终执行 execute 函数执行后续流程。

```
1  int execute() {
2      try {
3          if (mRequest.mayWait) {
4              return startActivityMayWait(mRequest.caller, mRequest.callingUid,
5                  mRequest.callingPackage, mRequest.realCallingPid, mRequest.realCallingUid,
6                  mRequest.intent, mRequest.resolvedType,
7                  mRequest.voiceSession, mRequest.voiceInteractor, mRequest.resultTo,
8                  mRequest.resultWho, mRequest.requestCode, mRequest.startFlags,
9                  mRequest.profilerInfo, mRequest.waitResult, mRequest.globalConfig,
10                 mRequest.activityOptions, mRequest.ignoreTargetSecurity, mRequest.userId,
11                 mRequest.inTask, mRequest.reason,
12                 mRequest.allowPendingRemoteAnimationRegistryLookup,
13                 mRequest.originatingPendingIntent, mRequest.allowBackgroundActivityStart);
14          } else {
15              return startActivity(mRequest.caller, mRequest.intent, mRequest.ephemeralIntent,
16                  mRequest.resolvedType, mRequest.activityInfo, mRequest.resolveInfo,
17                  mRequest.voiceSession, mRequest.voiceInteractor, mRequest.resultTo,
18                  mRequest.resultWho, mRequest.requestCode, mRequest.callingPid,
19                  mRequest.callingUid, mRequest.callingPackage, mRequest.realCallingPid,
20                  mRequest.realCallingUid, mRequest.startFlags, mRequest.activityOptions,
21                  mRequest.ignoreTargetSecurity, mRequest.componentSpecified,
22                  mRequest.outActivity, mRequest.inTask, mRequest.reason,
23                  mRequest.allowPendingRemoteAnimationRegistryLookup,
24                  mRequest.originatingPendingIntent, mRequest.allowBackgroundActivityStart);
25          }
26      } finally {
27          onExecutionComplete();
28      }
29  }
```

由于参数重 mRequest.mayWait 的值为 true，所以调用到 startActivityMayWait。

```
1 private int startActivityMayWait(IApplicationThread caller, int callingUid,
2     String callingPackage, int requestRealCallingPid, int requestRealCallingUid,
3     Intent intent, String resolvedType, IVoiceInteractionSession voiceSession,
4     IVoiceInteractor voiceInteractor, IBinder resultTo, String resultWho, int requestCode,
5     int startFlags, ProfilerInfo profilerInfo, WaitResult outResult,
6     Configuration globalConfig, SafeActivityOptions options, boolean ignoreTargetSecurity,
7     int userId, TaskRecord inTask, String reason,
8     boolean allowPendingRemoteAnimationRegistryLookup,
9     PendingIntentRecord originatingPendingIntent, boolean allowBackgroundActivityStart) {
10     .....
11     int res = startActivity(caller, intent, ephemeralIntent, resolvedType, aInfo, rInfo,
12         voiceSession, voiceInteractor, resultTo, resultWho, requestCode, callingPid,
13         callingUid, callingPackage, realCallingPid, realCallingUid, startFlags, options,
14         ignoreTargetSecurity, componentSpecified, outRecord, inTask, reason,
15         allowPendingRemoteAnimationRegistryLookup, originatingPendingIntent,
16         allowBackgroundActivityStart);
17     .....
18     return res;
19 }
```

去掉一些多余逻辑，留下关键路径，看到转移到 startActivity 继续执行，

```
1 private int startActivity(IApplicationThread caller, Intent intent, Intent ephemeralIntent,
2     String resolvedType, ActivityInfo aInfo, ResolveInfo rInfo,
3     IVoiceInteractionSession voiceSession, IVoiceInteractor voiceInteractor,
4     IBinder resultTo, String resultWho, int requestCode, int callingPid, int callingUid,
5     String callingPackage, int realCallingPid, int realCallingUid, int startFlags,
6     SafeActivityOptions options, boolean ignoreTargetSecurity, boolean componentSpecified,
7     ActivityRecord[] outActivity, TaskRecord inTask, String reason,
8     boolean allowPendingRemoteAnimationRegistryLookup,
9     PendingIntentRecord originatingPendingIntent, boolean allowBackgroundActivityStart) {
10     .....
11     mLastStartActivityResult = startActivity(caller, intent, ephemeralIntent, resolvedType,
12         aInfo, rInfo, voiceSession, voiceInteractor, resultTo, resultWho, requestCode,
13         callingPid, callingUid, callingPackage, realCallingPid, realCallingUid, startFlags,
14         options, ignoreTargetSecurity, componentSpecified, mLastStartActivityResult,
15         inTask, allowPendingRemoteAnimationRegistryLookup, originatingPendingIntent,
16         allowBackgroundActivityStart);
17     .....
18     return getExternalResult(mLastStartActivityResult);
19 }
20
21
```

中间一顿操作，最终调用到下面

```
1 private int startActivity(IApplicationThread caller, Intent intent, Intent ephemeralIntent,
2     String resolvedType, ActivityInfo aInfo, ResolveInfo rInfo,
3     IVoiceInteractionSession voiceSession, IVoiceInteractor voiceInteractor,
4     IBinder resultTo, String resultWho, int requestCode, int callingPid, int callingUid,
5     String callingPackage, int realCallingPid, int realCallingUid, int startFlags,
6     SafeActivityOptions options,
7     boolean ignoreTargetSecurity, boolean componentSpecified, ActivityRecord[] outActivity,
8     TaskRecord inTask, boolean allowPendingRemoteAnimationRegistryLookup,
9     PendingIntentRecord originatingPendingIntent, boolean allowBackgroundActivityStart) {
10     int result = START_CANCELED;
11     final ActivityStack startedActivityStack;
12     try {
13         mService.mWindowManager.deferSurfaceLayout();
14         result = startActivityUnchecked(r, sourceRecord, voiceSession, voiceInteractor,
15             startFlags, doResume, options, inTask, outActivity, restrictedBgActivity);
16     } finally {
17     }
18 }
19
20 postStartActivityProcessing(r, result, startedActivityStack);
21
22 return result;
23 }
24
```

终于调用到了 startActivityUnchecked 这个地方看下里面都做了什么。

```
1 private int startActivityUnchecked(final ActivityRecord r, ActivityRecord sourceRecord,
2     IVoiceInteractionSession voiceSession, IVoiceInteractor voiceInteractor,
3     int startFlags, boolean doResume, ActivityOptions options, TaskRecord inTask,
4     ActivityRecord[] outActivity, boolean restrictedBgActivity) {
5     setInitialState(r, options, inTask, doResume, startFlags, sourceRecord, voiceSession,
6         voiceInteractor, restrictedBgActivity);
7     .....
8     // If the activity being launched is the same as the one currently at the top, then
9     // we need to check if it should only be launched once.
10    final ActivityStack topStack = mRootActivityContainer.getTopDisplayFocusedStack();
11    final ActivityRecord topFocused = topStack.getTopActivity();
12    final ActivityRecord top = topStack.topRunningNonDelayedActivityLocked(mNotTop);
13    final boolean dontStart = top != null && mStartActivity.resultTo == null
14        && top.mActivityComponent.equals(mStartActivity.mActivityComponent)
```

```
15         && top.mUserId == mStartActivity.mUserId
16         && top.attachedToProcess()
17         && ((mLaunchFlags & FLAG_ACTIVITY_SINGLE_TOP) != 0
18         || isLaunchModeOneOf(LAUNCH_SINGLE_TOP, LAUNCH_SINGLE_TASK))
19         // This allows home activity to automatically launch on secondary display when
20         // display added, if home was the top activity on default display, instead of
21         // sending new intent to the home activity on default display.
22         && (!top.isActivityTypeHome() || top.getDisplayId() == mPreferredDisplayId);
23     if (dontStart) {
24         // For paranoia, make sure we have correctly resumed the top activity.
25         topStack.mLastPausedActivity = null;
26         if (mDoResume) {
27             mRootActivityContainer.resumeFocusedStacksTopActivities();
28         }
29         ActivityOptions.abort(mOptions);
30         if ((mStartFlags & START_FLAG_ONLY_IF_NEEDED) != 0) {
31             // We don't need to start a new activity, and the client said not to do
32             // anything if that is the case, so this is it!
33             return START_RETURN_INTENT_TO_CALLER;
34         }
35
36         deliverNewIntent(top);
37
38         // Don't use mStartActivity.task to show the toast. We're not starting a new activity
39         // but reusing 'top'. Fields in mStartActivity may not be fully initialized.
40         mSupervisor.handleNonResizableTaskIfNeeded(top.getTaskRecord(), preferredWindowingMode,
41             mPreferredDisplayId, topStack);
42
43         return START_DELIVERED_TO_TOP;
44     }
45     .....
46 }
```

把一些乱七八糟的逻辑去除掉，抽丝剥茧，看到了dontStart 这个变量，下面这个变量的值控制了activity 启动过程中的复用流程，当它是true 的时候，则通过 resumeFocusedStacksTopActivities直接从栈中拉起一个已经存在的activity ，当他是 false 的时候新拉起一个 activity 接收这次请求。

```
1     boolean resumeFocusedStacksTopActivities(
2         ActivityStack targetStack, ActivityRecord target, ActivityOptions targetOptions) {
3
4         if (!mStackSupervisor.readyToResume()) {
5             return false;
```



```
6      }
7
8      boolean result = false;
9      if (targetStack != null && (targetStack.isTopStackOnDisplay()
10         || getTopDisplayFocusedStack() == targetStack)) {
11         result = targetStack.resumeTopActivityUncheckedLocked(target, targetOptions);
12     }
13
14     for (int displayNdx = mActivityDisplays.size() - 1; displayNdx >= 0; --displayNdx) {
15         boolean resumedOnDisplay = false;
16         final ActivityDisplay display = mActivityDisplays.get(displayNdx);
17         for (int stackNdx = display.getChildCount() - 1; stackNdx >= 0; --stackNdx) {
18             final ActivityStack stack = display.getChildAt(stackNdx);
19             final ActivityRecord topRunningActivity = stack.topRunningActivityLocked();
20             if (!stack.isFocusableAndVisible() || topRunningActivity == null) {
21                 continue;
22             }
23             if (stack == targetStack) {
24                 // Simply update the result for targetStack because the targetStack had
25                 // already resumed in above. We don't want to resume it again, especially in
26                 // some cases, it would cause a second launch failure if app process was dead.
27                 resumedOnDisplay |= result;
28                 continue;
29             }
30             if (display.isTopStack(stack) && topRunningActivity.isState(RESUMED)) {
31                 // Kick off any lingering app transitions form the MoveTaskToFront operation,
32                 // but only consider the top task and stack on that display.
33                 stack.executeAppTransition(targetOptions);
34             } else {
35                 resumedOnDisplay |= topRunningActivity.makeActiveIfNeeded(target);
36             }
37         }
38         if (!resumedOnDisplay) {
39             // In cases when there are no valid activities (e.g. device just booted or launcher
40             // crashed) it's possible that nothing was resumed on a display. Requesting resume
41             // of top activity in focused stack explicitly will make sure that at least home
42             // activity is started and resumed, and no recursion occurs.
43             final ActivityStack focusedStack = display.getFocusedStack();
44             if (focusedStack != null) {
45                 focusedStack.resumeTopActivityUncheckedLocked(target, targetOptions);
46             }
47         }
48     }
```

```
49
50     return result;
51 }
```

resumeFocusedStacksTopActivities 函数中首先找到了目标栈，定位到目标栈中需要拉起的 activity，最终通过 resumeTopActivityUncheckedLocked 函数直接 resume activity，具体流程这里不做展开，有兴趣的同学可以自己阅读源码。

回到 dontStart 这个变量，名字看起来就是是否需要启动一个新的 activity，而且他的赋值逻辑里面有一个

```
1    mStartActivity.resultTo == null
```

这个逻辑，而这个resultTo是在什么时候赋值的呢,看到进入startActivityUnchecked的时候调用了setInitialState函数来初始化一些状态。

```
1    private void setInitialState(ActivityRecord r, ActivityOptions options, TaskRecord inTask,
2        boolean doResume, int startFlags, ActivityRecord sourceRecord,
3        IVoiceInteractionSession voiceSession, IVoiceInteractor voiceInteractor,
4        boolean restrictedBgActivity) {
5        reset(false /* clearRequest */);
6
7        mStartActivity = r;
8        mIntent = r.intent;
9        mOptions = options;
10       .....
11    }
```

在这个函数里面用传入的 ActivityRecord 初始化了 mStartActivity 这个全局变量，继续往回回溯这个 ActivityRecord 是在那里构建的。

```
1    final class ActivityRecord extends ConfigurationContainer {
2
3        ActivityRecord(ActivityTaskManagerService _service, WindowProcessController _caller,
4            int _launchedFromPid, int _launchedFromUid, String _launchedFromPackage, Intent _intent,
5            String _resolvedType, ActivityInfo aInfo, Configuration _configuration,
6            ActivityRecord _resultTo, String _resultWho, int _reqCode, boolean _componentSpecified,
7            boolean _rootVoiceInteraction, ActivityStackSupervisor supervisor,
8            ActivityOptions options, ActivityRecord sourceRecord) {
```

```
9      mAtmService = _service;
10     mRootActivityContainer = _service.mRootActivityContainer;
11     appToken = new Token(this, _intent);
12     info = aInfo;
13     launchedFromPid = _launchedFromPid;
14     launchedFromUid = _launchedFromUid;
15     launchedFromPackage = _launchedFromPackage;
16     mUserId = UserHandle.getUserId(aInfo.applicationInfo.uid);
17     intent = _intent;
18     shortComponentName = _intent.getComponent().flattenToShortString();
19     resolvedType = _resolvedType;
20     componentSpecified = _componentSpecified;
21     rootVoiceInteraction = _rootVoiceInteraction;
22     mLastReportedConfiguration = new MergedConfiguration(_configuration);
23     resultTo = _resultTo;
24     resultWho = _resultWho;
25     requestCode = _reqCode;
26     setState(INITIALIZING, "ActivityRecord ctor");
27     frontOfTask = false;
28     launchFailed = false;
29     stopped = false;
30     delayedResume = false;
31     finishing = false;
32     deferRelaunchUntilPaused = false;
33     keysPaused = false;
34     inHistory = false;
35     visible = false;
36     nowVisible = false;
37     mDrawn = false;
38     idle = false;
39     hasBeenLaunched = false;
40     mStackSupervisor = supervisor;
41 }
42 }
43 private int startActivity(IApplicationThread caller, Intent intent, Intent ephemeralIntent,
44     String resolvedType, ActivityInfo aInfo, ResolveInfo rInfo,
45     IVoiceInteractionSession voiceSession, IVoiceInteractor voiceInteractor,
46     IBinder resultTo, String resultWho, int requestCode, int callingPid, int callingUid,
47     String callingPackage, int realCallingPid, int realCallingUid, int startFlags,
48     SafeActivityOptions options,
49     boolean ignoreTargetSecurity, boolean componentSpecified, ActivityRecord[] outActivity,
50     TaskRecord inTask, boolean allowPendingRemoteAnimationRegistryLookup,
51     PendingIntentRecord originatingPendingIntent, boolean allowBackgroundActivityStart) {
```

```
52     .....
53     ActivityRecord resultRecord = null;
54     if (resultTo != null) {
55         sourceRecord = mRootActivityContainer.isInAnyStack(resultTo);
56         if (DEBUG_RESULTS) Slog.v(TAG_RESULTS,
57             "Will send result to " + resultTo + " " + sourceRecord);
58         if (sourceRecord != null) {
59             if (requestCode >= 0 && !sourceRecord.finishing) {
60                 resultRecord = sourceRecord;
61             }
62         }
63     }
64     .....
65     ActivityRecord r = new ActivityRecord(mService, callerApp, callingPid, callingUid,
66         callingPackage, intent, resolvedType, aInfo, mService.getGlobalConfiguration(),
67         resultRecord, resultWho, requestCode, componentSpecified, voiceSession != null,
68         mSupervisor, checkedOptions, sourceRecord);
69     .....
70 }
71
```

发现在 `startActivity` 函数中构建了 `ActivityRecord`，而 `ActivityRecord` 里面的 `resultTo` 变量是通过 `resultRecord` 进行赋值的，这个变量会根据传入的 `requestCode` 是否大于 0 来判断是否赋值。

那就很明白了，我们直接调用 `startActivity` 的时候 `requestCode` 是 -1，而调用传入 `startActivityForResult` 的时候传入的值大于 0 导致在这个位置错过了启动过程中 `activity` 复用的逻辑。最终导致了虽然是一个 `singleTask` 的 `activity` 但是却在同一个栈中同时存在两个实例。

总结

其实这个现象也是可以理解的，当我们调用 `startActivityForResult` 的时候，当然是希望拉起来的 `Activity` 生命周期结束退出后还能回到当前的 `Activity`，但是如果直接走 `singleTask` 的复用逻辑，就会把原 `Activity` 清掉，则无法返回了，可以说这是一个 bug，也可以说 Android 为开发同学做了这一层兜底。

但是 `singleTask` 的 `Activity` 只能在栈内存在一个实例的面试题是不是需要改改了。

测试篇

• 软件测试需要学什么？

- 作者：吴雄琳（荣木）
- 出品：淘系技术

本文从阿里测试工程师亲身经历的角度，和大家聊聊测试一行学习成长的经历。

对自动化测试个人看法

自动化是一个老生常谈的话题，也是一个软件领域非常有技术广度和技术深度的活动，特别是在大型软件的生命周期上。

个人觉得开展自动化测试的难度不亚于传统意义上的软件开发。

从产品角度来看：质量领域本身要求从业人员要全面了解产品、有全局风险意识，例如：产品需求/设计阶段能否发现设计缺陷、产品测试阶段能否发现深层次的bug、产品运维阶段能否制定良好的灰度策略、快速发现、定位线上问题，甚至如何做好新/老系统线上过渡切换等等，这里面都有自动化测试可发挥的空间。

从技术的广度和深度来看：

- 从技术广度来说，不同的技术领域的质量保障需要使用不同的技术（这些技术领域都有一些代表性的工具，但不一定能完全满足实际的项目自动化测试需求），例如有做JUnit接口测试的、有做Web/App/桌面客户端 UI测试的、有做性能测试的、有做用户体验测试的、有做AI算法测试的、有做IoT的、有做压测的、有做各种专项（如兼容性、安全、多媒体、网络）测试的等等，实在太多了.....。

如果考虑到测试工具本身的可用性、系统性，除知道使用工具以外，可能还需要掌握一些基础开发技能，例如：Java/Node/Python后台、React/H5前端、或者Android/iOS客户端；

- 从技术深度来说，想通过开发软件去测试另一个软件是否正常，本身就是一个很具挑战的事情，特别是在黑盒的状态下，举个例子，试想你能否开发一款自动化测试工具能够模拟人的意识形态，它能够对当前多

如牛毛的App开展自动化测试，很多人此时想起了Monkey、Appium、AirTest或者Applitools，其实这远远不够，因为目前并不具备解决场景构建甚至自我发现缺陷的能力，简单来说，还不具备“认知”App的能力。这个想法不是天方夜谭，事实上很多人正在往这个方向努力ing。自动化测试远远不只是一个已有的工具上开发自己的脚本，达到所谓的一个通过率或覆盖率，更核心是思考如何在软件生命周期各个阶段提升产品研发效能及稳定性甚至用户体验。

技术新人如何学习自动化测试

首先简单了解下QA在软件研发迭代过程中的定位

传统软件使用较多的是瀑布模型。测试人员的活动区域是有限的，活动的时间区域主要是提测至上线前。



传统瀑布模型中，QA发挥的空间比较有限，质量压力都集中在测试阶段。随着软件规模的扩大、部门职能的划分、敏捷迭代模式的发展，互联网或者大型软件项目绝大部分演变成了DevOps：



DevOps是软件文化上的一次飞跃，它强调产品、开发、测试、交付、运维各个环节的沟通合作，将敏捷的方式延伸到整个产品。从QA的角度也有了测试左移和测试右移的概念。

测试左移：

测试左移的思想是需求阶段、开发架构设计阶段或是未到系统测试或集成测试前就进行测试，目标是降低时间成本、减少风险，从用户角度描述产品行为、从技术角度建立好开发与产品需求的连接，防止产品设计上的雷或缺陷。这有利于减少无效代码的开发、以投入更好的时间在正确的产品上。也可以在代码编写阶段进行单元测试或覆盖率统计。

日常工作中，QA都期望只对修改的代码或受连带影响功能/需求进行测试，从而减少重复回归的工作量，即“精

准测试”。但是实际上，往往得到开发同学的回复要么是“最好全回归或者核心流程全回归”，要么“是没关系的，就回归下A功能就好”（实际可能已经带雷上线了）。设想如果能够有个工具能够帮我们将需求与相关的代码调用栈联系起来，在相关代码依赖变动时都能够自动评估有效回归范围，可能是“精准测试”实现的一个方向（我相信业界应该已经有人在做了）。

测试右移：

测试右移简单来说是指产品上线以后开展的一系列质量活动。事实证明，在快速迭代以及产品复杂化、多样化的今天，几乎不可能做到0缺陷上线，当然，对硬件产品或涉及资金的产品而言，存在缺陷可能意味着产品召回或是资损，会给公司带来巨大损失，对于某些互联网产品而言，由于产品发布的天然优势，一般具备热修复、热发布能力，因此在时间和产品质量维度，可能会更强调快速上线，比如facebook就提倡灰度快速上线。因而如何监控产品的稳定性、第一时间发现线上用户问题、用户反馈并使问题及时得到解决、如何了解更好的用户需求（如AB测试）变成了QA在测试右移活动中的关注点。期间也有大量自动化测试可发挥的空间。

由此可见，QA发挥的空间是在整个软件的生命周期的，DevOps的理念也强调流程自动化，我理解的在各个阶段能够代替人工工作、提升测试效率的都可以称之为自动化测试。这也反过来要求QA具备更高的软件产品流程/风险意识以及更强的自动化理念、编码落地实践能力。

QA做自动化测试应该掌握哪些技术？

说到具体的技术，其它回复也有提到，感觉整体太散了，初学者可能有点摸不到边，不知从哪里开始，个人建议顺序是这样的：



那让我们先修炼下最基本的内功吧！

软件工程&测试理论基础

各个公司产品形态迥异，因此也制定了不同的软件研发流程。大多数大公司都设置有运营、产品、视觉/交互、开发、测试、运维、技术支持、客服等岗位，应当明白各个角色的职责，以及了解整个产品运转的逻辑。至少应该了解所在公司的研发流程以及当前主流的研发流程（如敏捷开发Scrum），并在项目过程中积极思考，形成

自身的软件意识与理念。在校的同学可以多在网上找找资料，有个大概了解。个人理解，软件工程本身是一个浩大的工程，也在日新月异不断地向前发展，它需要长期积累、不断修炼内功，并在实际项目中实践驱动，从业2年、5年、10年、20年都会有不同层次/深度的理解，自动化测试亦是如此。

关于测试理论基础这里不赘述了，网上资料一大把，搜白盒/黑盒、等价类、边界值等关键字就可以找到。

通用计算机基础（其实就是计算机专业相关的大学课程）

建议至少掌握一门编程语言（C/C++/Java/Python，推荐Python，学习成本相对更简单一些）。相比于特定需求/领域的开发人员来说，测试人员对编码技术要求相对会弱化一些（当然并不意味着不需要极客精神、架构思想）。涉及到Web、桌面GUI、Android/iOS的可以到具体应用再学习相应的框架。

- 掌握基本的数据结构以及在具体程序语言中的应用，例如：list、map。
- 掌握面向对象程序设计的基本思想。
- 掌握一种代码管理工具，如git、svn。
- 掌握Linux的使用及基本命令使用，如：cp、grep、vi/vim等。
- 掌握关系数据库的基本理论和关系数据库（如MySQL）SQL基本使用、NoSQL（如Redis）的基本使用。
- 掌握基础的计算机网络理论，如TCP/UDP协议、IP划分。

接下来，我们就需要站在巨人的肩膀上了。这部分可以根据实际需要进行学习，涉及的内容实在太多了，我这里主要从App自动化测试的角度给出一些工具使用、方向学习建议，大家搜关键字应该都能找到一些资料。

服务端：

- 白盒单元测试：JUnit(Java)、unittest(Python)、gtest(C++)
- http接口测试：Postman
- 抓包工具：Charles、Wireshark
- 压测：Jmeter，在大厂里面都会有特定的一些写好的工具可以使用。
- 链路依赖分析：梳理应用间的依赖关系，提供压测模型，大厂里面也有一些工具可以使用。
- 监控&日志分析：应用稳定性监控，如qps、rt，服务器负载、cpu监控等。日志分析这块可以做一些基于
- 规则的错误日志监控、甚至基于AI的方式（如：机器学习）对日志大数据进行聚类、问题分析/定位。

客户端（Android/iOS/H5）：

- UI: Appium、Macaca、Airstest
- 性能（CPU/内存/帧率）：Android Studio、Instruments(iOS)
- 稳定性: Monkey
- 兼容性: 各种云真机平台

事实上，即使非常熟练掌握了以上工具，也无法达到完全释放人力的目的，甚至在自动化实践过程中会存在各种各样的问题（例如如何针对具体的场景设计自动化用例、提升覆盖率、如何维护/构造测试数据、如何进行精确校验、如何提高执行稳定性、如何缩短执行时长、如何监控线上问题等等）。

这就需要我们更加深度的去了解产品形态、在已有工具解决不了问题时，怎么去用创新的思维看待各个阶段面临的问题、甚至创造工具，这已经不仅仅只是技术本身的问题了，而是如何去挖掘、思考问题、如何去运用技术的问题了。实际上自动化测试可以归纳为如下几个阶段，这也是近2年智能化测试的研究方向：



另外，个人觉得目前市面上对自动化测试其实是存在的一些误区的：

- 认为自动化测试无所不能，只要有自动化，人工可以完全释放。对于复杂产品，目前来看，这几乎是不可能的。因此，目前市面上并没有类似“统一宇宙理论”的自动化测试工具。
- 自动化测试没技术含量（测试没开发有前途），如果仅仅是对工具使用，没有创造力，那确实没有什么太高的技术含量。但是如果是在DevOps各个阶段发挥最大价值，个人认为比传统意义的开发岗位难度更高，并且可发挥空间更大。举个例子，能否创造一种工具，能够根据视觉稿或者App UI自动生成测试用例（啊？怎么可能，试想下人是怎么设计用例的，得益于AI技术的发展，这完全有可能，目前也有一些根据视觉稿生成前端代码的工具了）。我不觉得这是一个没技术含量或没有价值的自动化测试。近些年，质量领域的大会越来越多，大家也可以关注关注。例如QCon、MTSC、TICA、Tid等等。
- 自动化覆盖率至高无上。这部分往往来自人们对自动化测试过高的期望，为了提升覆盖率，未考虑好ROI，以至于南辕北辙，入不敷出。著名的测试金字塔给了最好的解释。
- 人云亦云，泛而不专。相比于开发人员，个人认为QA开展自动化测试需要掌握的技术知识可能会更广泛，例如：开发人员可能专注于Android、iOS或者Java后端，亦或是某类AI算法，即对开发人员的要求是对某一技术掌握的足够深入，对QA来说，精力有限的情况下，可能又要懂点服务端、又要懂点客户端、又要懂点算法等等。再加上各种产品、技术形态不一，也衍生出了形形色色的工具和研究方向，初学者容易受到误导，不知所措。自动化测试和开发角色一样，也是一个逐步积累、实践的过程。

关于自动化测试确实有非常多的内容可以交流、学习，篇幅有限，先写到这里啦。以上内容是个人对自动化测一些理解，当然，如果继续往上走，到管理者，需要掌握的知识也远不止这些了。

● 基于强化学习的质量AI在淘系互动业务的实践之路

- 作者：徐兵（纵马）
- 出品：淘系技术

导读：AI人工智能的概念由来已久，因为alphago在围棋领域击败李世石掀起了全世界范围内的AI热潮，最近又随着DeepMind破解蛋白质折叠难题这一诺奖级成果再次让我们发现AI已经进化到了如此强大的程度。与AI在学术界的关注度相反，工业界中AI的应用则显得悄无声息。但是深入到各行各业中却会发现，从人脸识别到物流路径规划，从服务器数据传输到数据安全，AI已经无处不在。

作为软件行业的重要组成部分，质量保障的技术体系经历了脚本CI时代/大数据测试时代，也正在向智能化时代演进。质量保障工作最主要的价值和目标，是全面找到业务的风险点，并围绕这些风险点构建解决方案。遗漏的风险也许就是毁灭拜占庭帝国的凯尔卡门，让产品的用户体验崩塌。

AI如果能类比alphago，在质量保障领域中做得比领域专家更为出色，才能让这一技术成长到奇点，真正改变质量行业。

互动的业务场景如盖楼最为大家所熟知的，本文将从互动的业务场景出发，介绍互动如何构建质量AI，并用质量AI解决了一系列业务痼疾，让质量AI落地生根。

背景&挑战

互联网电商中，迭代速度在竞争中能决定业务的生死。迭代速度和慢工出细活打磨用户体验看上去非常矛盾，在互动这种强用户体验的产品线矛盾更加突出。用户的游戏化体验是由一条条的用户行为路径的顺畅度和产品提供给用户的获得感来决定的，那我们就找到一个很好的切入点，如何用AI将用户行为路径全面模拟出来？

在解决这个问题之前，我们先来看看互动的业务场景。第一类是手淘上现有的三个典型互动产品，金币庄园，淘宝宝人生和芭芭农场。第二类是大促互动的产品，也就是大家都接触过的盖楼/养猫这样的大促互动。



这些互动产品的共同特点是极度追求用户的体验效果，不断在玩法上求新求变，没有那次大促大家会玩到不变的互动。同时电商场景下对迭代速度的要求也不断趋向极致，2个月需要完成一款其他游戏公司需要半年才能完成的作品。

在这种情况下，互动需要解决如下三个问题：

- 环境：互动的测试环境数据构造除了基础的部署环境意外，还有大量的用户状态数据、规则配置数据和权益数据，可能构造一个拉100人在盖楼获胜的场景用例需要一整天而测试执行只需要1分钟。如何构造一个高频易用的环境？
- 用例生成：互动的特色决定了大部分的功能都是新功能，是没有线上流量可以参考的，这种情况下大数据测试方案会陷入没有数据输入的困境。如何解决新功能的更全面的用例生成？
- 资损：互动有大量的权益发放，基于现有基于对账的事后资损防控方案，在互动这样的巨量场景下只能说是亡羊补牢。如何提供事前更全面的资损风险识别方案？

构建基于强化学习的质量AI

这个时候再回过头来看互动产品的特色，无论是做任务升级领红包，还是拉人盖楼pk领红包，都是由一条条用户的行为节点构成的行为路径来构成，如果我们能把互动的产品按照一个个的行为节点解构成一个有向图，那互动的用例生成也就抽象成了如何构建一个有向图和求解有向图中的路径集的问题？环境问题是AI执行的前置条件，而风险包括资损风险和架构风险的探测则是AI的用武之地。

有了思路，我们再来一步步看如何让AI真正在业务中落地。

高频易用的环境

现有测试环境的解决方案更多基于如何降低系统部署对于分布式系统的影响来解决，包括监控应用的部署/构造隔离的分布式系统的镜像环境/从变更管控角度管控部署的时间段等等，主要解决不同代码部署情况下的环境问题。对于互动这样迭代速度极致求快的业务，这些方案都还是成本太高。

互动的解决思路是不改变部署结构的情况下，通过数据隔离的方式来解决。



互动环境的的目标是隔离稳定和高频易用。

- 隔离稳定：通过测试账号和影子数据库来实现流量隔离；通过系统架构改造，将互动所有的配置数据读取的方式进行收口，实现配置隔离，只有指定的账号能读取指定的配置。这样可以实现一套部署结构，同时允许正常的业务逻辑运行和测试流量执行，且互不影响
- 高频易用：由于没有代码和部署解构的变化，环境的准备变成了测试数据包括配置数据的准备，理论上可以做到随备随用。

有了这样的稳定的环境，AI的落地才有了前提条件。

基于强化学习AI的用例生成

既然互动的业务场景抽象成了基于行为节点的有向图构建，已经有了很多针对有向图的算法解决方案。这个时候深入业务的特点和诉求看看要解决的核心问题，再来选取合适的解决方案。

- 新功能无输入或者少输入：互动的玩法大部分是新功能，能提供的没有大规模可以调用的线上请求数据或者日志数据，只有代码和代码相关的一些接口定义。
- 要全面：盖楼这样的玩法的用户量级决定了遗漏的风险真的会摧毁整个产品的体验。生成的用例需要更全面，能否发现人难以发现的问题是一个很重要的衡量标准。

• 自动化：智能化的前提一定是充分的自动化。从给定的输入到训练，从训练到用例生成，从用例生成到用例执行需要完成全自动化。



我们的解法是基于强化学习来构建用例生成的AI。

- 新功能少输入：用户的行为节点分为两种，一种是无系统数据传输的比如页面的滑动等纯前端交互事件，另一种是有系统数据传输的如按钮点击，这一种我们定义为用户的有效行为。在系统实现上有效行为最后会变成对后端接口的调用。因此基于接口定义而不是基于请求数据来建立有向图模型能极大减少需要的输入。在互动的实践中，只需要配置好需要的接口和对应的入参类型即可。
- 全面的用例生成：互动的接口调用我们认为有先后顺序的，这是从接口的参数来体现的。因此有向图的路径构建问题，我们选用了强化学习来构建。这其中最关键的路径覆盖全面问题是一个十分复杂的问题，逻辑调用路径是代码接口调用路径和数据逻辑路径等众多因素的组合物。在互动的实践中，我们针对最急迫的冒烟用例覆盖问题，对探索策略也结合pagerank和dfs做了一些改进，同时对激励函数做了接口参数状态覆盖优先的优化。从最终的实践效果来看，发现了很多容易遗漏的状态遗漏和动态概率计算问题。
- 全面自动化：这里最重要要解决的工程问题是将算法跑出来的接口调用路径，转化成可以有最终用户使用的可重复执行的用例，这里不做赘述。。

体验还原

自动化生成出来的用例要最终在业务线产生世纪作用，提高易用性是关键。而提高易用性的关键是打通端到端，让生成出来的用例不只是数据，更要成为具像化的可交互的端上的效果。



受益于近几年前端的数据驱动架构演进的红利，我们可以方便地在工程上实现给定接口数据，即可在端上复现出对应的交互行为。结合图像断言等技术可以打通用例自动验证这个测试智能化的最后一公里。

质量AI解决业务问题

当我们在互动场景中解决了环境的问题和用例的自动化生成问题，在工程上也打通了端到端的体验还原，构建了基于强化学习的质量AI，需要真正解决业务中的问题才能业务真正带来价值。

权益链路预跑

资损问题最近两年成为了业界的一个棘手问题，也出现了很多基于对账的资损问题发现的思路和产品。但是如前所述对于互动特别是盖楼这样量级和短周期的产品，事后的亡羊补牢虽然能减少损失但是损失的量级依然很大，互动需要能事前探测资损风险的方案。这样的方案难点主要有二：

- 什么样的场景会发生资损难定位：资损场景虽然是业务逻辑场景的其中一个子集，如果业务逻辑场景本身都有遗漏，资损场景遗漏更加难以避免。现在的大数据测试方案对于回归类型可以解决，对于新功能新变更还是存在输入数据不够的问题。
- 资损发生的方式难模拟：资损发生的方式花样繁多，代码逻辑错误/异常处理/并发处理/数据污染/配置错误等情况层出不穷。



对于质量AI来说，问题1可以归纳用例如何生成更全，问题2可以归纳为环境准备问题。在实现了冒烟用例的AI化之后，互动通过对AI的迭代加入更多如代码路径覆盖/数据血缘分析/异常的执行环境注入等方式逐渐演进风险探测的准确率。目前互动已经实现了核心权益发放场景完全基于智能AI的权益链路预跑。

容错探测

在分布式系统中，单点的故障不仅仅影响自身功能。因此每个系统在做好自身功能验证的事后，还需要考虑对关联系统和关联功能的容错能力，保障极端情况下的自身功能可用性和用户体验。



目前互动核心的功能故障点的容错探测都以实现AI化。

总体展望

质量保障整体在整个研发体系中处于守门员的角色，在整个世界都在全力拥抱数字化的大背景下，系统的复杂度提升越来越快，业务对迭代效率的诉求也越来越强，传统的解决方案越来越难以满足时代的要求。

智能化给出了质量行业的另一个产品化的解决，未来的质量保障一定是通过研发更好的质量产品来保障业务的快速迭代，而根据各个业务定义的质量AI会是下一代质量产品的最佳选择。虽有曲折，必有改变。



● “真正的机器人测试” — 淘宝泛终端机器人自动化测试实践

- 作者：荣木
- 出品：淘系技术

“业务先赢”、“极致体验”一直是质量人的主旋律。

很荣幸本文内容入选了阿里巴巴内部《阿里集团21财年最值得读的前沿技术领域热文合集》，也是唯一一篇质量领域的文章入选“人工智能”板块系列，也在此跟大家一起分享交流。

另外，本文介绍的工具也很荣幸作为软件绿色联盟云测试平台工具之一，并逐步开放给行业使用。

背景

随着移动互联网的蓬勃发展以及5G时代的到来，智能移动终端普及从智能手机拓展到 IoT，商业模式也逐步实现线上线下载通，产品设计时，多App/多终端之间的交互也越来越多，例如消费者与商家互动、用户与主播互动等。随之而来的终端系统也拓展到了Android、iOS、鸿蒙、Win Touch、RTOS等等，多APP、多端矩阵也逐渐成为业务触达用户的常态。下图便是一位测试人员在不同平台上模拟消费与商家沟通的一个场景。



UI自动化作为“业务先赢”的一种重要手段一直在不断演进，从泛终端与多端UI自动化视角看，行业内主要有以下或类似的解决方案：



从“极致体验”角度来看，加载时长评测占据半壁江山（还有普遍关注的内存/CPU性能、帧率/卡顿、Crash稳定性），行业内主要是以下或类似解决方案：



传统的方式存在一些缺陷：

- 1) 人工点击录屏耗费人力；
- 2) 基于埋点的方式能够反映技术优化结果，但很难作为用户真实场景体验的精确数据；
- 3) 基于系统驱动的方式对待测设备有侵入，多个系统多套脚本，覆盖场景比较有限（如非Android/iOS的IoT设备），难以跨平台支持；

综上所述，在“业务先赢”、“极致体验”的道路上，依然存在需要解决的质量保障难题：

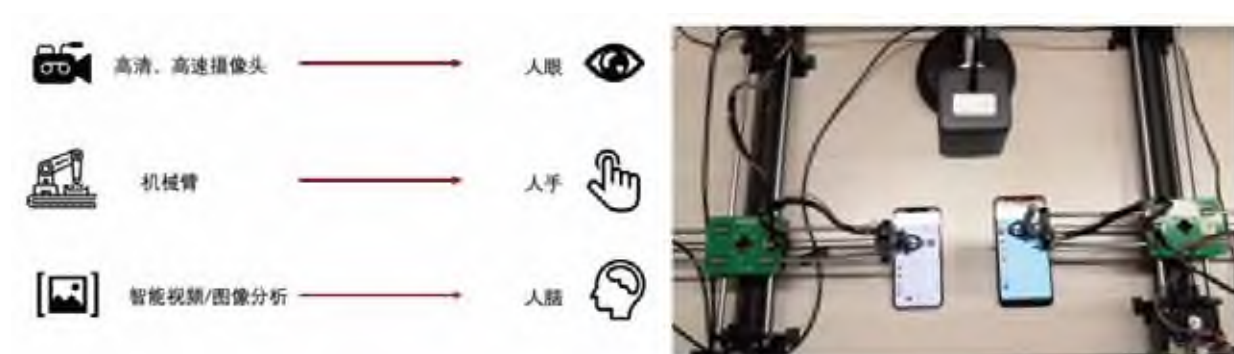
- 从泛终端多端交互全流程的视角来看，目前功能自动化测试依然面临每个端（iOS/Android/PC）各自维护一套脚本或是“伪”一套脚本，成本高；
- 多APP矩阵间的交互场景以及同场景下的竞品分析日益增多，亟需支持多设备互动的自动化手段；

互联网市场厮杀日益激烈，产品形态多样性放缓，拥有极致用户体验的互联网产品才能赢得更多的市场与消费者，期望寻求一种立足“用户真实感受”、对被测系统无侵入、计量精确的解决方案。

- 在快速迭代/变更、复杂功能需求、AB测试的互联网产品背景下，完全依赖人工编写/维护自动化脚本已无法满足需求：系统变动自适应能力差；不能自行探索构建测试场景；传统的单点验证测试覆盖面小，很难在项目中快速应用和广泛推广。

我们的策略

为了应对以上难题，我们跳出了常规思维，尝试使用引入摄像头、机械臂、算法代替人眼、人手、人脑开展泛终端时代多端交互的UI自动化与用户体验评测，同时，我们提出了UEE自动化测试的概念。



什么是UEE自动化测试：

通过IoT（如机械手、摄像头）等外部设备模拟真实用户操作，实现以外部真实用户视角对AUT（APP Under Test）进行的端到端黑盒自动化测试。以“真实”模拟为核心目的，具体包括两部分：

1）用户交互体验评测：通过IoT（如摄像头、传感器）等外部设备采集并将真实用户感受数字化，来获取最接近真实用户体感的性能及用户体验指标；软件绿色联盟在2020年3月份也发布了《软件绿色联盟智能终端性能流畅评测标准V2.0》，并以机械臂+摄像头的方式作为加载耗时的评测标准，我们在2019年初就开始使用这种模式。

2) 多设备端到端功能自动化：通过机械手实现同时多设备的跨终端的自动化操作执行；通过AI算法实现IoT设备的调度控制、场景理解、测试步骤执行、测试路径探索、体验指标计算、异常处理等能力。

通过机器人对系统进行黑盒测试，类似的做法在国外Axiz机器人也有尝试。

为什么要UEE设备端到端自动化测试

E：满足深度用户体验评测的诉求（Experience）

更高精度：随着深度用户体验提升的需求日益增加，端侧的性能和体验优化会精确到ms级别，因此需要找到进一步提升用户体验指标度量精度的方法。

多设备间交互测量：随着多端交互的场景越来越多。不光是APP之间，设备层面APP和PC乃至IoT之间交互的场景也不断出现，需要度量多APP多设备之间的交互性能及用户体验，需要度量多设备多系统间（从A手机到B手机，手机到PC/IoT 等）的交互式体验指标。

真实模拟：软件模拟点击和真实用户操作还是存在差异，要探索一种更模拟用户真实操作的方式。

E：对多设备端到端功能自动化测试的支持（End1 to End2）

业务客户端发版经常需要对Android/iOS/Pad甚至PC进行覆盖测试，同时APP之间的业务交互场景也逐渐增多，比如淘系的消费者-商家互通全流程，测试不仅仅只是单手淘、天猫或者千牛APP覆盖，还要能覆盖和串联起从消费者到商家的整个业务流程端到端自动化。此时，我们面临着新的挑战：

多机交互自动化：无法快速支持多手机 App 间、手机与PC/IoT之间交互的操作；

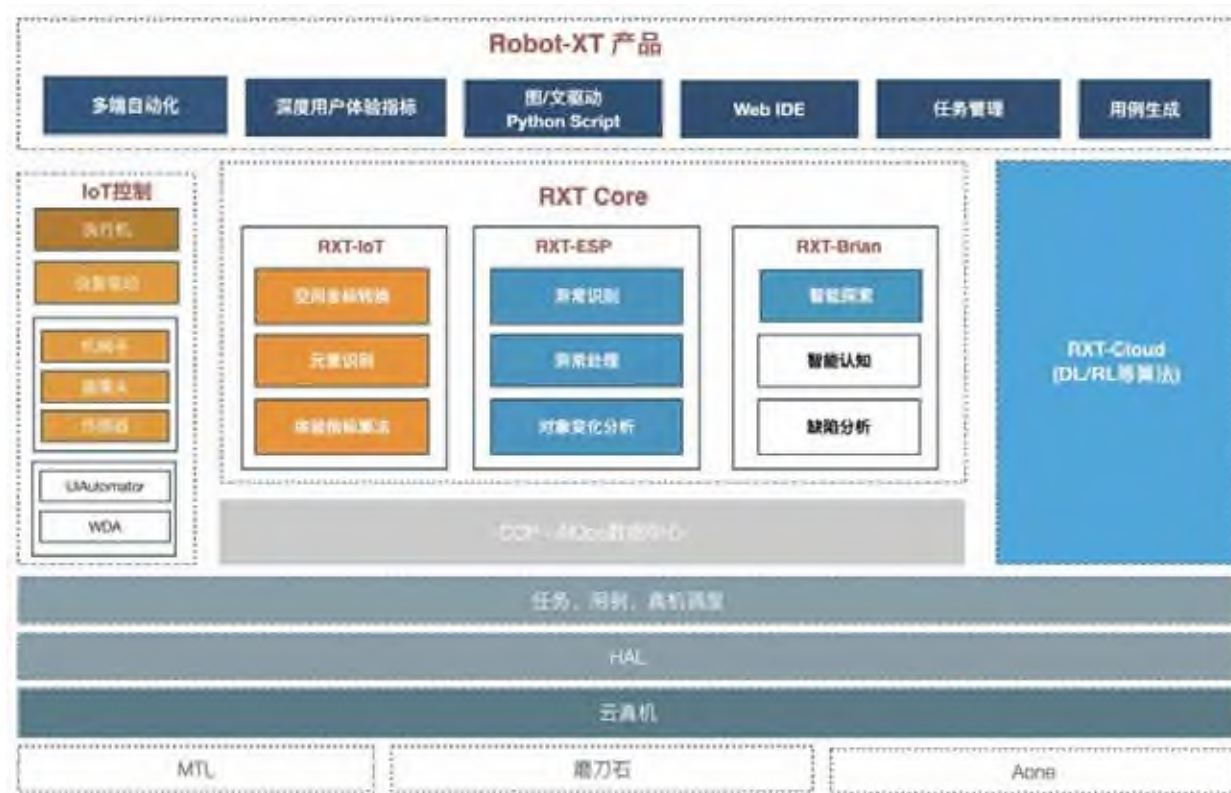
基于外部视角的对象识别：CV和AI算法的加持让UI自动化测试在对象识别上有了新的突破，但依然无法摆脱软件层API操作的局限，受所在操作系统限制，依旧存在部分特定场景下元素无法识别的问题（如系统内Push消息操作）；

一套跨终端设备的通用脚本：还不是真正意义上的多设备多平台一套脚本驱动，Android/iOS/PC/IoT受系统版本或环境影响，可能仍需要多套自动化脚本，维护成本高；

一个脚本一次执行覆盖多个设备和系统：由于不同操作系统有不同的自动化执行引擎，一个脚本一次执行无法同时对Android/iOS/PC/IoT 完成回归。

RXT(Robot-XT)迎接挑战，应运而生

基于UEE自动化的理念，我们尝试使用机器人的方式获取用户体验指标以及开展UI自动化测试，设计并研发了RXT。它基于IoT架构，从“用户真实感受”出发，零侵入被测系统获取用户体验指标（加载耗时、跳闪白分析），同时支持图/文驱动跨设备交互的UEE自动化用例批量执行。下图是RXT的能力介绍：



深度用户体验指标

RXT立足“用户真实感受”获取用户体验指标，用机械臂手代替人手操作，用高速摄像头代替人眼记录操作过程，并具备体验指标算法(不需要目标对象的加载耗时分析、白屏分析)的大脑结构，RXT也可支持设备交互场景及竞品对比场景。

多端自动化

RXT在传统的单机操作之上引入一层新的“主控大脑”，接管、调度跨设备的时序操作，完美支持了多端多设备、多App间交互场景的自动化能力；RXT使用原生Python程式化(基于unittest)组织用例，用户可灵活控制/增加定制逻辑，并基于Python module理念抽离公共逻辑，降低用例维护成本，此外，用例可通过任务组织批量执行；RXT上层用例基于图文方式1套脚本驱动，底层执行引擎支持机械臂引擎和UIAutomator/WDA自动化引擎，其中机械臂方案做到与平台、设备无关，打通手机和PC(平板)间自动化的壁垒，满足各种场景的测试需

要；“非预期”弹窗(突如其来的红包Poplayer、首次安装需要的各种权限、过渡动画广告等等)是自动化测试的灾难性问题，RXT-ESP提供了相关的识别和处理机制；RXT提供真机用例调试能力，用户无需准备任何本地环境即可运行、调试用例。

对象识别

多场景传感器融合下的对象识别，使用ROI截取、图像增强/去噪优化被测对象图片，坐标变换关联手机元素位置与机械臂操控位置，基于OCR获取目标图像文本信息，改进SURF/SIFT特征匹配算法提升图像识别准确率。

异常处理

RXT具备非预期弹窗的检测与处理能力，例如可以检测首页Poplayer红包弹窗、权限弹窗等，并且使用NLP对互动按钮(如“我知道了”，“允许”，“不再提醒”)进行语义理解、分类，然后加以处理。

用例生成

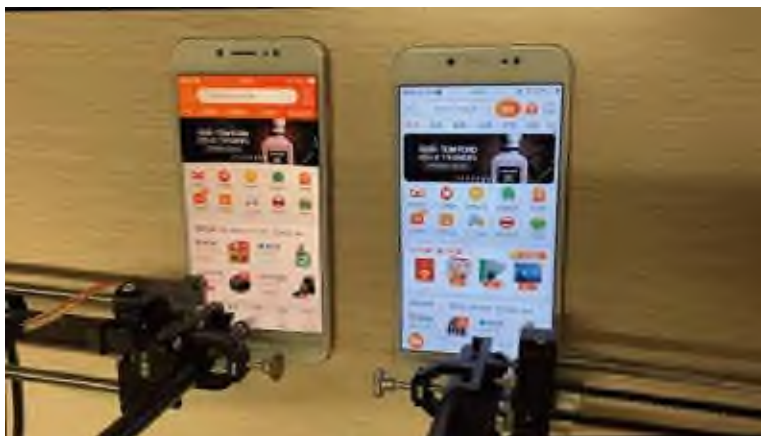
RXT能从分散杂乱的用户埋点数据中，通过清洗、聚合、去重构建手淘消息用户行为访问路径及Top N热点路径，从而转变成具备业务语义的用例，后者正在进行中。

项目实战

手淘消息场景用户体验升级项目

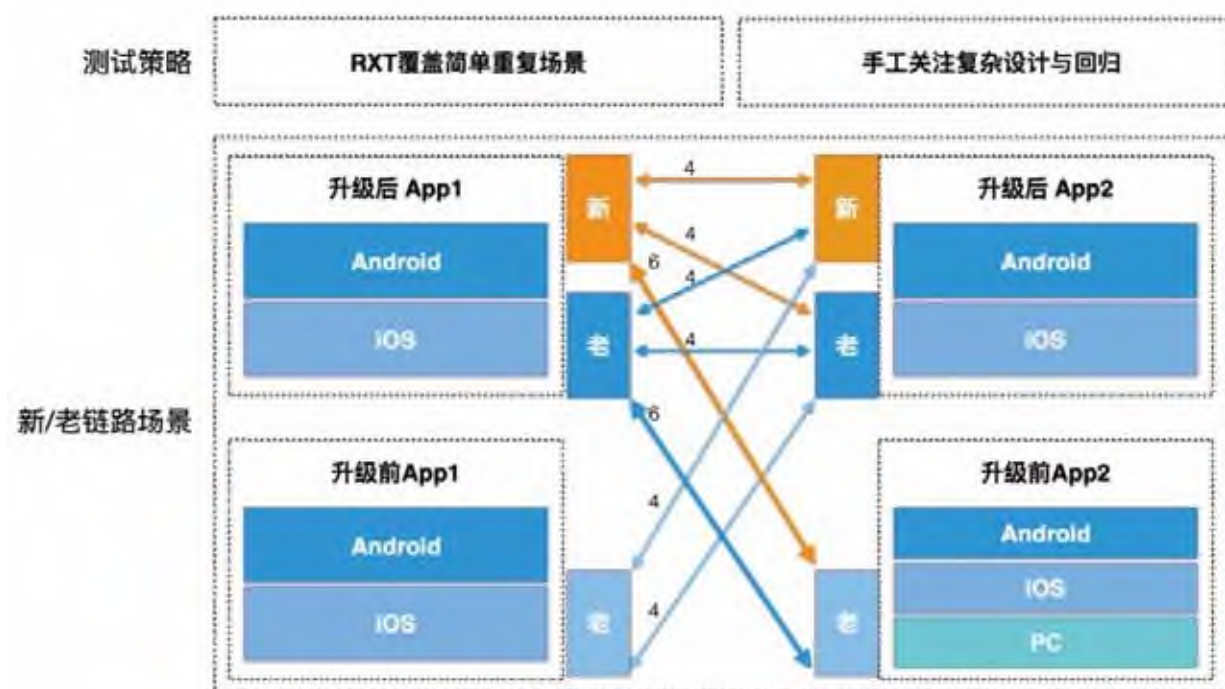
项目目标简介：手淘消息是世界上最大的商业沟通场景，该项目目标是大力提升手淘消息的加载体验、并进行跳、闪、白治理

策略&效果：下面是该项目优化前后，使用RXT观察到的用户使用前后版本打开消息业务的效果对比视频。



手淘消息多通道升级项目

项目目标简介：该项目涉及到底层链路新/老切换，与上层客户端业务进行对接，为降低质量风险，上线期间兼容了新/老链路方案，对应的测试回归场景涉及到手淘消费者(Android/iOS)、千牛商家(Android/iOS/PC)双端的新老链路交互，按笛卡尔乘积增长，共有48种，每种场景都基本对应同一套测试用例。



策略&效果：上层业务上存在大量重复Case回归，我们的策略是将简单、重复的用例交给RXT，人工投入更多的时间在复杂场景的用例设计和回归上。

大促态消费者-商家下单核心链路验收

项目目标简介：大促态下，消费者提交订单是核心链路。不仅需要保障手淘消费者(Android/iOS)提交订单成功，并且需要保障千牛商家侧（Android/iOS/PC）能够实时收到消费者提交的订单信息。

策略&效果：如下是实战效果(手淘-千牛移动端)：

总结及未来展望

本文主要介绍了UEE自动化测试理念，并在此基础上设计并研发了RXT以及相关的一些项目实践结果。我们还将继续在“业务先赢”、“极致体验”的道路上继续前行。未来我们将继续优化RXT产品，继续提升产品稳定性、智能化能力。同时也希望我们的能力可以更多对行业开放，让智能化测试服务行业，实现真正的测试智能。欢迎大家提供合作机会！！

同时我们也会在后续陆续发布与本文相关更多的技术方案和细节，例如：多传感器融合复杂场景的元素识别、智能异常处理、工程化开展等，敬请期待！

附：什么场景下建议开展UEE端到端用户体验测试？

建议UEE自动化作为UI自动化的辅助和补充。以下场景建议使用UEE自动化测试：

- 1、单设备APP用户体验指标评测：单手机APP对精度有更高要求，更真实模拟用户体感的场景。
- 2、多设备APP间的用户体验交互或对比评测：比如手淘消息和微信消息的用户体验指标对比、不同APP启动时长对比等等。
- 3、多设备交互的端到端功能自动化测试：比如社交类场景的卖家B和买家C的消息互通测试、手淘买家和千牛PC卖家之间业务依赖的B到C全流程测试等。

● 淘系产业级移动应用智能测试最新方案 MonkeyBot公开！测试效能提升3倍以上

- 作者：居凡
- 出品：淘系技术

产业级移动应用测试面临操作系统碎片化人工测试负担过重、功能复杂场景繁多、自动化测试探索效率低效的问题。淘系技术质量团队与北京大学讲席教授谢涛老师团队合作，在业界提出了首个基于计算机视觉（和多模态强化学习）的自动化视觉测试框架Monkeybot，解决了遍历测试效率低下，自动测试工具难以跨平台使用的问题，为移动应用测试提供了通用高效的解决方案。并于21年9月底在QECon全球软件质量效能大会上对行业公开了这一技术方案。随后，在10月份云栖大会上谢涛教授对MonkeyBot落地效果进行了分享展示。



Monkeybot在阿里巴巴集团包括淘宝、千牛等在内的十余个大型产业移动应用上落地部署，服务于他们的日常测试和质量保障，特别是双十一等重要场景的质量保障和测试工作中，并通过软件绿色联盟向全行业提供用户体验测试，智能探索测试，智能验证等服务。已在Android、iOS、鸿蒙等操作系统落地实践，推动测试效能提升3倍以上。

接下来一起看下我们是如何来实现的吧。

研究背景

手淘拥有十亿级用户，产品快速迭代和功能复杂多样化给质量保障工作带来巨大挑战，面临功能测试回归成本高、易漏测等问题，迫切需要一个高效智能的自动化测试方案来保障软件质量。

解决复杂交叉场景易漏测问题

随着淘宝业务的扩大，当前手淘不仅具有购物下单的功能，还有很多其他的业务功能，比如有消息聊天、直播、逛逛、互动游戏等。不同的业务都是划分在不同的团队负责，因此各业务的上线和回归都是相对比较独立的。但是，从用户角度很难区分每个业务的边界，用户使用产品过程一定是一个多场景组合的情况，比如用户买东西的时候想跟商家沟通一下尺寸号码的问题，用户大概率会从商品详情页找到客服，然后进到聊天页面发消息；结果商家没有及时回复，下次可能会从消息tab找到会话去聊天，然后逛着逛着商家回复了又可以从新消息弹框点进去聊，聊完了可能还要返回继续逛，逛着逛着又想起一些事情要跟商家交代，于是从快捷入口进到消息再找到会话聊天。一个简单且常用的聊天功能，用户可能会有很多路径去完成，过程中也会涉及到很多消息之外的业务，对于测试同学很难全面覆盖组合场景验证，也就容易出现交叉场景漏测导致线上问题发生。



解决快速迭代回归成本高问题

手淘客户端每次迭代发布各个业务都需要进行回归验证，每个月手淘至少一次正式版本发布，大概每周都有一次灰度发布，对于每一次发布都要进行至少两轮回归，每次回归至少覆盖安卓和iOS两个系统，保守计算每个月至少要做16次功能回归，每次回归都需要大量的人力成本。

为了降低回归成本，UI自动化测试广受大家喜爱，Selenium、appium、Airtest、RXT等工具陆续出现，通过维护自动化脚本来实现测试自动化，也取得了不错效果。但是随着产品快速迭代，维护自动化测试脚本仍然需要很大的人力成本和时间成本。

为了更好地为测试提效，让机器尽可能以接近用户的视角来自动执行测试流程，尽可能多地覆盖不同业务场景，受遍历测试和Monkey测试的一些启发，我们尝试让机器在给定app内自主探索，结合智能的探索策略，具备类人测试能力，实现全流程无人参与的自动化测试。

思路分析

为了让机器具备类人测试能力，我们制定了以下几个目标：首先，在执行层面我们希望一套测试工具支持多系统多平台验证；其次，测试工具需要摆脱对测试脚本依赖，具有主动理解业务的能力；最后，在理解业务的基础上能智能做出决策执行操作，有效覆盖更多用户操作路径。

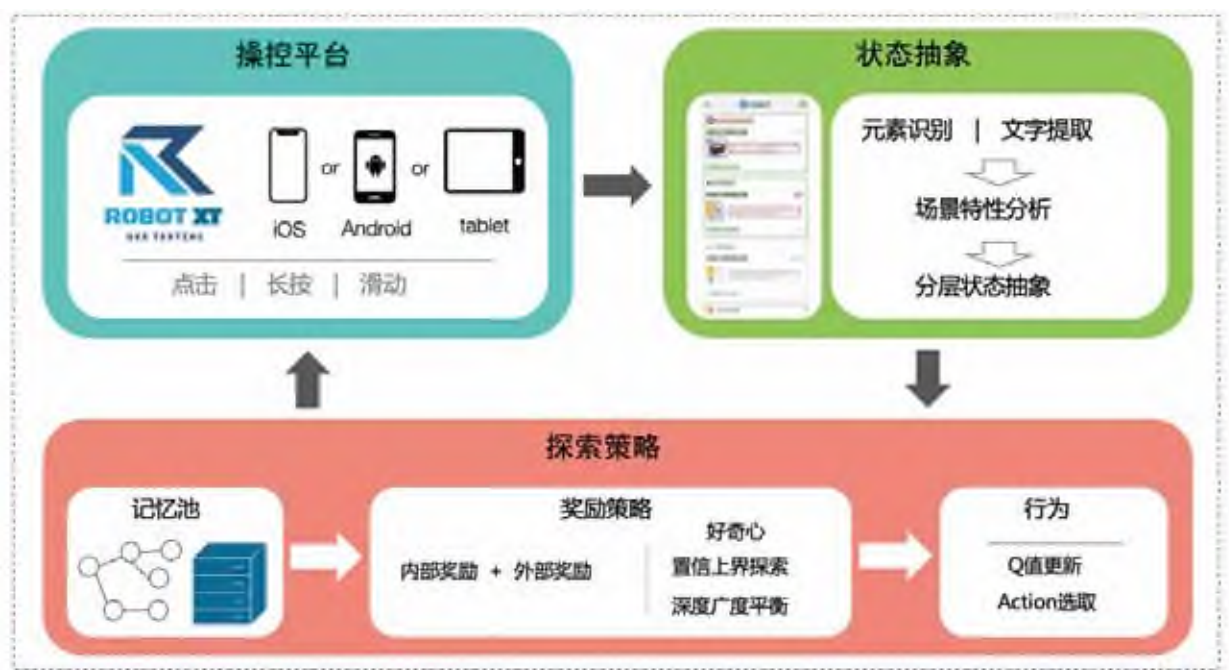
显然，这些目标是很难实现的。但是，我们仍然要在这布满荆棘的道路上摸索前行！

通过长时间的技术调研和尝试，我们找到了如下图所示的一套解决方案。第一，基于纯视觉的UI分析，不再依赖系统信息完成分析，更好的适配不同系统；第二，通过图像特征、文本信息和UI的结构等信息来对UI状态进行建模获取图文多模态信息，然后对于图片进行不同层次多粒度状态抽象，更好地理解图像业务含义；第三，引入强化学习探索策略，结合探索历史和智能探索策略，实现更有效地路径覆盖。



MonkeyBot 技术框架

MonkeyBot测试执行是依赖RXT平台完成的，RXT本身具有兼容不同系统的优势，因此也达到了摆脱系统依赖的目的。对于状态抽象层主要对获取的UI界面的图像特征、文本特征及图像结构等进行分析，多模态多粒度的状态抽象为我们压缩了UI信息，也为决策智能技术的入场打好了地基。探索策略部分则主要包含奖励设计，动作选择和参数更新，结合历史状态访问情况和设计的奖励策略进行行为决策。



具体执行流程如下：

1. 执行进入app初始状态。
2. 对当前状态UI界面进行抽象，通过元素识别、文字提取和图像结构解析后的信息进行场景特性分析，然后再进行分层状态抽象。
3. 对于抽象的状态信息与历史访问状态信息进行比对，结合好奇心策略、置信上界探索策略以及深度广度平衡等策略分析后，得到一个奖励值分布，并完成参数更新。通过奖励值分布情况选取下一步的行为。
4. 通过RXT完成执行操作，并获取新的状态。
5. 以此循环执行，直到指定探索轮数或者指定探索时间结束。

UI 元素识别技术挑战和解决方案

近十年来计算机视觉在自然图片处理中展现出了傲人的效果，无论是基于统计学习的类人视觉模型还是基于深度学习和数据驱动的模式化神经网络，计算机视觉在诸如图片分类，语义分割，目标检测等任务上都有接近人类的表现。尽管计算机视觉在自然图片处理任务上如火如荼的发展，在UI这一特定场景上的研究却相对缺乏。

我们通过在手淘上的大规模实证研究表明，直接将现有的计算机视觉技术不加改动地应用到UI分析任务上去会导致较差的结果，想要把整个UI界面中的所有元素信息比较精准的分割识别出来是十分困难的：

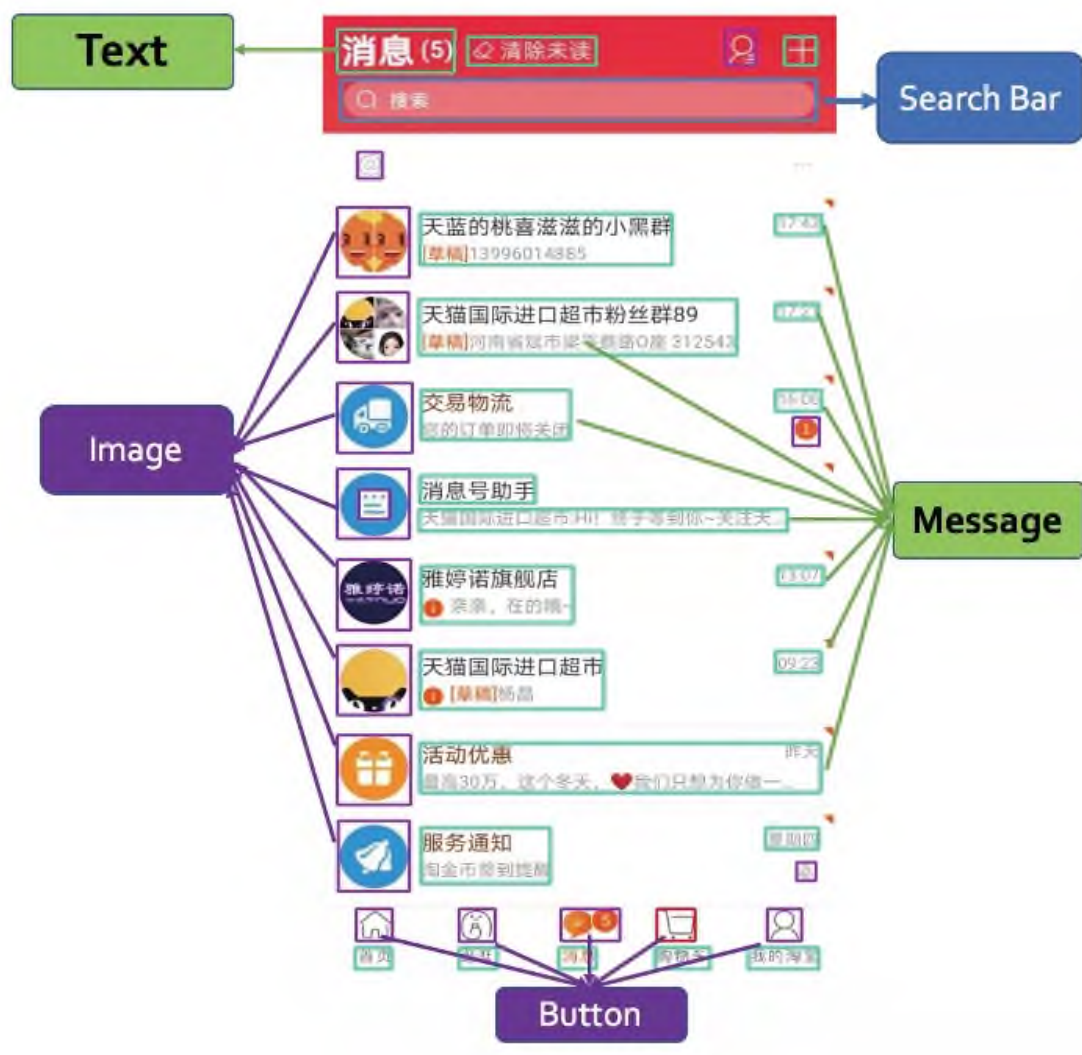
- 1.通过传统图像分割方式很难设定一个较好的阈值，容易把多个元素合并成一个，或者把一个整体元素拆分成多个小的元素。
- 2.想完全依赖深度学习的方法来训练识别元素也是行不通的，对于手淘这种大型流行应用，UI界面大部分的元素是基于业务设计的，样式多变，不统一，且更新迭代快，因此几乎不可能完成对所有元素数据集的收集，难以在实际使用中落地的。

为此，我们发展了特别针对UI图像的综合视觉分析技术。

首先，我们的技术是低数据依赖的：针对没有测试历史数据的移动应用，我们的技术基于动态图像滤波，使用多种滤波算法和自底向上的基于视觉统计的梯度分割与区域聚合，无需训练数据即可获得较高的分析精度；同时我们的UI分析技术可以自动在分析过程中收集待测应用数据，借助测试历史反馈数据，驱动基于深度学习和机器学习预训练模型的参数微调，进一步提高UI分析的效果。

其次，我们吸收了autoML的思想，通过自动化参数组合和检验选择出最适合于指定待测应用和待测机型适的参数组合。

此外，我们的技术对于UI能够进行基于图像语义和业务语义的分割，对于UI的不同组分进行异质视觉分析，进一步提升了UI分析的稳定性和健壮性。

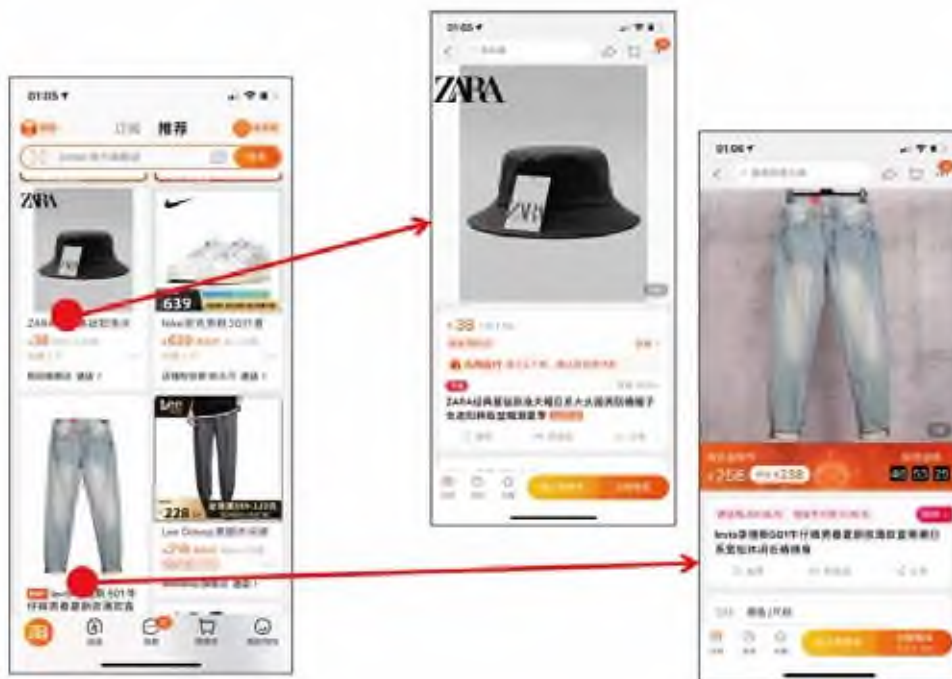


上图展示了我们的UI 分析技术的效果。我们在手淘核心场景上的控件定位，控件分类和语义分割准确度超过了95%，在UI的层次提取和业务逻辑规约上也实现了极高的精度。

探索过程遇到的问题和解决方案

在遍历和探索测试过程中常常会遇到这样三个问题：

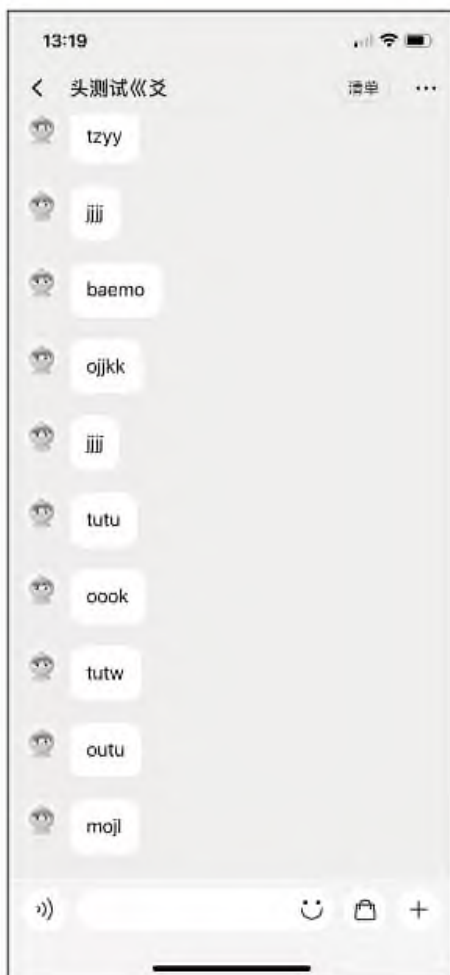
- 1.类似或者相同的业务场景重复测试，比如在手淘中下单的场景，虽然是不同商品，但是功能都是一致的，因此没必要对每个商品详情页重复测试。



2. 在场景交错的情况下，可能会出现循环的情况，即A场景可以点击元素进入B场景，然后B场景又可以点击某元素回到A场景。比如下图所示与商家聊天的场景和店铺首页场景间可以来回切换，这时可能会导致机器一直陷入循环做无效探索。



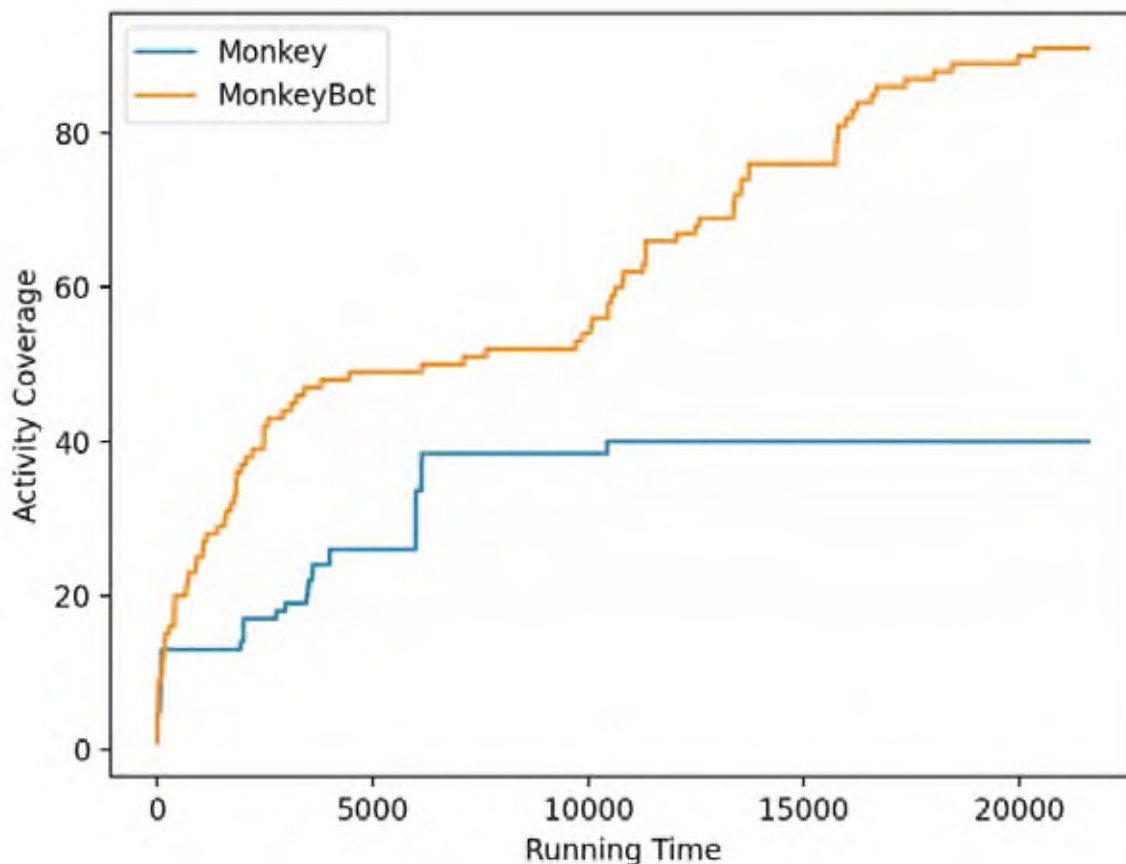
3. 在一些场景中可有效点击的元素非常少，很多元素是点击无反应的，此时如果一直探索的话会影响探索效率，如下图聊天页面，每条聊天信息都被识别成一个独立的元素，但是均无法点击。



我们的探索过程是基于强化学习完成的，针对以上问题做了很多探索策略的优化设计。对于重复探索和循环探索都会跟历史状态有非常大的关系，因此我们引入好奇心探索机制，增加对历史状态记忆的能力，促使代理人选择更容易触发新状态的动作。并通过深度广度平衡能力，决定是否继续在当前页面探索，还是回到之前的页面寻找其他操作路径。

效果展示

在手淘app测试中，设计后的探索策略对比Monkey测试随机策略activity覆盖率提升120%，有效功能路径覆盖提升2倍以上，大大提升了覆盖路径的有效性。



总结及未来展望

对于拥有海量用户或者错综复杂业务场景的应用，都会面临功能测试回归成本高且容易漏测的问题，如何在测试中全面覆盖用户操作路径成为质量保障的一大难题。

本文介绍了一套由淘系技术质量团队和北京大学谢涛教授团队一起合作研究的基于融合AI技术打造的智能探索测试方案，并展开分析了方案解决的一些技术问题和业务问题。

为打造一套通用的类人测试的工具，采用RXT机器人测试工具完成执行操作，然后打造一套纯视觉分析的解决方案。对于UI能够进行基于图像语义和业务语义的分割，在手淘核心场景上的控件定位，控件分类和语义分割准确度超过了95%。

通过图像特征，文本信息和UI的结构等信息来对UI状态进行建模，同时容许状态抽象的多粒度表达。多模态信息的使用充分的提高了状态抽象的准确性和可靠性，而多粒度的状态抽象则使得业务语义和多种测试需求的同步表达成为了可能，增加了智能探索测试的灵活性。

引入强化学习探索策略，结合探索历史和智能探索策略，解决了相同业务场景重复探索、探索陷入循环和界面无效点击等问题，对比随机策略activity覆盖率提升120%，有效功能路径覆盖提升2倍以上，实现更有效地路径覆盖。

未来在探索策略上还有更高的提升空间，我们希望通过状态抽象和探索策略能力的提升，机器可以实现在给定app内更高效探索，同时结合智能验证能力主动发现过程中遇到的bug，完全具备类人测试能力，实现全流程无人参与的自动化测试。

音视频与图像技术篇

• TaoAudio: 淘宝直播短视频音频解决方案

- 作者：吴雄琳（荣木）
- 出品：淘系技术

背景介绍

2019年是电商直播爆发的一年，被称为电商直播元年，2020年电商直播持续发展，越来越多的平台开始发力直播带货业务。淘宝直播作为电商直播第一平台，成为连接上亿主播和粉丝的桥梁。

直播传递信息的途径，不外乎声音和画面两种。其中声音作为“第一媒介”，需求主要包含以下两个部分：第一是音质，好的音质能让人产生“声临其境”的感觉，而声音刺耳，断续，音量过大或过小，都会让人听起来不舒服，从而影响购物体验；第二是互动体验，视觉互动已广为人们熟悉，但声音互动其实更加自然，比如主播通过语音控制发红包，上货，粉丝通过语音消息与主播互动，或通过语音连麦与主播沟通；除此以外，语音在安全管控方面也显得尤为重要，在防盗播，涉黄涉政方面，都可通过语音来识别异常。



在短视频上，声音的需求也越来越多。随着2020年底点淘，逛逛等新业务的推出，短视频成为内容业务发展的新引擎，为了帮助商家达人更加便捷的生产高质量内容，“亲拍”APP随即推出，大幅提高编辑效率，成为淘系商家，达人生产优秀短视频的有力工具。与此同时，“亲拍”APP的深度编辑功能还提出了大量音频需求，比如变速，混音，快速播放，语音降噪，自动字幕，语音旁白等等。



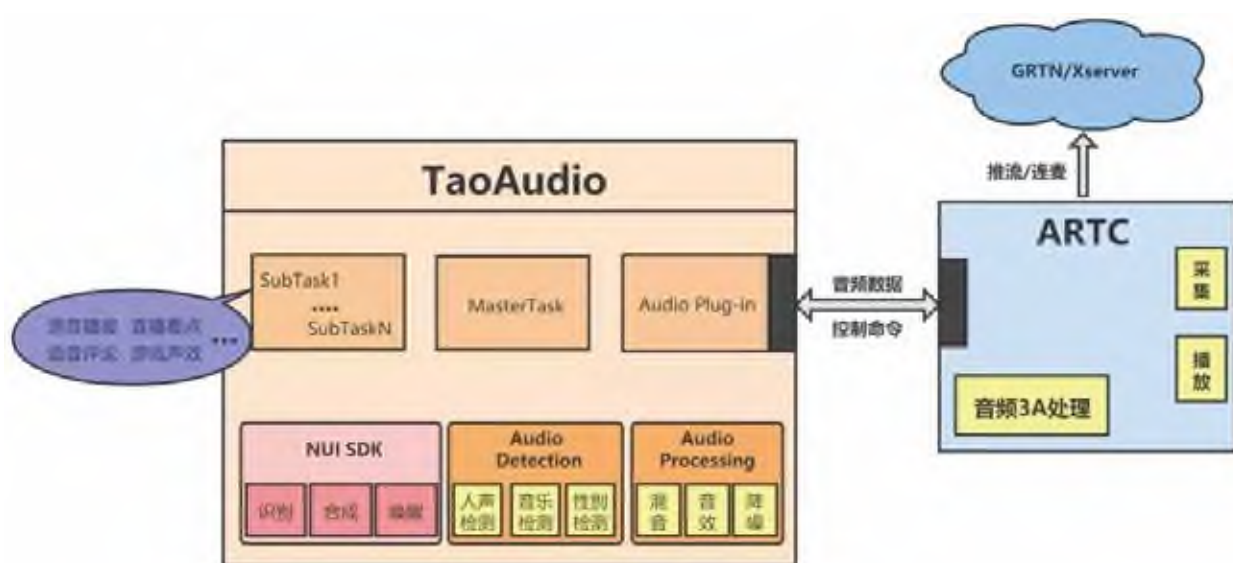
音频业务解决方案TaoAudio

TaoAudio是面向直播和短视频声音需求的完整解决方案。其分成三个主要模块，分别是音频处理，音频安全，语音交互。TaoAudio支持端云一体化部署，部分模块部署在端侧，比如音频编解码、降噪、场景检测、水印等，可充分利用端上算力，以低成本方式实现实时处理，另一部分模块部署在云上，比如语音识别，语音合成，确保算法的精度和效果。同时，TaoAudio还具备友好的部署方式，通过JNI和Object-C实现跨平台调用，上层只需配置模板和送入音频数据，即可源源不断的拿到处理后的声音或标签，实现极低成本的接入。



直播业务

直播业务对音频的三个核心需求是实时，安全，交互。面对众多的音频需求，如果只是提供算法原子能力给业务层调用，集成和联调成本会非常高，并且业务层不可避免会碰到一些音频专业相关问题，解决难度大，TaoAudio抽象了需求共性，实现了对实时音频流的检测和处理，如下图所示，在直播场景中TaoAudio起到连接上层业务和推流底层的作用，让业务音频需求和底层推流解耦合，业务音频相关的需求和问题全部收口到TaoAudio里面，这对提升音频类需求实现效率和加快迭代优化速度有非常大的帮助。



NUI SDK

NUI SDK提供了智能语音交互能力，包括：语音合成、语音识别和关键词检测等功能。

音频检测

音频检测模块实现了直播声学场景和声学事件检测功能，具体算法包括：人声检测、噪声检测、音乐场景检测、性别识别等，整体算法思想基于深度学习和信号处理相结合进行。

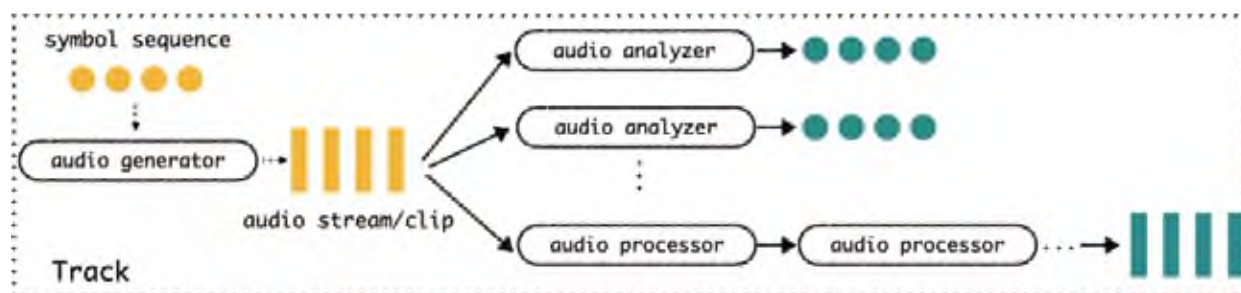
音频处理

音频处理模块包括声音自适应混音、语音重采样、语音变调变速，音效处理、语音降噪等。其中较为核心的是自研的智能降噪Alidenoise和音效处理Soundmod SDK，能够很好的满足上述需求。

短视频业务

在非实时通讯业务中，由于不存在实时音频采集和回声消除等场景，TaoAudio可以以一种更加单纯的形式存在，这种形式我们叫做智能数字音频工作站（intelligent digital audio workstation，以下简称iDAW）。如下

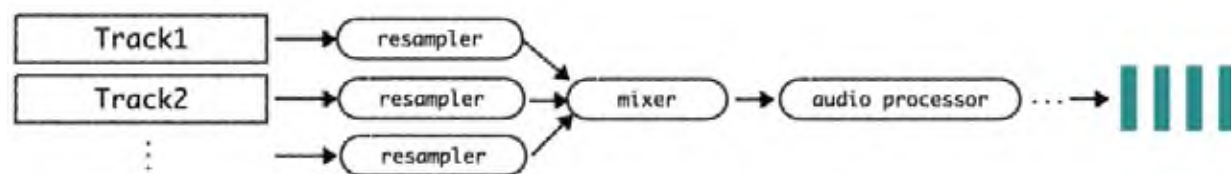
图所示，在iDAW里面，我们可以有多个音轨（track），每个音轨的源头（source）可以是一个音频片段（audio clip），一个音频流（audio stream），或者一个符号序列（symbol sequence）。每个track的结构如下图：



其中，黄色代表输入，绿色代表输出，虚线表示可有可无。每个track里面有三种围绕audio的变换。

- 1.一种是“符号-音频”的变换，可以理解成是audio generator，输入符号序列，输出音频，例如TTS便属于这种变换；
- 2.第二种是“音频-符号”的变换，可以理解成是audio analyzer，输入音频，输出符号序列，例如VAD、ASR便属于这种变换；
- 3.第三种是“音频-音频”的变换，可以理解成是audio processor，输入音频，输出音频，例如上述soundmod里面的所有变换，以及3A里面的ANS、AGC等。

每个track可以通过重采样，进行混音，输出混音后的音频，如下图：



在TaoAudio里面，一个iDAW对应一个TaoAudioWorker（TAW）实例。整个TaoAudio的基础架构是用C++实现，目前支持的audio generator有NUI SDK里面的语音合成，audio analyzer有NUI SDK里面的语音识别、唤醒词识别和文件极速转写，还有我们整个音频检测系统，audio processor有soundmod和alide-noise等。

TaoAudio把恰当的连接能力和处理能力提供给调用者，不同的业务方能各取所需，通过TaoAudio定制出自己的整个音频工作链路。例如在短视频生产工具中，利用TaoAudio可以做到无缝的对接几乎所有音频相关的业

务，比如自动生成旁白（TTS）、字幕提取（ASR）、自动去除空白（VAD）、原声增强（soundmod）、变速变调（soundmod）、降噪（alidenoise）、声音美化（soundmod）、背景音乐生成（chord-arranger）等等，每一个功能都可以以track为单位进行添加和连接。

TaoAudio音频核心技术

TaoAudio作为一套音频业务解决方案，底层核心技术主要涉及两块：一块是音频处理技术，另一块是语音交互技术。其中，音频处理包括：智能语音降噪、智能声学场景检测、音效处理、回声消除，以及音频指纹和音频水印技术等；语音交互技术主要包括：语音识别、唤醒、语音合成、敏感词识别等；

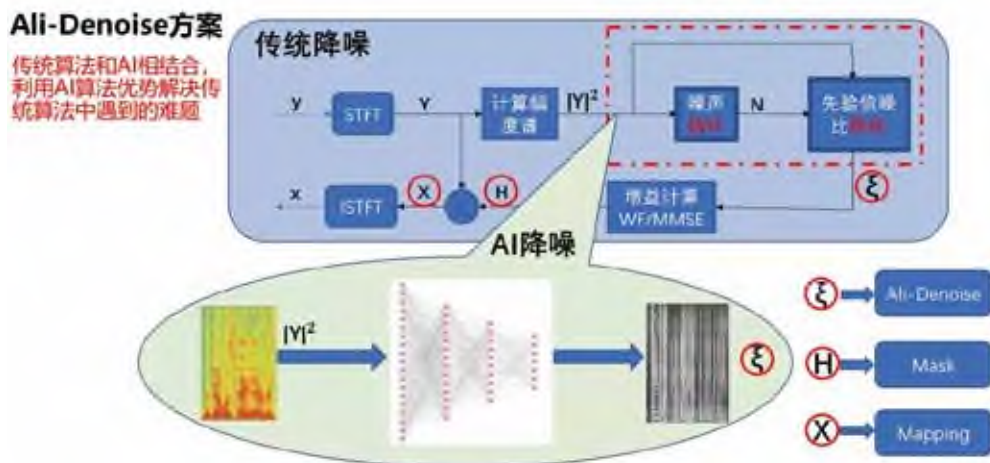
音频处理技术

智能语音降噪

传统语音降噪算法速度较快，计算消耗较低，能在多种多样的低端设备上运行，但面对复杂且多变的非平稳噪声和低信噪比环境，传统方法的效果不尽如人意。基于学习的、数据驱动的降噪算法，通过对大量数据的学习，在真实的噪声环境中展现出了较为突出的优势，取得了较好的效果。

但是这类“基于学习的”方法，由于参数多、模型大，较为复杂，因此可解释性欠佳，稳定性不易受控，泛化能力不易保障，缺陷不易排查。这些问题的存在，导致基于学习的方法常被称为“无法观察”且“不易调整”的“黑箱”。同时，基于学习的算法虽听感效果出众，但相比传统算法，复杂度偏高，运算速度较低、电量消耗较多，更容易导致硬件发热、系统降频、程序卡顿等问题。

为将最好的音频体验带给用户，淘系技术音视频算法团队基于深度学习和信号处理相结合的思路，在反复的研发、试错、创新中，针对降噪的效果、质量、算法的速度、能耗、延迟与泛化的稳定性等诸多方面，应用了一系列技术，对模型结构、框架、约束等进行了研发改进，最终研发了智能语音降噪算法Alidenoise。



Alidenoise技术特性

- 1) 传统信号处理方法与深度学习类方法相结合，兼具可处理非平稳类噪声和算法复杂度低的特点
- 2) 在神经网络的设计上，结合训练目标，以人声的语谱纹路作为主要学习对象，噪声泛化性强语音保真度高
- 3) 采用Cache buffer技术，实现流式处理
- 4) 轻量小模型，支持移动端实时增强，覆盖低、中、高设备
- 5) 灵活的网络模型配置，支持降噪算法延时可调

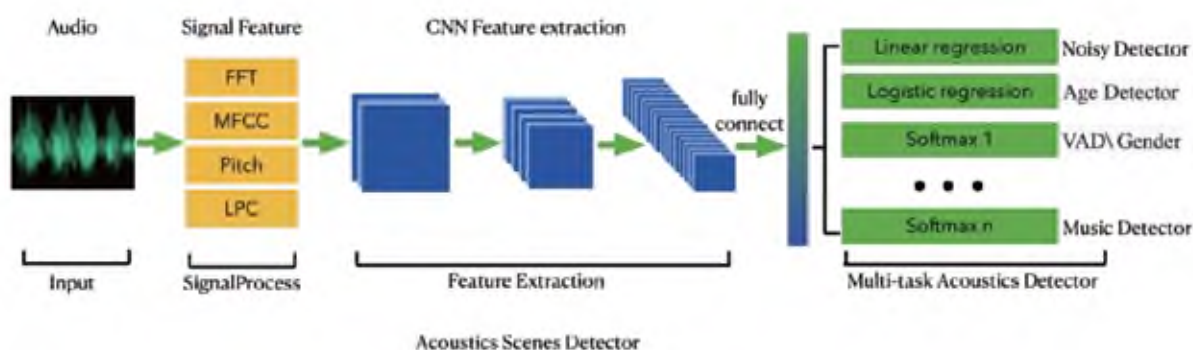
智能声学场景检测

淘宝直播、短视频的多媒体需求日益增长，在基础通讯链路与上层业务需求中，声音与音频需求层出不穷。

需求的增长与基础技术的发展成为了主要矛盾。因此我们提出多任务的智能声学场景检测项目，并逐步实现。

- 1) 信号处理：为了提升声学场景检测的泛化性，我们对音频进行声学信号处理的特征提取，比如fft频域、mfcc、lpc系数、基音周期等经典有效的特征，来表征声音信号，同时也对声音向量有效降维。
- 2) 端侧AI：端上轻量化模型，是声学场景检测的主要关键之处，我们借鉴了经典的MobileNet、ShuffleNet中轻量化模型技巧，采用矩阵分解、分组卷积、小卷积核等卷积优化，全局池化替代繁重的全连接，同时尝试空洞卷积对于模型轻量化的优势。为了在端侧取得更高的适配能力，模型使用了量子化来进一步降低算耗，并使用短时记忆单元提高上文信息量，在提速的同时保障算法效果。
- 3) 多任务联动：在声学场景自适应的多任务中，我们共享卷积特征权重，有效增加任务的泛化性。同时在相关任务设置一级分类与二级分类关联设计，不断的有效提升二级分类的准确度，比如是否人声检测为一级分类、性别检测是二级分类，两者联动可以让性别检测更加准确。
- 4) 能力迭代扩容：随着场景检测算法与上层玩法之间的相互促进与丰富，检测类目的需求会动态地、甚至敏捷地变迁。为了在此业务场景下保证检测能力与需求俱进，我们将借助迁移学习与增量学习技术，敏捷升级识别能力，快速反应逐步扩充的检测需求。
- 5) 业务承载：多任务的智能声学场景检测，承载着两大类业务。第一类淘宝直播通讯场景的自适应增强业务，比如检测不同的嘈杂程度、是否音乐来进行配置不同的算法参数，利用不同性别年龄检测来配置不同的美声需

求。第二类是业务需求，比如轻直播的人声检测需求，空镜检测的人声检测需求，音乐可视化的音乐检测需求等。



音效处理

我们有一套完整的从音频到音频的变换工具Soundmod SDK，它支持对音频进行实时的和离线的处理，Soundmod支持的处理包括了最常用的“三大件处理”（动态压缩、均衡、混响）、变速不变调、变调不变速、男变女女变男，也包括了其他常用的效果，比如合唱效果、颤音效果、延时效果、哇音效果、声码器效果等等：

- phaser vocoder类音效：time-stretching、pitch-shifting、vocoder、robotic、whispering
- delay类效果：delay、chorus、flanger、vibrato
- dynamics类效果：compressor、limiter
- filter类效果：equalizer、autowah、phaser
- modulation类效果：ring-modulation、tremolo
- spatial类效果：reverb

智能语音交互技术NUI

- NUI SDK (Natural User Interaction)是阿里巴巴达摩院语音实验室自主研发的语音全链路方案。
- 方案聚合端和云的核心算法能力，包括信号处理，语音唤醒，语音识别，自然语言理解，语音合成，声纹识别等，具备完整端云一体语音能力。
- 方案通过能力可插拔组合的方式，实现一套SDK满足不同产品的语音需求。
- 方案支持iOS/Android操作系统，集团内外累计激活设备3.7亿+。

产品功能		Android	iOS
√: 支持, 默认内置该功能 X: 不支持该功能 Δ: 支持, 默认不内置该功能, 可定制或添加	离线唤醒	√	√
	主唤醒词定制	Δ	Δ
	离线命令词识别	Δ	Δ
	唤醒词二次确认	Δ	Δ
	声纹识别	Δ	Δ
	属性识别	√	√
	在线ASR	√	√
	离线ASR	Δ	Δ
	在线NLU	√	√
	离线NLU	Δ	Δ
	在线TTS	√	√
	本地TTS	Δ	Δ

淘宝直播场景作为丰富的语音技术使用场景, 在应用内存在多种对语音能力的使用。例如“CRO敏感词检测”使用了“离线命令词识别”, “评论播报”使用了“在线语音合成”, “直播看点”使用了“在线语音识别”, 以及后续还可能上线的其他功能会有更多能力的扩展。这就要求我们的SDK具备多实例并行的能力, 能够使得不同调用业务之间可以并行不受影响。因此在本次淘宝直播和达摩院的共建中, 语音实验室完成了NUI SDK的全面升级, 将原有的面对交互场景的SDK扩展为更加灵活, 可支持多种原子场景独立使用的版本。

直播场景是算法技术使用非常密集的场景, 信号处理, RTC, 敏感词检测, 视觉算法等等都在这个场景下全速运行, 这意味着对于CPU的消耗非常大, 同时App对包大小有着严格要求。语音实验室全面升级命令词检测技术方案, 采用MNN推理框架, 并同MNN进行了深度优化, 在实时率上提升明显。通过“PAI模型压缩”技术将唤醒模型量化到INT8计算, 进一步减少模型大小, 相对对上一代方案减小50%。再者通过直播端侧的模型下发链路, 能够将模型进行云化, 进一步减小App包大小。

应用案例

直播评论和提示音播报

语音播报功能给主播提供的是直播间用户意图的声音提醒能力, 该功能是通过语音, 将粉丝进入直播间、粉丝评论和粉丝连麦申请等行为在主播手机中播报出来, 目的是提醒主播当前直播间的实时动态, 做好接待客户的准备。语音播报功能直接涉及到的语音技术点, 主要包括: 文字转语音TTS、语音混音、信号重采样以及语音数据流管理和控制等。

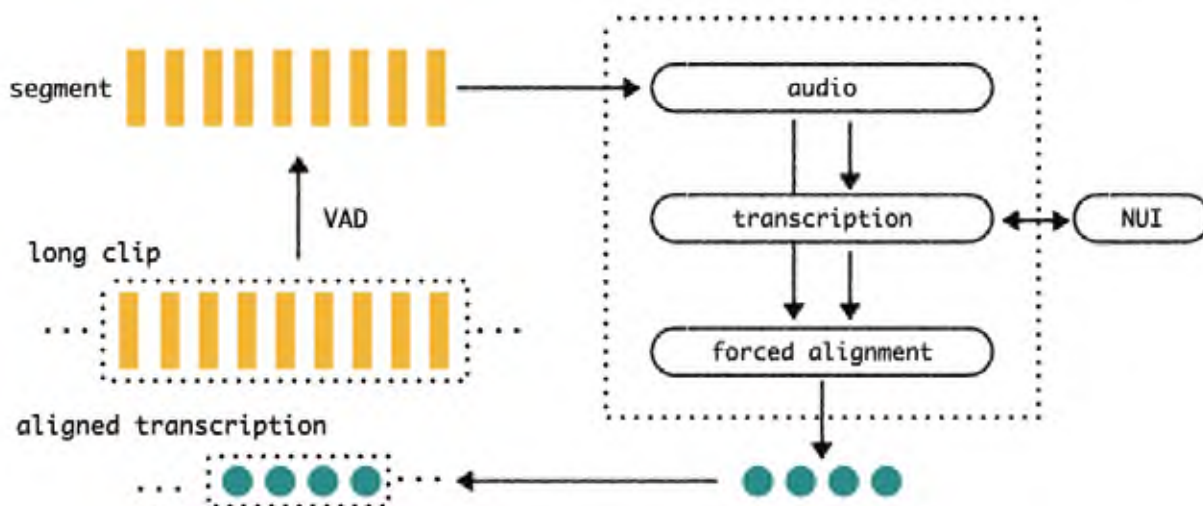
除此之外，播报声音要在一边推流的情况下—边播报，所以还涉及到播报声回声消除，防止播报的声音被粉丝听到。另外，语音播报具体实现在业务上层，回声消除是音频底层技术，如何协调好两者之前的关系，相互配合好达到好的效果体验，具有一定的挑战性。

TTS是语音评论播报的基础，评论在主播端播报之后会被推流的麦克风采集，如果不进行处理会导致直播间所有粉丝都听到播报声。消除本地播报的声音，可以使用回声消除技术，ARTC底层有回声消除算法，但是不支持本地播放消除，经过对ARTC层系统的改造，实现了播报声消除的功能。

但是，当主播讲话时进行播报，会形成回声消除“Double Talk”抑制过大问题，导致主播声音损伤严重。为了解决该问题，从两个角度出发，一个是播报时优化回声消除算法，另外一个通过主播说话智能检测控制播报声音减小或者不播报。最终通过算法优化，解决了播报消除导致主播语音损伤的问题，使评论语音播报功能体验更佳。

短视频智能编辑

在短视频领域，口播类的短视频在商品介绍的短视频里面占了绝大多数。口播编辑能让短视频生产者很轻松的剪掉不必要的长停顿，多余的字句，以及字与字时间的琐碎的停顿。这个功能的核心技术是语音识别和字幕转写对齐的服务，如下图所示：



一段长音频通过在端上的分片分成以1分钟为单位的音频切割文件（segment），传输到服务端，服务端拿到文件之后先进行一次极速文件转写（transcription），这个转写的过程需要与NUI服务进行交互，拿到结果转写结果之后，把结果和原音频一起送到强对齐算法（forced alignment），输出带时间戳的逐字转写结果（aligned transcription）。使用了该功能之后，能让这一类口播视频的生产达效率大幅提升。

短视频语音降噪

在口播的短视频里面，人声是前景声，其他声音一般都是背景声。背景声有时是有用的声音，比如背景音乐，或者和讲解场景或物品相关的关键性声音；有时是无用的声音，比如商场内的嘈杂声音，或者马路上的噪音。业务上通常有一种需求是去除无用的背景声音，以让前景声音变得更加突出，让短视频里面所呈现的信息更加清晰。淘宝音视频算法团队自研的智能降噪算法Alidenoise已经集成到TaoAudio中，经过和竞品降噪能力对比表明Alidenoise降噪效果优于竞品，且处理的实时率能达到1%，目前智能降噪能力已经输出给短视频编辑业务使用。

直播看点ASR和推荐

为了提升直播时宝贝转化的效率，优化用户观看直播时的购物体验，淘宝直播向所有商家主播、达人主播，提供了“直播看点”的功能。直播看点有两种形式，一种是主播手动打点，另外一种智能打点。智能打点主要根据直播视频中的目标识别进行自动打点，主播口播为智能看点提供了另外一个维度的信息，使用语音识别技术把口播语音全部转成文字信息，再使用语义理解技术提取主播所介绍的商品信息。

直播看点ASR完整的功能实现涉及到语音识别、信号重采样、语音降噪等技术，其中语音降噪的选择对识别的准确性有较大的影响。除了把声音内容作为打点的特征，还可以从声音类型的角度出发，检测当前主播是否在说话、直播间是否有音乐，或者处于哪种声学环境，把声音作为一个维度信息进行直播间的推荐。当前TaoAudio已经具备上述技术能力，能够快速辅助业务实现对应的功能。

直播语音评论

评论区作为直播间的核心互动功能，可以通过评论提升直播间互动率和用户直播间停留时长。直播评论如果只有文字的话，形态比较单一，语音评论可以让粉丝在观看直播的时候用语音进行评论，使粉丝和主播的互动更加有趣。

语音评论技术链比较长，涉及到客户端语音采集、编解码、播放、语音识别转文字和服务端语音存储、内容审核，在主播端，主播播放出语音评论，还涉及到声音的控制和回声消除算法。语音评论链路上所涉及到的音频相关的需求，TaoAudio均能很好的满足。



短视频旁白生成和音效处理

短视频里面需要有旁白，这些旁白可以是机器生成的人声，或者真实录制的人声。很多时候用户不希望自己的真实声音出现在短视频中，便会选择生成的声音作为旁白，或者对自己录制的声音进行各种音效处理以隐藏自己的真实声音。

在淘系的短视频制作工具MAI编辑器和亲拍APP中均都接入了TaoAudio的文字转语音的功能，这些功能可以很方便的生成不同音色、不同语速、不同语调的人声。在有人声的基础上，用户可以很自由的对人声进行各种变换，包括变速、变调、变声、美声等等，这些技术均可通过TaoAudio内集成的Soundmod实现。

总结&展望

总的来说，当前TaoAudio作为内容平台的音频解决方法，支持了多种音频相关的业务需求，从刚开始遇到的各种问题，到现在问题逐步的收敛，SDK的功能越来越稳健。淘系技术音视频团队会继续丰富算法能力，比如AI结合的语音增强、智能音频检测、短视频自动配乐，音乐理解和生成，同时达摩院语音技术团队在不断优化升级语音交互性能，例如和MNN共建的端上唤醒技术、离线语音识别等也都相继要上线来满足直播业务的需求。

未来，TaoAudio将基于现有的功能继续打磨，保证稳定性的同时进一步提升体验，并且将朝以下几个方向继续发展：

- 1) 算法能力扩展：丰富直播互动音频类算法，从音频层面支持更多业务玩法和功能需求
- 2) 端侧AI：结合传统信号处理和AI，提升算法效果，基于MNN框架提升算法效率，实现大部分音频AI算法端上部署，比如端上ASR技术，智能VAD技术，智能PLC技术等；
- 3) 云端一体：端云相互配合，实现云端部署复杂算法和功能的能力

• 淘宝直播低延迟架构演进和实践

- 作者：ECUG 活动组
- 出品：淘系技术

本文根据杨宽（阿里巴巴淘系技术 音视频技术专家）于 2021 年 6 月 26 日举办的 ECUG Meetup 第 1 期 | 2021 音视频技术最佳实践 · 杭州站上的分享整理而成。本文将从传统直播技术痛点、低延迟架构演进、互动体验升级和关键技术四个方向展开。

传统直播技术痛点



大家看，这是线上的一些直播场景。

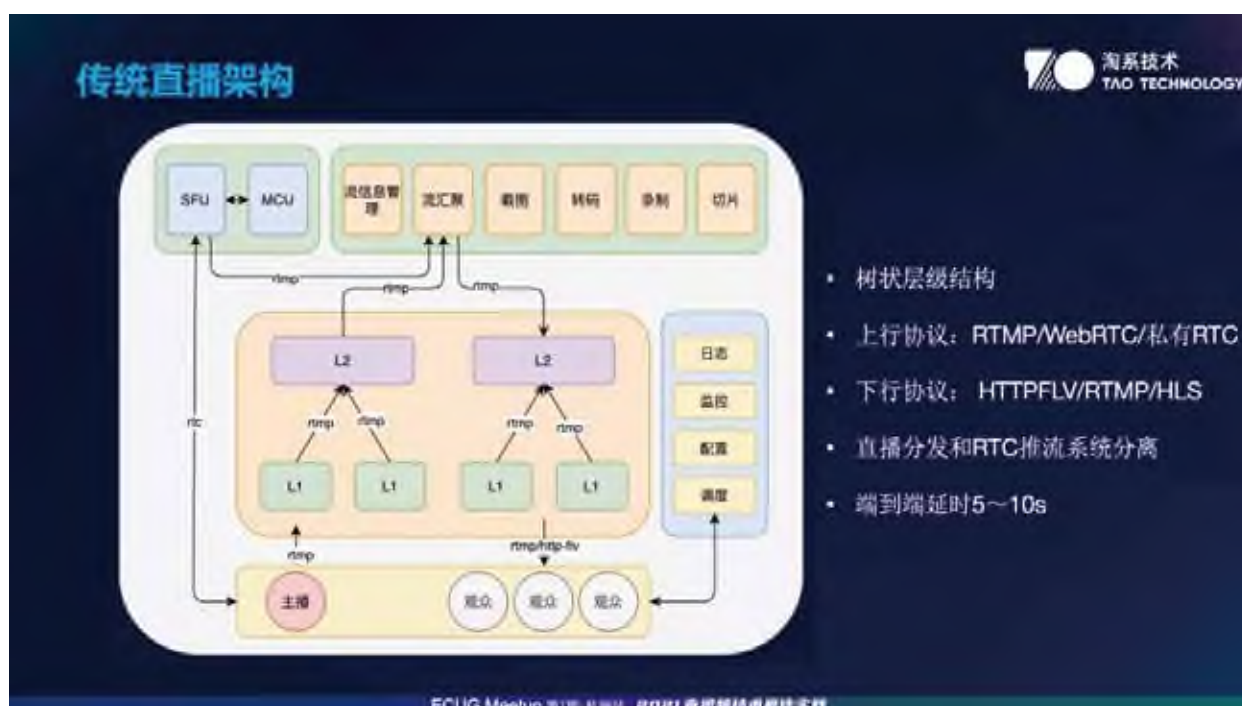
第一张图主要是服饰的场景，下面有观众与主播的评论交互。在传统直播的场景下，可能观众看的是 5-10 秒以后的画面，当观众评论提问的时候，主播可能已经不介绍这款商品了，所以会有一个延迟的 Gap，交流就比较困难。

第二个场景是珠宝的场景，介绍完镯子以后可能会换下一款，但是观众可能看的还是上一款镯子。

传统直播的主要痛点有三个：

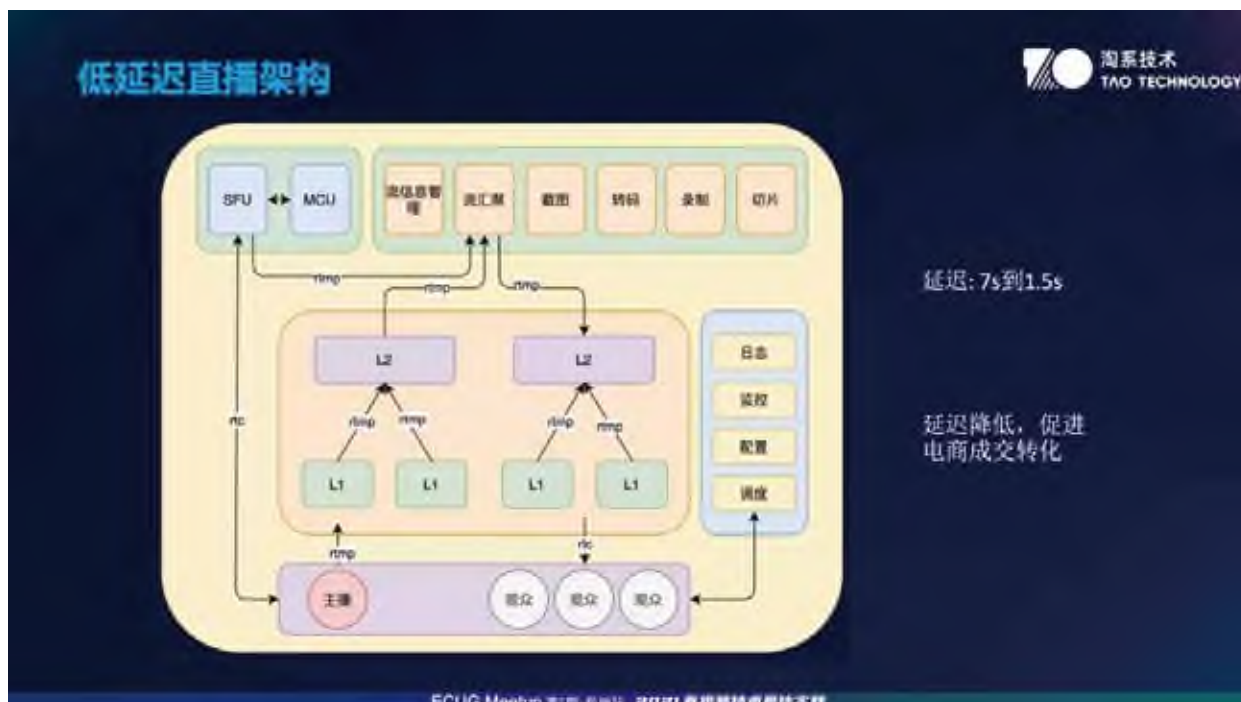
- 1.延迟高，5-10 秒的延迟；
- 2.主播和观众互动不及时；
- 3.直播和连麦场景切换复杂。

低延迟架构演进

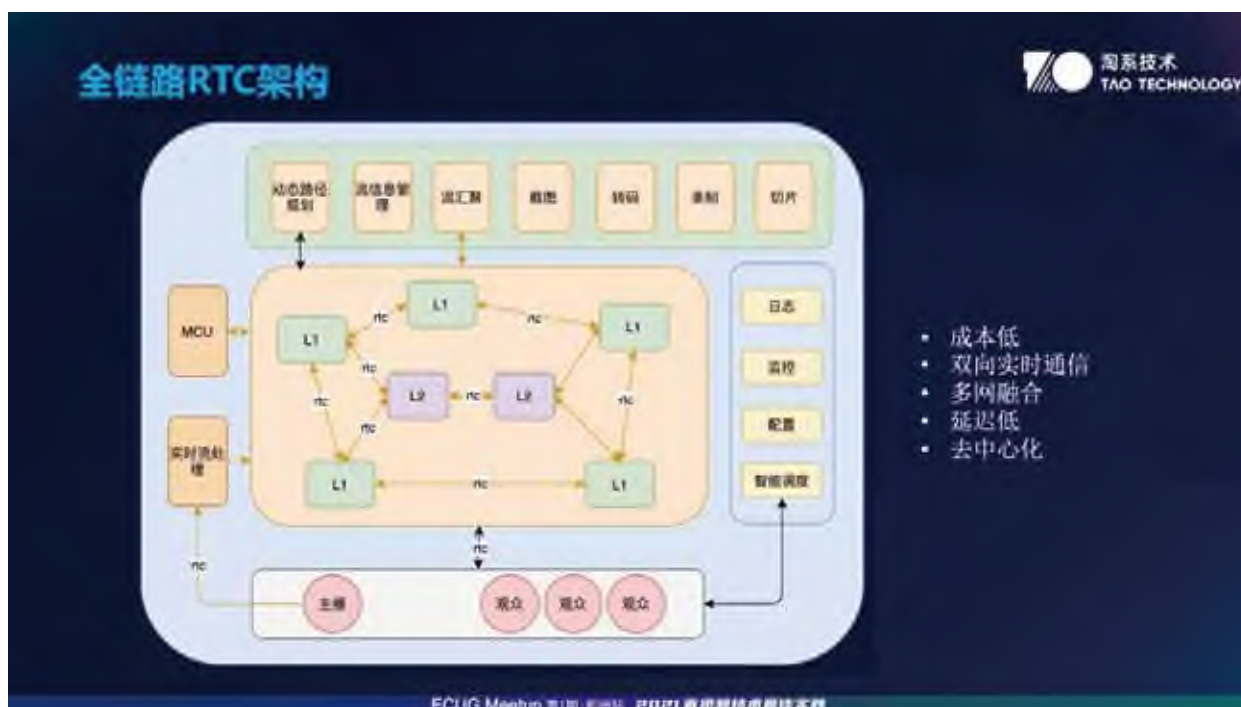


这是传统的一个直播架构。中间是CDN的分发网络，最左边是自建SFU 和 MCU 集群。最右边是支持的一些系统，比如说日志、监控、配置、调度，最下面的图是推拉流的 SDK。最上面是一个直播中心，有截图，转码，录制，切片等服务。

传统的直播分发网络是一个树状的结构。因为直播的量很大，树型结构分发可以控制成本。上行协议主要是 RTMP/WebRTC/私有 RTC，推到自建集群，然后通过rtmp推到CDN。下行协议主要是 HTTP-FLV/RTMP/HLS。



这是我和CDN共建做的低延迟直播架构的改造, 改造的点主要是 CDN 和观众播放器之间采用了 RTC 相关的技术, 把延迟从 7 秒降到 1.5 秒左右。在业务上不仅延迟降低了, 方便观众和主播间的交流。而且在线上分行业AB显示, 在促进电商成交转化方面也取得了不错的效果。



这个架构大概是我们 2019 年开始和 CDN，视频云，企业智能，XG 实验室共建的架构，也是目前正在开始大规模使用的。可以看到，中间 CDN 框架那边全部走的是 RTC 的链路，不再是树型的结构，是去中心化的架构。L1 和 L1 之间可以互通。如果 L1 和 L1 之间的通信有问题，可以走 L2 中转。MCU 合流服务可以像客户端一样直接连接到 RTC 的网络上，把需要处理的流拉下来，处理完之后再重新推给 RTC 网络。

主播通过 RTC 可以和网络直接通信，如果需要它做一些实时流的处理的话，比如说加一些 AI 的特效，可以通过实时流处理，处理完之后直接推给 RTC 整个网络。观众的话，也是通过 RTC 链路直接和这张网交互。

整个全链路 RTC 架构有几个比较好的点。它是一个直播、通话、连麦、会议共用的一张网。不同的业务高峰使用的时间段不一样。比如说直播，一般晚上观看的人数比较多。会议基本上是白天开会。这样不同业务就可以错峰地使用这张网。

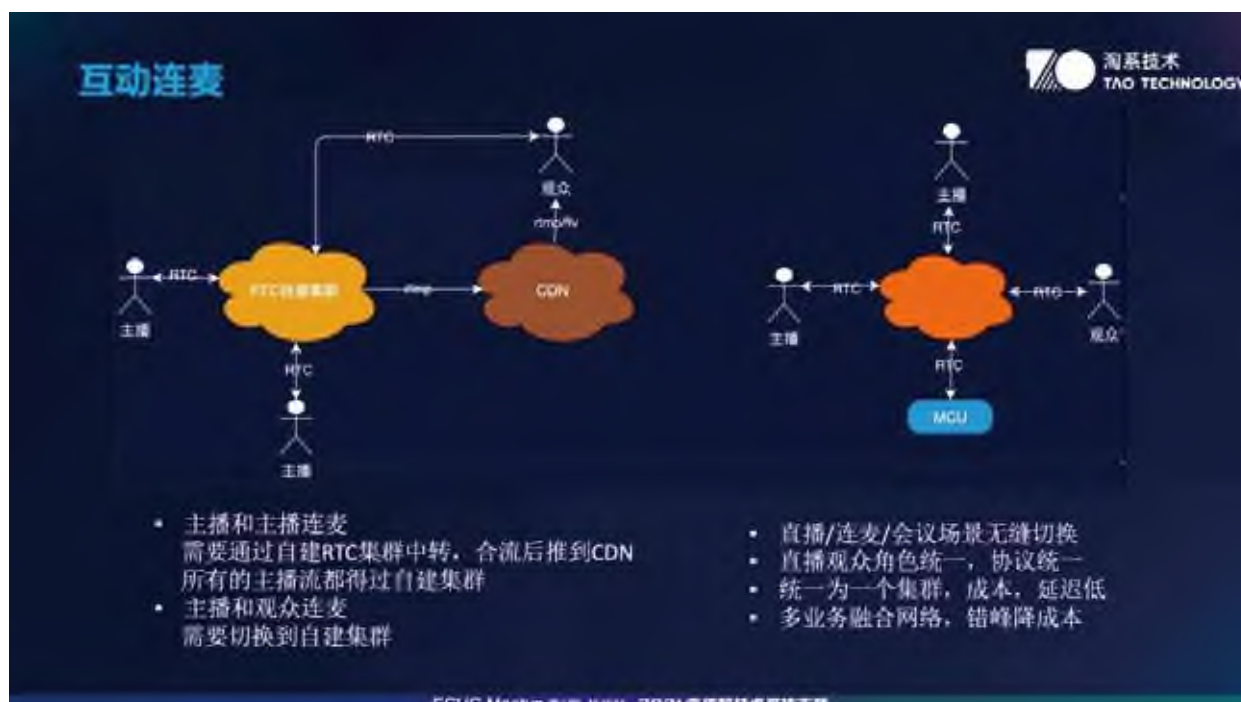
因为 CDN 一般是根据每天峰值带宽来计费，这样白天可以跑通话会议的业务，晚上会有一些直播的带宽上来。通过错峰，不同的业务错峰来降低整体的成本。双向实时通信这块儿，因为 RTC 既可以推流也可以拉流，在同一个链接里可以走多路流，这是不限制的。比如说，推一路流，拉十路流，都是可以的。

互动体验升级

互动体验升级，对直播场景来说，主要是两点。

第一，**提高互动效率**。消费者通过直播间和直播交流的时候，原来老的系统观众看到的画面是 5-10 秒前的画面，所以主播和观众互动的时候时间点对不上。升级以后的延迟优化为 1 秒左右，使消费者和主播互动更及时。

第二，**指标优化**。从延迟上说，可以做到 1 秒以内，在通话会议场景可以做到 200 毫秒左右。秒开率相对于原有的传统直播系统能提升 32%，卡顿率降低 79%，卡顿 vv 降低 44%。



这里要讲一下互动连麦直播原来的架构和目前架构的区别。

可以看到，原来的架构主播是通过 RTC 自建集群来转发 RTC 的流，保证低延迟，然后连麦。观众是通过 CDN 来拉流观看，中间走的是 rtmp 链路。如果是主播和主播之间的连麦，基本上是主播自己拉到对方的流，然后合流推到 CDN，然后再分发给观众看，这样的流程。如果是主播和观众连麦，观众本来是在 CDN 这条链路上。在传统的链路上，中间有一个延迟差，这个延迟差大概是 6-7 秒左右。通过 RTC 发起连麦的时候，通过 RTC 来连到自建集群，然后和主播互相通信的时候，画面是有延迟差的，这样体验就非常不好，有一些等待切换的逻辑在里面。

在新架构下，可以做到自建集群和 CDN 是完全融合为一体。也就是说，CDN 内部也是走的全链路 RTC，不管是主播和主播的通信，还是观众和主播的通信，完全走的是 RTC 的链路。RTC 链路可以通过一些传输优化保障，还有一些其他技术的优化，能够保障它所有的体验是统一的，比如说延迟、卡顿、流畅性等等。

下面是一个 MCU，就是合流服务器。如果主播或者说观众的设备是低端机的话，它的性能不太好，所以需要通 MCU 来合流，这个合流服务器也是类似于客户端的方式来接入这张网。然后通过把流拉下来，合完之后再推出去，因为我们延迟优化的很低，基本上能做到无感的切换。

这个架构模糊了观众、主播以及其他一些服务的角色。作为 RTC 的这张网来说，其实都是一个统一的接入者角色，既可以推流，也可以拉流。协议的话，都统一为私有的 RTC 协议。我们 RTC 的私有协议可以做到一个

RTT 就可以快速地拉流建连。场景这块儿，目前直播场景、连麦场景、会议场景以及通话场景，可以做到无缝切换。通过配置系统，针对不同的场景会启用不同的策略。

还有一个好处是，从自建集群加CDN 统一为了一个集群，节省了自建集群的成本。整体的延迟也是比较低的。业务融合网络，刚才介绍了一部分，不同的业务可以跑在同一张网上，因为业务主要使用的时间段不一样，我们可以通过错峰使用来降低整体的带宽成本。

关键技术



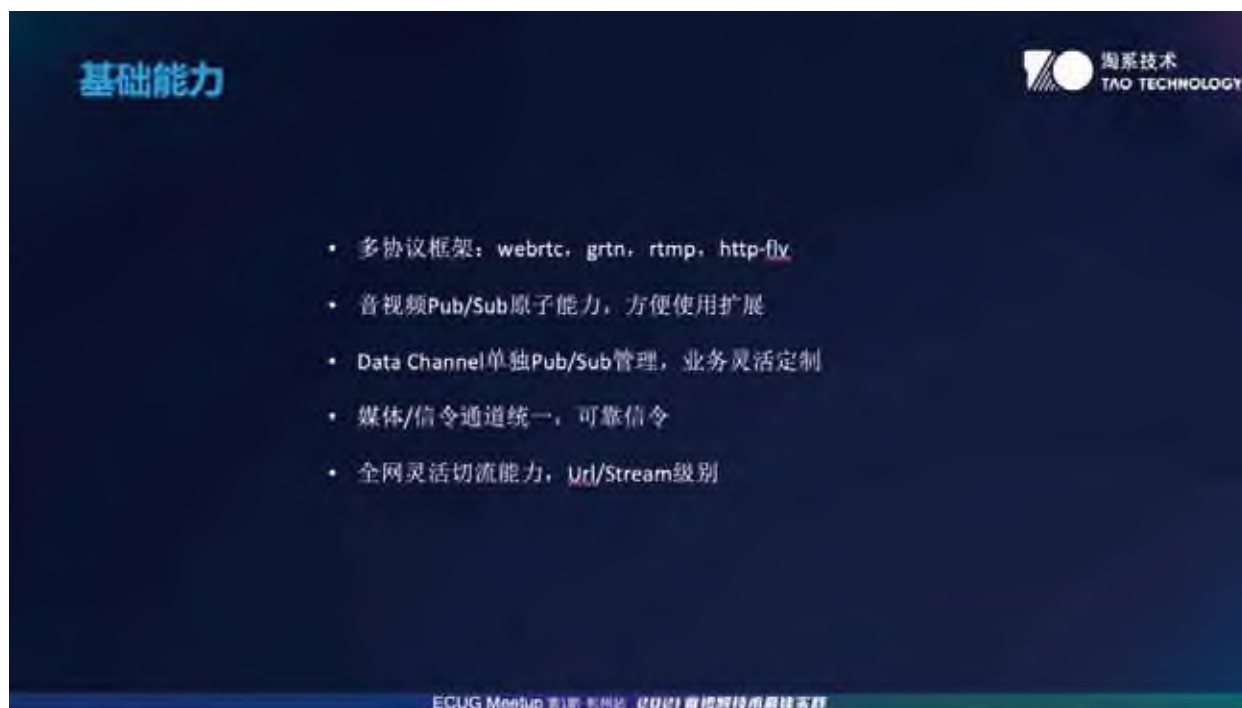
我们这边和CDN是共建的关系。RTC Module 提供RTC相关的核心处理能力，以模块化的方式嵌入到系统。

内部的模块大概有以下这些：RTP/RTCP、BWE、QOS、PACE、PACKER、trickle/jitter、frame buffer、SRTP、setting等等。BWE，主要是一些拥塞控制算法，QOS 这块会有一些 QOS 的策略，像重传，SVC，FEC以及大小流相关的。trickle/jitter 主要作用是去抖和组帧。PACKER，有一些场景需要从视频帧或者音频帧打包成 RTP格式。SRTP 主要是提供加解密的模块。目前支持grtn私有协议和webrtc标准协议的交互。

在管理层，主要是有以下四种：

- 1.回调管理
- 2.订阅管理
- 3.session 管理
- 4.推拉流管理

回调管理在系统和RTC 核心处理层之间，通过一些回调函数来处理它们之间的交互，做到这两层之间的隔离。



基础能力总结一下，主要是五点：

第一，多协议的框架。支持 webrtc、grtn协议。通过grtn私有协议接入的话，可以保证整体接入效果。还有其他一些私有扩展功能，比webrtc标准的交互效率会高很多。rtmp 和 http-flv 在这块儿上也是支持的。

第二，音视频 Pub/Sub 原子能力，方便使用扩展。

第三，Data Channel 单独 Pub/Sub 管理，业务灵活定制。Data Channel 目前主要应用在控制信令和一些云游戏的场景。比如说观众和主播会通过 Data Channel 通道发一些控制信令，控制一些摄像机的动作或者说游戏里人物的控制。设计的时候也考虑了它的灵活性，也是提供单独的 Pub/Sub 的能力，可以单独使用。也就是说，如果在一个 URL 下只单独做一个数据的通道，这也是没问题的。Data Channel 通道的特点，我们会提供单独的传输保障。延迟可以控制在全国范围内大概 100~200 毫秒左右。如果是同一地域的话，基本上是一个 RTT。线上的话，平均大概的 RTT 是 20 毫秒左右。

还有一个运营场景是一些直播间的消息，如果通过传统的消息系统基本上是秒级订阅，或者是推拉的模式。我们现在可以做到百毫秒延迟以内。业务这块可以灵活定制，比如说可以单独挂消息通道的服务器，通过这个服务器各个端可以单独订阅这个消息通道。做一些业务的活动，还有一些实时消息的下发，可以做单独的灵活定制。

第四，媒体和信令通道统一，可靠信令。信令我们会保证它的可靠到达。大家都知道，UDP 在公网上是很容易丢的，所以我们有一些保证它快速可达的策略，会有一些快速的重传逻辑以及冗余的逻辑。

第五，全网灵活切流能力，Url/Stream 级别。大家都知道，连麦的场景下，主播推直播间的流，他和别人连麦的时候，原来的方式是走 RTC 的自建集群，RTC 自建集群会处理切流的动作。因为所有的主播流都会走自建集群，他在切流的时候，因为数据源可控，可以做到切流的时候不会影响后面所有的播放。在全链路 RTC 的时候，其实是一个分布式的系统，主播和主播要连麦的时候，接入点不同，整体是分布式的系统，切流能力就会保证他在切换过程中，能够保证全网的观众在同一时刻切到合流的画面。可以做到 Url 和 Stream 级别，也就是说不同的 Stream 也可以自由切换。



传统的直播链路中间是走 TCP，因为 TCP 在内核层，系统会提供一些策略，定制优化很难。我们现在用全链路 RTC，用的是UDP，UDP是不可靠的。全链路 QoS 策略这块就显得很重要。

音视频系统有一个特点，从采集到前处理到编码，到传输然后到解码、渲染，其实是一个串行的过程，中间任何一个环节有问题都会影响整体的体验。

分阶段延迟优化我大致说一下。

做延迟优化的话，如果只是一个简单的场景，比如说一些数据量比较少的场景，这是比较简单的，但是我们的用户量，包括主播的用户量、观众的用户量，每天上亿的观看，如何保证全网平均延迟降下来，这个挑战是很大的。

延迟这块我们是这么思考的：直播系统主要分为推流端和播放器，中间走的是传输网络，需要针对每个阶段单独优化。传输网络部分，在网络比较好的情况，延迟还是比较可控的，如果网络不好，会启用一些 QoS 的策略。采集、前处理、编码，编码这块又分硬编、软编，还有一些自研的编码器。

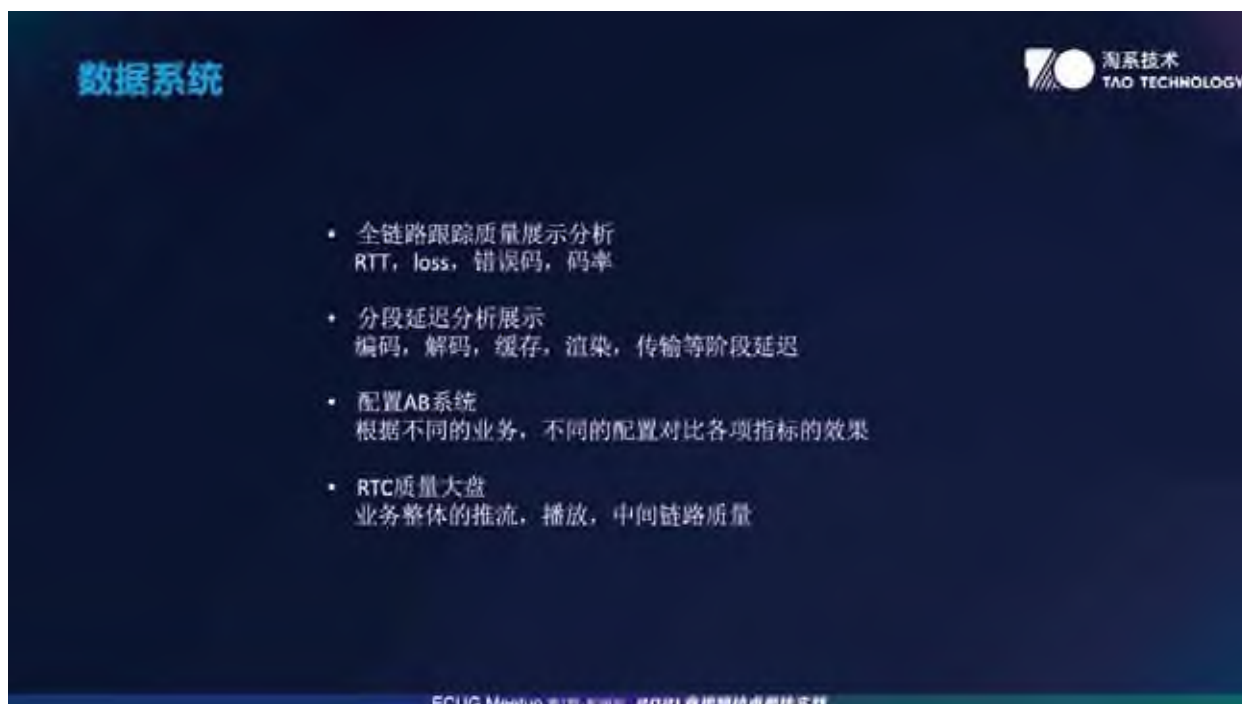
发送缓存，也会动态的有一些策略。接收侧的话，延迟比较大的主要是接收缓存，接收缓存这块儿，如果和上行的网络之间有丢包、抖动，接收缓存都会动态调整，把整个延迟加上去，为了对抗前面的一些问题。解码这块，包括解码速度、解码缓存以及不同的解码器，比如说硬解、软解之类的。还有一些后处理，解码数量后处理的一些算法处理延迟。最后是渲染的延迟。

不同的平台是不一样的，IOS、Android、PC。比如IOS平台前处理、编码是怎么样，大致的延迟分布是什么样的，都要有数据报表，在线上采集出来，去做针对性的优化。Android，PC 也是同样的。

每个阶段会有针对性的优化，比如说编码这块，我们会和算法团队一起优化，也会分不同平台进行编码部分的处理。编码这块儿也会放在线上，根据数据埋点系统分不同的平台，分不同的编码器，不同的版本做展示。

第二点，中间链路部分，主要是应用一些 QoS 的策略。还有一些调度，调度是和 CDN 合作的。规划出根据不同链路之间的网络质量、成本，规划出一条最短的传输路径。是质量、成本的一个综合策略。再加上 QoS 策略，保证传输的这块的延迟。

第三点，用户量比较大，会统计分析每一阶段的大盘延迟分布，每天都会不同的报表展示。。



数据系统部分主要介绍四个：

第一，全链路跟踪质量展示分析。根据这个系统可以跟踪每一条链路中间经过的每一跳的服务器，它们之间的链路状况是怎么样的，包括错误码以及码率，这是一个全链路的分析。从每个用户到主播这条链路，根据唯一的一个 ID 就能查到这条链路上每一跳的质量，这样就可以针对线上不同的问题进行详细地分析。

第二，分段延迟分析展示。这个系统会根据不同的平台、不同的主播、不同的阶段，比如说编码、解码、缓存、渲染、传输等阶段延迟的情况。

第三，配置 AB 系统。不管是端上的推流端还是播放器端，都有大量的配置，比如说一些时延性的算法，一些业务的策略，可以做到根据不同的业务进行分析，对比效果。举个例子，要证明一下我们优化的延迟对电商成交有没有正向效果。我们会选比如珠宝类的一些直播，还有一些衣服类的直播等等，根据不同的行业去做 AB，到底提升效果是怎么样的。

第四，RTC 质量大盘。针对不同的地域、不同的域名会有一个整体的质量展示。包括推流质量、拉流质量，以及一些中间链路的质量等等。

● 淘系的音视频编辑方案：非线性编辑引擎

- 作者：午王
- 出品：淘系技术

在经历移动设备的更新换代，网络速度的持续提升和费用降低，手机用户已经经历了从文字阅读到图片浏览再到视频观看的内容消费的变革后，淘系音视频技术如何灵活根据需求做出技术创新与变革。

非编定义

非编是非线性编辑的简称，如果按照维基百科的解释，线性编辑和非线性编辑是相对的，是影视制作中的两种不同方案。

它们两者分别是录像带和数码两个时代不同的产物，最主要的区别在于是否是对原始数据进行破坏性编辑。

如果按照此解释，我们这里就应该把文章名改为《淘系的音视频多轨道编辑引擎设计》，才能更准确的说明我们想介绍的是什么。

所以我们更乐意的是抛开历史背景，从字面上来解释：无论音轨还是视轨，在时间线上无需依次线性排列的编辑方式，就是我们所说的非线性编辑。

如下图示例，使用非线性编辑的方式，我们可以灵活自由的根据我们的需要，编排我们的音频和图像资源。可以按时间顺序依次编排，也可以在时间上重叠在一起，也可以相互交错。

随着移动设备的更新换代，网络速度的持续提升和费用降低，手机用户已经经历了从文字阅读到图片浏览再到视频观看的内容消费的变革。现如今大家在利用手机进行休闲、购物、学习、生活记录等许多方面，都更乐于以视频作为媒介了。而非线性编辑作为视频生产中一个重要的技术环节，在 market 需求的推动下，也是发展的如火如荼。那么，淘系的音视频非线性编辑引擎Marvel是怎样设计的呢？



业务背景

近些年用户习惯逐渐从图文向短视频迁移，淘系大量的商家和用户，对于短视频内容的生产和消费诉求也日益提高。淘系生态下有着诸多内容生产的入口，面向不同的用户群体，不同的用户群体期望生产的内容也存在差异，如商家、达人和买家三者的生产目的可定就存在巨大差异。Marvel引擎就是期望为淘系生态下的所有入口，提供相对统一的解决办法，满足不同用户群体对于短视频的差异化生产需求。

对于短视频内容生产，编辑工具通常会分化为两种，一种相对来说，功能不是那么强大，只具备一些简单的能力，使用起来非常简单，这一类我们称之为UGC工具。UGC追求简单，降低生产门槛。另外一种，功能非常齐全，可以编辑生产出特别复杂的内容，有一定的使用门槛，我们称之为PGC工具。PGC追求齐全，满足专业需求。淘系的短视频内容生产，显然是这两种工具都需要。

淘宝逛逛中的发布器就是淘系短视频UGC工具的代表，而淘宝为淘宝商家打造的亲拍，就是淘系短视频PGC工具的代表。这两个业务底层的音视频编辑能力，都是有Marvel引擎提供的。

目标挑战

淘系短视频内容生产的工具，需要围绕淘宝电商生态之下，包括商家、设计师、ISV、达人、买家等等诸多角色的用户而展开。为满足各维度用户的生产工具诉求，保证各维度用户的相互促进，我们需要打造一个体系化的跨平台音视频非线性编辑引擎。满足UGC、PGC、OGC等多维度的短视频内容生产工具的业务诉求。并以资源为媒介，在工具间进行流转，满足各维度的用户生产需求。

需要打造这样一个体系化的音视频非线性编辑引擎，满足淘系不同目标的短视频内容生产工具的发展需求，我们也面许多挑战：

场景覆盖

淘宝用户众多，包括商家、设计师、ISV、达人、买家等等诸多角色，各角色对于生产工具的诉求存在巨大差异，需要引擎同时满足不同用户、不同场景的生产诉求。

平台覆盖

从用户生产数据来看，Android和iOS等移动端是重要的内容生产入口，但是也存在大量精品内容，来自PC制作。引擎不仅要支持移动端，也要支持PC端的生产工具建设。

功能体积

我们需要保证足够完备丰富的功能，支持PGC用户的设计创意。同时，我们又需要严格控制引擎大小，尤其是支持手淘UGC发布器时，避免带来应用体积增量过大的负担。

稳定安全

作为短视频内容生产体系的技术基石，面向众多的生产用户，在保证多业务多平台快速发展的同时，我们也要保证引擎稳定安全。

技术选型

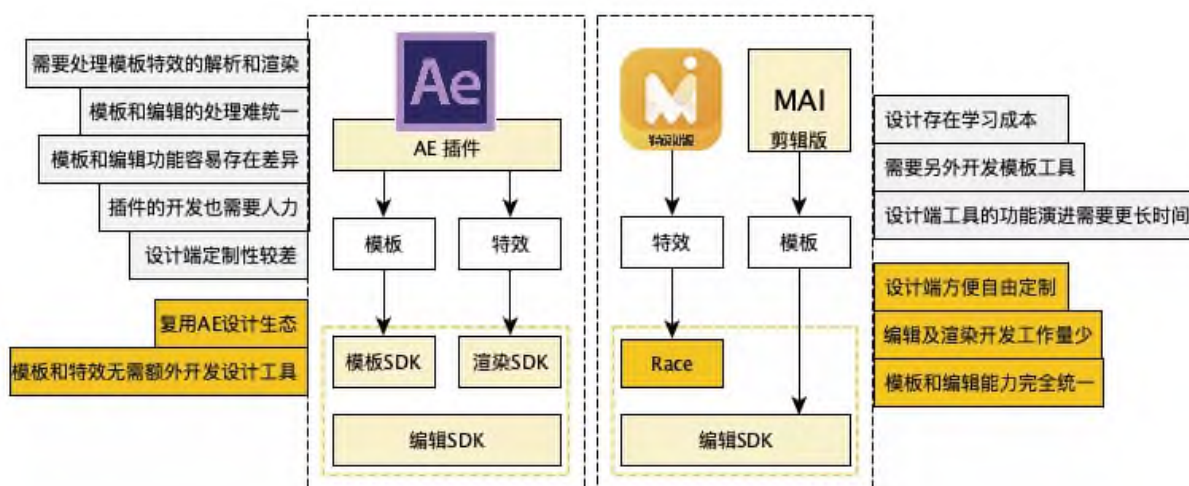
淘系的短视频内容生产工具的需求，不仅仅在移动端工具。整个淘系的短视频内容生产中，PC端、云端等配套工具也是不可或缺，有大量精品内容是来自PC制作。

所以我们的引擎构建目标，不仅要支持移动端生产工具的构架，也需要兼顾PC端、云端等生产场景。

业界许多短视频内容生产的解决方案，会选择复用AE生态，来解决设计工具与资源的问题。

一种是用AE来生产视频模板，然后通过SDK解析模板，生产视频。

还有一种会利用AE来生产特效，通过SDK来解析渲染特效，应用到编辑场景中。在当前阶段，我们对于复用AE生态的诉求并不强烈，当然，后续也不排除会考虑兼容AE生产的资源。



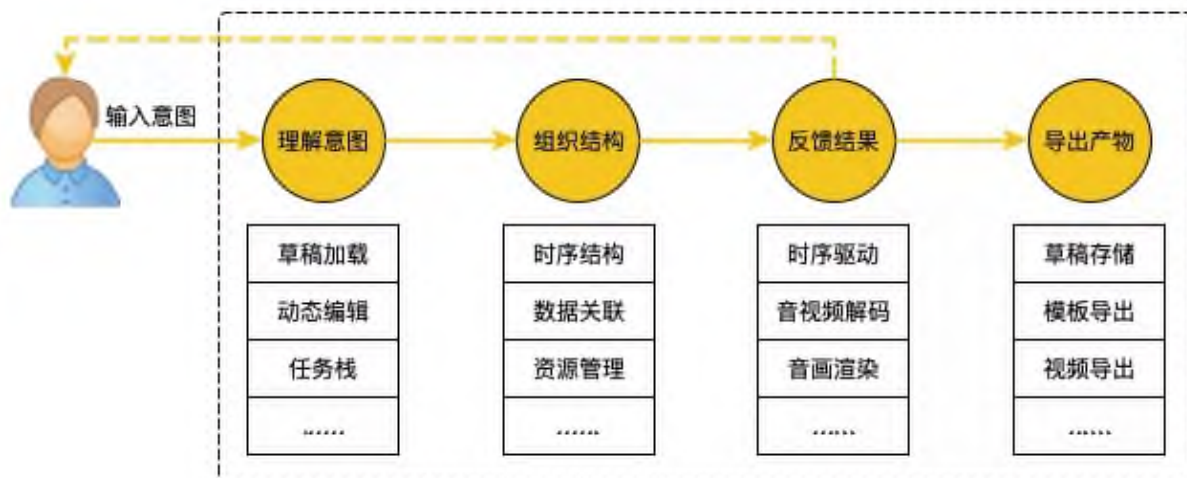
另外，对于引擎的实现，有些解决方案会选择基于开源音视频框架进行实现，例如GStreamer。GStreamer框架非常强大和通用，其内核小、模块化设计、功能强大、支持所有主流平台，但是也存在一些问题。

1. 它的模块化和强大的功能，是以复杂的设计为代价的，这导致它上手较难，二次开发也不方便。
2. 插件+流水线架构设计，在涉及时序功能、效率优化的方面，会存在链路过长、实现复杂的问题。
3. GStreamer内核小、可裁剪，但是随着功能的增多，也会因其设计，导致引擎的大小不容易优化。
4. 它是基于GObject+Glib进行面向对象的程序设计，对于我们这些Java/C++开发人员来很不友好。

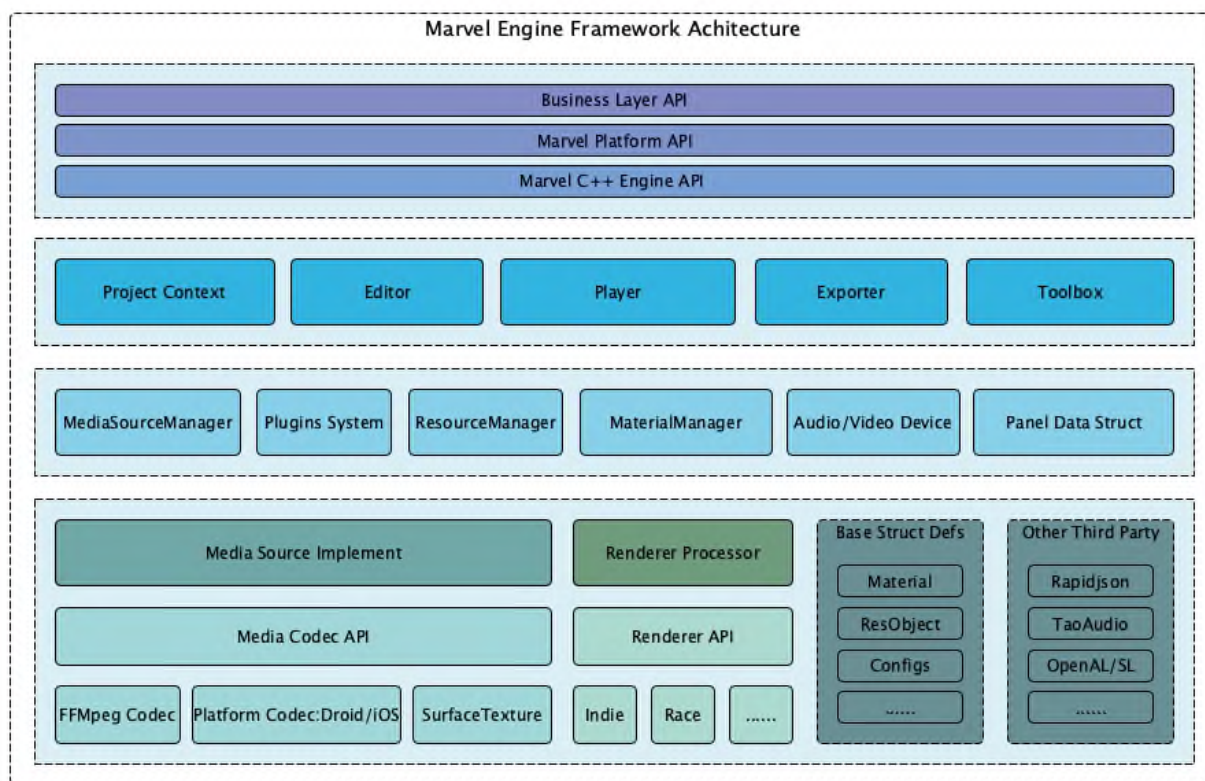
所以，最终我们还是选择用C++复用淘系已有的渲染、音频处理等框架能力自己撸一个跨平台音视频非线性编辑引擎。

架构选型

我们可以粗略的将音视频的非线性编辑引擎的工作流程拆解为四个部分，依次为理解用户输入的编辑意图，把用户输入的编辑意图组织为引擎更容易理解的信息结构，然后将信息结构的内容按照可视听的方式反馈给用户，最后用户在需要的时候将编辑的产物进行导出。无论接入的业务如何变化，引擎内的流程都可以归到这四个步骤之内。



据此理解，为保证多种平台、不同场景下的业务诉求，我们对音视频生产处理引擎需要满足业务逻辑进行功能模块划分，构造满足通用多轨道非线性音视频编辑的数据结构，然后以此为核心，向下层设计提出要求，并在迭代过程中进行调整和补充，最终逐渐形成了如下图所示，自下而上包括基础层、驱动层、逻辑层和封装层四层结构的整体架构。



在封装层中，我们对接口分级，C++ Engine API提供相对通用的接口，平台层对其进行包装，然后业务层基于平台层的封装进行业务接口封装。使核心接口稳定，避免功能、场景、业务增加带来的接口膨胀。

在逻辑层中，我们按照非线性编辑引擎的工作流程，以综合的业务逻辑划分功能模块，剥离与具体业务场景的直接联系，让不同业务可以根据场景需要进行组合，来满足不同需求。

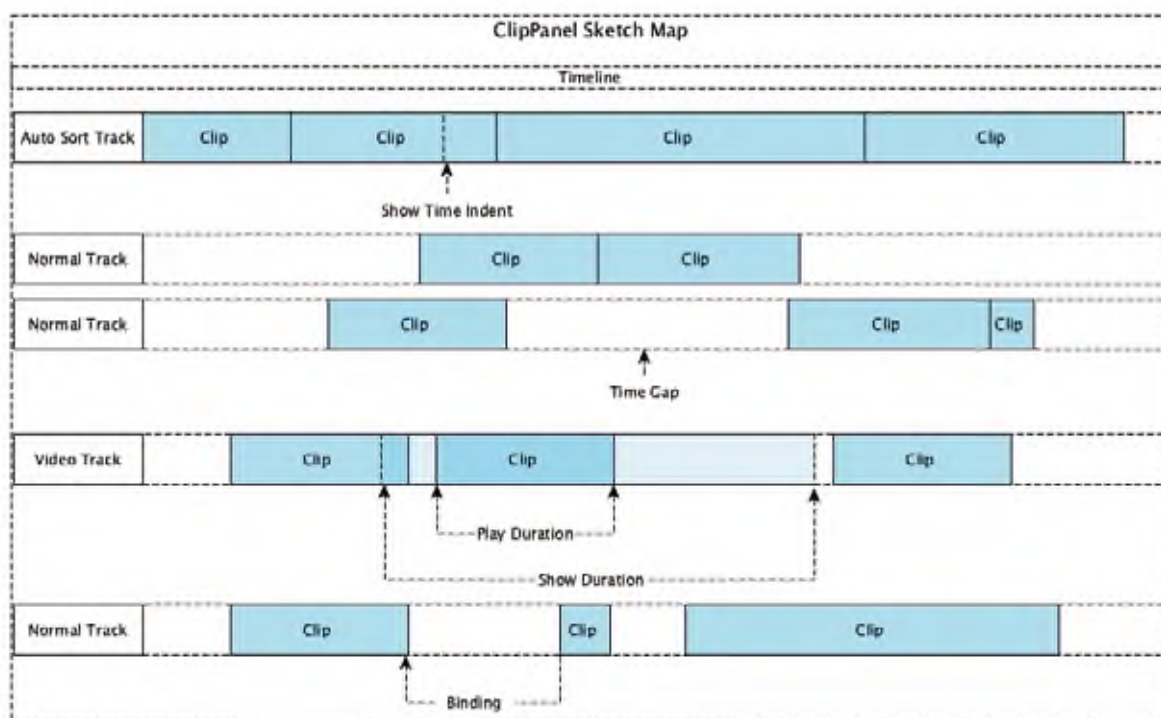
在驱动层中，我们构建了时序&数据关系、音画的驱动设备以及编辑上下文需要的多个子系统，用来管理和组织资源、数据源、材质、音画处理器等，依用户意图，结合基础层能力，按时序对外输出可视听的数据。

在基础层中，我们抽象了资源、音画源的接口，并针对不同类型做了实现，也提供了与业务逻辑无关的，相对底层的对象结构和基础能力。

数据结构

Marvel以“面板”、“轨道”、“片段”的概念来组合和处理多轨道非线性编辑的复杂时序逻辑，它们是Marvel数据结构概念的核心，以“材质”的概念，来区分、组合、满足对于非线性编辑中复杂的空间及功能逻辑，以“资源”的概念来保证和简化以Marvel为核心的短视频内容生产体系中资源的流转。

所有的资源的时序信息都通过面板（ClipPanel）来进行整合，一个编辑工程中有且仅有一个编辑面板，面板中可以存在一个或者多个轨道，每个轨道中可以存在一个或者多个片段。如下图所示。



抛开时序相关的功能和表现，不同类型的片段，对外的表现存在很大的差异，而且这些差异通常也要求外部可控可编辑，但是不同类型的片段，也会存在一些表现共同的地方。

比如一个视频片段显示到屏幕上，可以对这个视频图像进行缩放、位移、旋转、修改透明度、调整混合方式等，贴纸和文字等片段也需要能够进行这些操作，而这些操作对于声音片段、滤镜片段来说没有任何意义，也可以说无法应用。材质概念的设计就是为了处理这样的情况。

材质在渲染系统中常用的术语，表示的是表面各可视属性的结合。

Marvel借用了此术语，将其衍生为片段的所有附加属性（附加是相对片段定义已经存在的属性而言的）。

为了降低API上的理解成本，材质这个概念，会以片段的属性对外表现。

对于不同类型的片段，我们可以给他们增加不同的材质，在进行片段的处理时，我们根据片段的材质，找到对应材质的处理器进行处理，这样即对不同片段做了区分，也保证了相同处理流程的复用。

在Marvel中，当前材质可以按照其目标对象分为两类，一类是图像相关的，比如滤镜、特效、画面位移旋转缩放控制、画面效果调节等等。一类是音频相关的，比如变声效果、音量调整、降噪处理等等。Marvel中除时序

功能模块

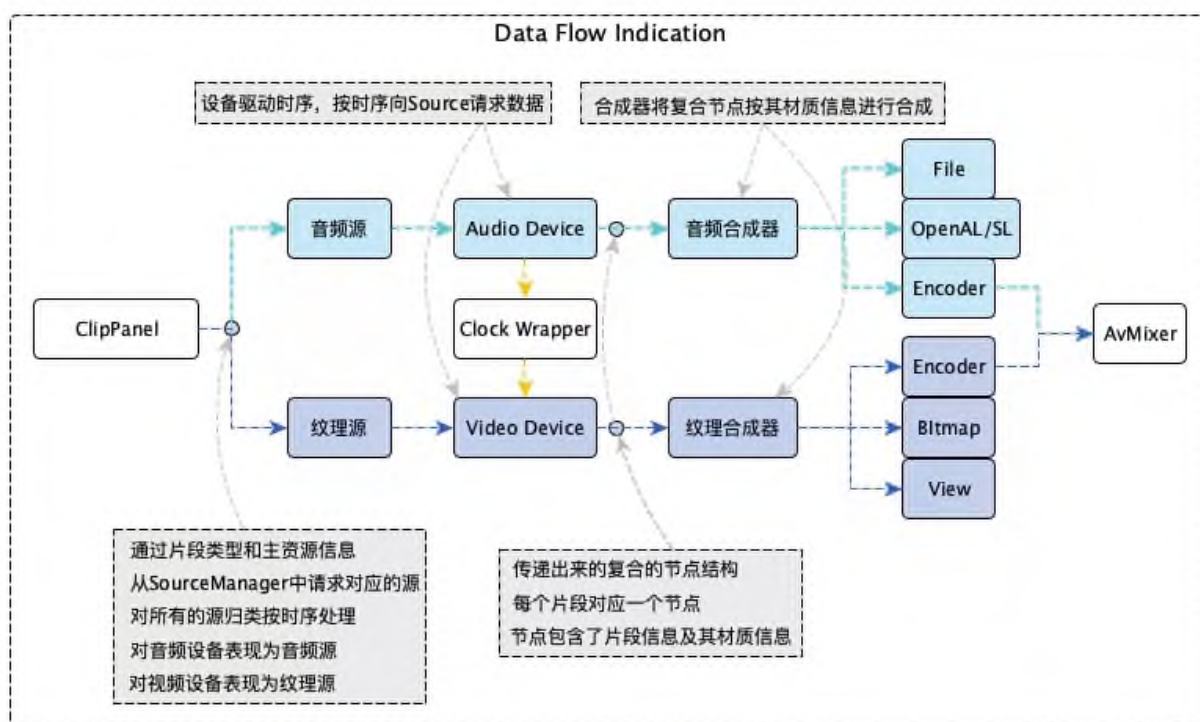
作为通用的跨平台音视频非线性编辑引擎，以市场上Top10的视频编辑工具为假想业务，从大的功能上进行划分，Marvel需要满足音视频多轨道非线性的编辑（包括贴纸文字特效等等）、编辑过程的实时预览、编辑产物的导出（视频、草稿、模版等）、为满足二次编辑能力的编辑产物加载。另外为了降低业务接入的门槛，Marvel可能还需要支持常见的音视频处理操作，比如音视频分离、视频截帧等等。Marvel的逻辑层模块划分就是由此而来。

逻辑层各个模块都是直接或者间接的围绕ClipPanel中的数据来按照各自的职责展开各自的逻辑。其中，我们可以理解为，Editor模块关注的是ClipPanel原始的数据，Player模块更关注的是按照ClipPanel原始数据构建出的源所提供的数据流（主要有音频流和图像流）。Exporter和Toolbox则按需决定其关注点，比如导出草稿的Exporter关注的是ClipPanel的原始数据，而导出视频的Exporter关注的ClipPanel构建的源提供的数据流。我们以数据结构和逻辑模块来对下一个层次提出设计要求，也就是驱动层。

这里的驱动层的并不是操作系统的驱动层那样驱动硬件运行，但是它也具有硬件驱动类似的含义。硬件驱动是按照硬件的时序图操作硬件，使硬件进行工作，而Marvel的驱动层是针对ClipPanel原始数据和运行时的数据流而言，是按照ClipPanel所描述的时序，以生产者-消费者的模型来展开设计，驱动数据流动，实现音视频的编

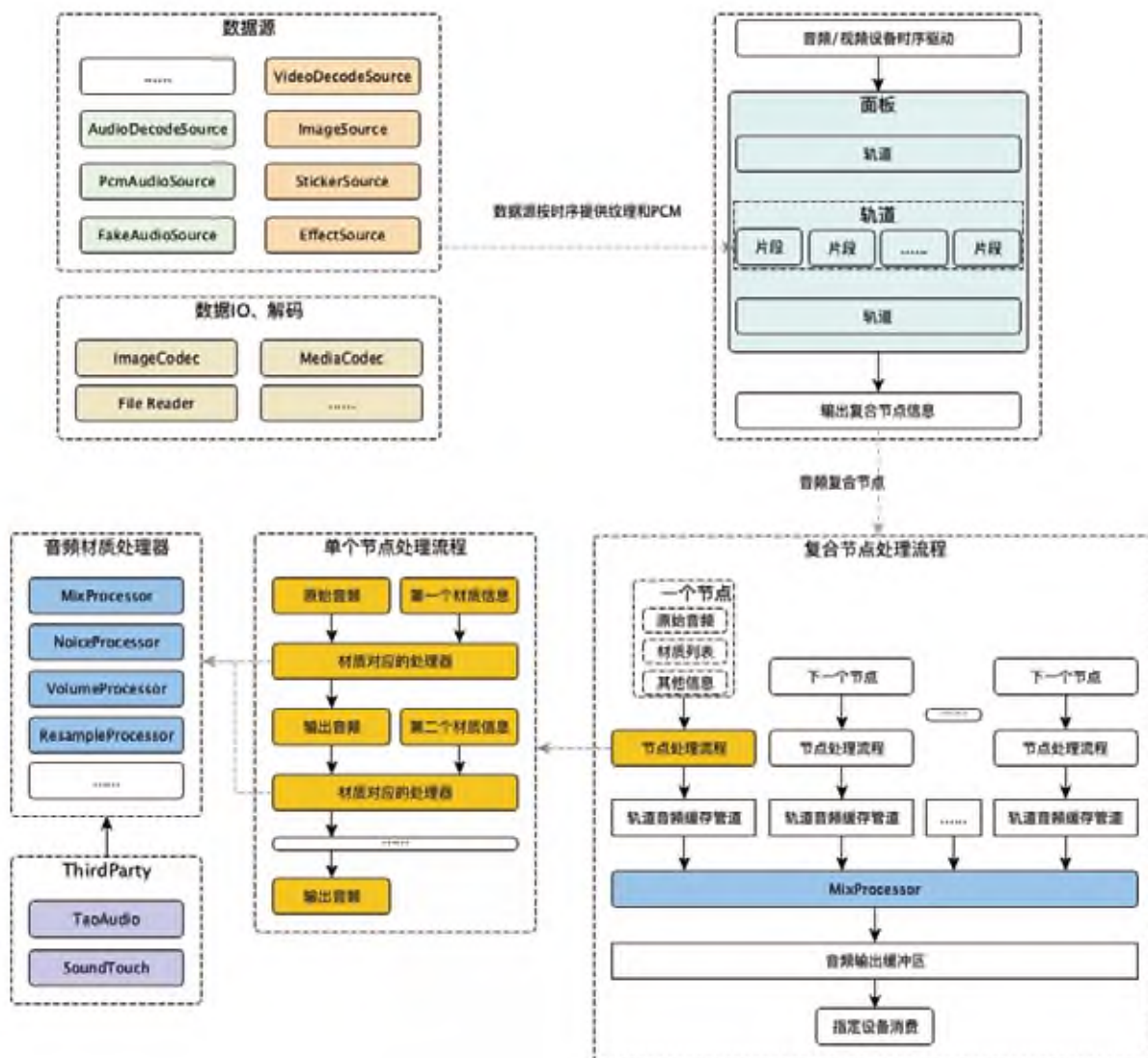
辑、预览及导出。

基础层的设计并不直接与逻辑层联系，是音视频生成处理中必不可却的部分。像音视频解码、图像解码、图像渲染、音频处理、文件读写等，这些基础的功能，无论上层如何设计，这些功能都必须提供，基础层的设计是剥离了逻辑层的需要进行思考和设计的，可能在后续的演进过程中，会因为逻辑层和业务的需要而有所改动，但是其原始的设计，我们认为是只与数据结构设计有关，和业务与逻辑层无关。



按照当前Marvel中的实现，内部数据流向大概如上图所示。Marvel中的ClipPane对Audio Device表现为一个提供音频原始数据的源，对Video Device表现为一个提供纹理数据的源。Audio Device和Video Device都是作为一个数据消费的中间装置而存在，按照时序获取数据，根据业务场景的不同，将数据传递给不同的消费者。

在处理流程中，音频源和纹理源提供出来的都是对于的复合节点，合成器用于解释并处理复合节点，将音频复合节点处理成PCM数据，将纹理复合节点处理成纹理数据。以音频为例，其处理流程大致如下，纹理的处理流程，也基本与此类似。



接口设计

与许多引擎或者SDK有所不同，由于视频编辑的复杂性，不同的业务场景对于视频编辑的诉求也有很大的差异，Marvel为了保证在维持接口简单清晰的基础上，满足不同业务的诉求，将接口进行了分层，这部分在架构图中也特意进行了区分和标注。Marvel Engine C++ API按照架构中逻辑层的划分进行类C接口定义，不区分业务场景进行接口设计。Platform API基本就是按照C++ API对各平台和语言进行的封装。

Marvel Engine C++ API的存在，主要是期望避免随着业务数量和业务诉求的增多，Marvel的API越来越臃肿最后难以维护，所以尽可能的屏蔽业务场景带来的影响，这样同样也使得这套接口对业务并不友好。所以在C++ API和 Platform API之上，还存在Business Layer API。Business Layer API是按照业务诉求对 Platform API或者C++ API进行封装，我们担忧的接口膨胀，主要来源于Editor，Marvel会按照常见的视频编

辑应用，内置一套通用的业务API，同时也允许业务接入方自行扩展。

C++接口和业务功能没有关联，把编辑动作简化为数据的增删改查，通过key-value来增删改查内部属性。业务层接口再对这个接口进行封装，可能是单一接口封装，肯能是多个接口组合，视业务需求而定，形成业务接口。如下，分别为C++接口和业务接口示例：

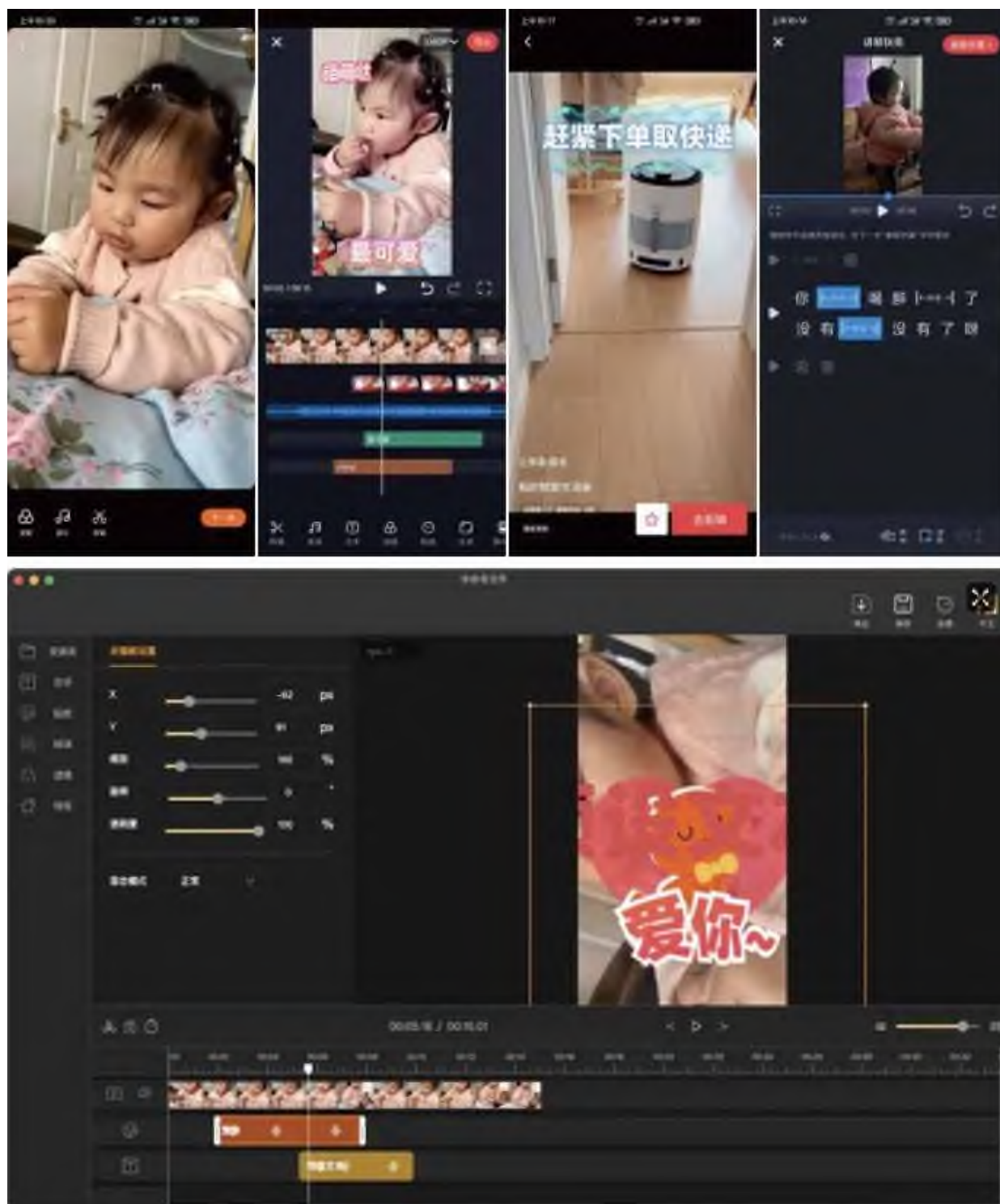
```
1 // 设置和获取片段的属性
2 int setClipStrProperty(EditorHandle h, const String& id, const String& type, const String& key, const String& value);
3 int setClipI64Property(EditorHandle h, const String& id, const String& type, const String& kye, int64_t value);
4 int setClipDbIProperty(EditorHandle h, const String& id, const String& type, const String& kye, double value);
5 int getClipI64Property(EditorHandle h, const String& clipId, const String& type, const String& key, int64_t& value);
6 int getClipDbIProperty(EditorHandle h, const String& clipId, const String& type, const String& key, double& value);
7 int getClipStrProperty(EditorHandle h, const String& clipId, const String& type, const String& key, String& value);
8 // 移除片段的属性
9 int removeClipProperty(EditorHandle h, const String& clipId, const String& type, const String& key);
10 // 设置资源的属性
11 int setResStrProperty(EditorHandle h, const String& resId, const String& key, const String& value);
12 int setResI64Property(EditorHandle h, const String& resId, const String& key, int64_t value);
13 int setResDbIProperty(EditorHandle h, const String& resId, const String& key, double value);
14 int getResStrProperty(EditorHandle h, const String& resId, const String& key, String& value);
15 int getResI64Property(EditorHandle h, const String& resId, const String& key, int64_t& value);
16 int getResDbIProperty(EditorHandle h, const String& resId, const String& key, double& value);
```

```
1 /**
2  * 设置一个片段上图像的裁剪信息，裁剪出来的区域为保留区域
3  *
4  * @param id 目标片段ID
5  * @param x x方向起始坐标
6  * @param y y方向起始坐标
7  * @param w 裁剪宽度
8  * @param h 裁剪高度
9  * @param rotate 裁剪旋转信息
10 * @param normalize 裁剪参数是否为归一化参数
11 * @param rotateWithCropCenter 旋转是否使用裁剪区域的中心作为旋转中，false时使用图片中心作为旋转中心
12 * @return 执行结果
13 */
14 public int setClipCrop(String id, float x, float y, float w, float h, float rotate, boolean normalize, boolean rotate
    WithCropCenter) {
15     int actRet = editor.cCheckToAddMtl(id, C.kMaterialTypeCrop);
16     if (actRet != 0) {
17         return actRet;
18     }
```

```
19 editor.setProperty(id, C.kMaterialTypeCrop, C.kMaterialKeyRotate, rotate);
20 editor.setProperty(id, C.kMaterialTypeCrop, C.kMaterialKeyXOffset, x);
21 editor.setProperty(id, C.kMaterialTypeCrop, C.kMaterialKeyYOffset, y);
22 editor.setProperty(id, C.kMaterialTypeCrop, C.kMaterialKeyWidth, w);
23 editor.setProperty(id, C.kMaterialTypeCrop, C.kMaterialKeyHeight, h);
24 editor.setProperty(id, C.kMaterialTypeCrop, C.kMaterialKeyNormalizeFlag, normalize);
25 editor.setProperty(id, C.kMaterialTypeCrop, C.kMaterialKeyRotateFlag, rotateWithCropCenter);
26 return 0;
27 }
28
29 /**
30  * 设置一个图像类片段的锚点
31  *
32  * @param id 目标片段ID
33  * @param x x方向锚点, 典型值为-0.5 - 0.5
34  * @param y y方向锚点, 典型值为 -0.5 - 0.5
35  * @return 执行结果
36  */
37 public int setAnchor(String id, float x, float y) {
38     editor.setProperty(id, C.kPropertyCanvas, C.kMaterialKeyXAnchor, x);
39     return editor.setProperty(id, C.kPropertyCanvas, C.kMaterialKeyYAnchor, y);
40 }
41
42 /**
43  * 设置一个图像类片段的旋转角度
44  *
45  * @param id 目标片段ID
46  * @param rotate 旋转角度, 弧度
47  * @return 执行结果
48  */
49 public int setRotate(String id, float rotate) {
50     return editor.setProperty(id, C.kPropertyCanvas, C.kMaterialKeyRotate, rotate);
51 }
```

业务案例

当前Marvel已经接入了淘系多个业务产品中, 包括手淘逛逛发布器、亲拍、MediaAI Studio、闲鱼、点淘等, 主要的业务形态, 包括拍摄后编辑、深度编辑、影集模板、口播编辑, 另外还有一些水印处理、云剪辑等等之类的业务形态。随着Marvel的发展, 未来还会支持其他业务, 满足更多的业务诉求。



总结展望

本文以淘系的音视频非线性编辑引擎Marvel为主题，介绍了它的应用场景和引擎设计。音视频非线性编辑涉及到许多技术，包括编解码、音频处理、图像算法、图像渲染等等诸多方面，都是非常有意思的东西，每一块都可以单独成为一个技术方向。非线性编辑引擎不仅仅是基于这些技术的组装，也需要对这些技术进行升华，挖掘它们在非线性编辑中的业务价值。

在当前的规划中，后续我们除了功能与效率方面的工作外，还将持续针对Marvel引擎围绕着高效、稳定、灵活来进行技术上的演进，如增加预处理流程、进一步推进图像渲染&音频处理&预处理的插件化、增加片段间在时序上的相对布局等等。针对诸如远程资源下载、资源依赖管理、草稿动态转换、三方资源支持等非编辑核心能力的扩充，我们也将进行梳理，后续以插件的方式进行注入支持。

未来我们将继续以Marvel为编辑核心，推进淘系完善的短视频内容生产体系构建，保障淘系包括商家、买家、达人等诸多用户角色的短视频内容生产诉求。并在此基础上，沉淀技术，开放生态，持续为其他业务赋能。

● 视频内容理解在手淘逛逛中的应用与落地

- 作者：李晓波（篱悠）
- 出品：淘系技术

随着多媒体技术的发展，直播、短视频、AR等多媒体内容表现形式层出不穷，异彩纷呈。视频内容的理解在视频生产感知，理解分发中有哪些应用，为淘宝电商带来哪些影响？这几年，手淘商品展示模式一直在变化。1998年做文本，2005年进入图文时代，2017年进入淘宝直播时代。发展到今天，手淘新的业务增长点在哪里？下面我会从视频内容理解的角度着手分享在手淘逛逛中的应用与落地。



淘宝探索人货场新形势的发展历程



电商的表现形式在不断发生变化。友商像抖音、快手等内容厂商的DUA规模、用户时长都有快速增长；技术方面目前多媒体技术向内容化、社区化方向蓬勃发展。基于以上几个方面，手淘下一步要做以下几个方向：升级传统RGB直播的形式，从2.5D到3D再到AR形式做升级；手淘会探索内容业务的的增长点，从原来商品介绍转至对内容的介绍。当然万变不离其中就是通过多媒体形式做商品的表征。

逛逛的愿景



逛逛想要传达的理念，第一传达生活方式，例如构造真实消费的场景进行购物；第二由于是真实的人，每个人都有自己的人设，有自身人格化，希望逛逛产品传递人格化；第三逛逛产品不希望是高不可攀的，而希望是每个人都消费得起、性价比高的产品。上述就是逛逛的愿景。

逛逛内容业务面临的问题



首先要有愿景做产品，最后落到技术上来看，面临技术挑战有以下几个部分：

生产者角度上的问题是：一是逛逛以短视频拍摄为主的产品，原来拍图片生产成本低，拍视频生产成本较高，需要选标题、封面图，做转场使视频生动有趣，做特效玩法（例如：张嘴、眨眼、口吐莲花）；二是视频生产可以是视频内容的二次加工得来，那商品、场景、特效素材从哪里获取？

平台的角度上的问题是：所有生产的视频汇集在平台侧，每日生产的视频量是一个海量的数据。一是海量内容参差不齐，如何对内容的快速审核、挑出优质视频内容；二是内容具有一定时效性。商品可以春夏秋冬卖一年，内容也会有热点信息，以及海量信息该如何进行快速处理？三是手淘商品有完整的结构化信息供运营，视频内容的raw data信息如何进行精细化运营？

消费者的角度上的问题是：每个人时间有限，除了工作，休闲时间刷刷抖音，看看逛逛的时间更加有限。另外对消费者而言，信息海量，例如，手淘每天产生几百万短视频，一个个看无法看完，如何形成消费者内容画像、根据消费者内容画像匹配合适内容就是我们要解决的问题。

逛逛内容视觉算法的整体技术架构



内容视觉算法分成两块：互动视觉和互动玩法来帮助内容分发；在内容理解方面拆成四部分：智能生产，智能审核，智能解锁，内容语义。本次分享重点讲的是智能生产、智能审核、智能解锁、智能语义。

智能生产

第一部分是内容生产面临的问题。

智能创作



智能生产上述提到了一些问题，在问题之下，分享一下解决方案。解决方案分为两部分。

第一部分是提高视频的发布效率，如何让发布者更高效发布内容，生产短视频。自动生产一些滤镜、标题，自动生成一些可以打动人心的话题、自动调节色彩空间，提升视频发布的成功率。

第二部分是素材再加工。将原始拍摄的视频内容拆散开来，将商品、人物从原始视频中抽出。与智能模板，智能滤镜等结合，二次加工成短视频。

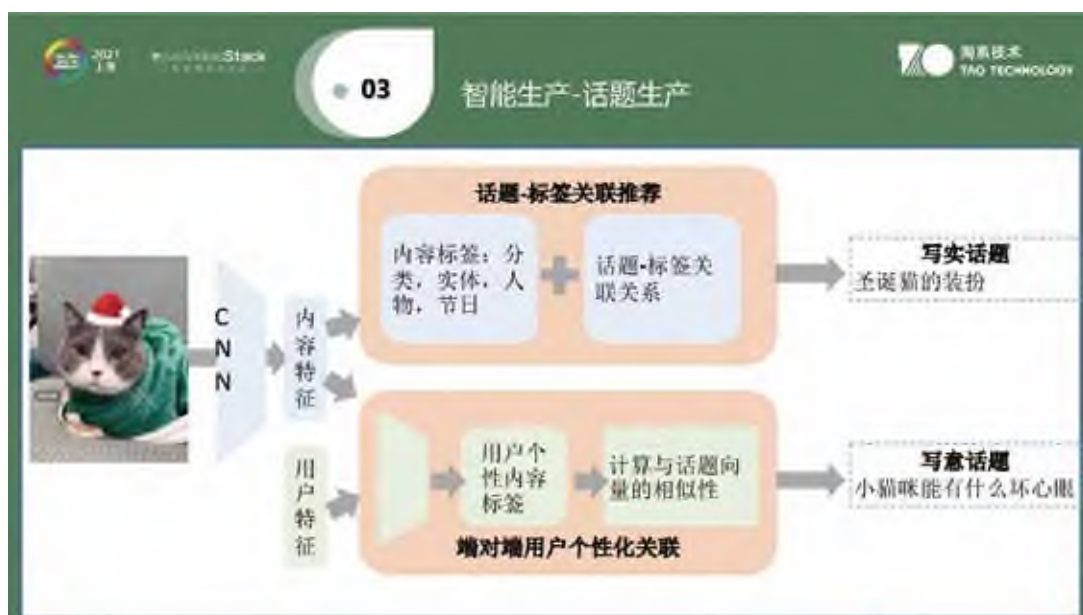
以上是智能生产这两部分要做的整体介绍。其中涉及内容很多，后续会挑选关键点讲一下具体如何来做到。

视觉元素解构



首先讲到关于智能生产中视觉元素解构。对内容重新编辑需要将原始内容从原有视频中拆出来。为了做到这些，我们构建了一套完整分割体系，从最简单的纯色分割、类目分割到头发分割、固定人像分割，到指甲、身体分割都可以将其分割出来。也就是说我们将商品或人从RGB信息中分离出来，后续根据商品的颜色和分发主题来搭配相应颜色空间（智能滤镜）等将其二次加工，快速生产想要的视频。那么完整的分割体系是保证视觉元素分拆的重要一步。

话题生产



一是一个视频在分发过程中如何打动别人。对于话题（这个视频的内容），可以用深度学习方案将用户特征提取出来，打上个性化话题。

智能封面图



二是在拍摄视频过程中，如果没有封面图，那么视频首帧就是封面图。但会有几个问题：视频首帧无法完整表达视频的整体含义；首帧容易黑屏、花屏。此时如何在拍摄的视频中精选最能够代表视频含义的帧就是我们所要做的事情。

互动玩活



三是如果有了封面图，并赋予它语义含义后，要将视频变好玩，需要互动玩法。例如，在友商的一些产品中，有眨一下眼睛就会出眼泪；一笑就会出桃花。这些互动玩法就可以快速将短视频变得生动有趣。而在这一板块，我们提供了一整套2D、3D的特效玩法。

智能审核

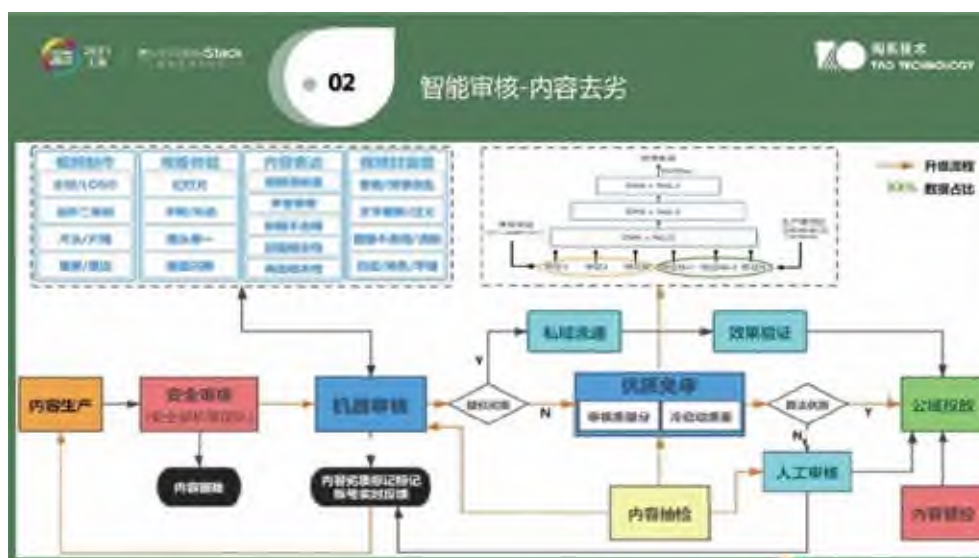
讲完了关于智能生产部分，第二部分分享智能审核。

OCR



一是OCR。整个视频审核过程中是跨模态的，有语音信息、文本信息、视觉信息。在整个过程中，我们会把所有视频、图片含有的文字信息摘出，有了它可以做很多事情。上图是整个OCR链路体系。基本流程和普通图片OCR相似。

内容去劣



二是有了跨模态信息，智能审核分为两大步骤，第一步是内容去劣，第二部是内容择优。在整个视频中出现抖动、二维码LOGO不符合场景分发等情况时，则会对内容进行降权。这一块有60多种模型来解决内容去劣的问题。

内容择优



第二步是内容择优。所有短视频分发前都需要进行人工审核，在有限人力下，如何将好视频快速分发？内容择优就是通过算法优先找出相应的优质视频，找出后并不是直接分发，而是优先将其分配给人工进行审核，保证优质视频时效性，在最短时间在公域中流转。上述就是内容择优的内容。封面图不都是自己生成，会有用户上传封面图，但其与内容完全没有关系。那么可以在择优模型中，从商品人物一致性，商品调性出发，使用美学模型方法将优质视频挑选出来。

内容检索

第三部分是关于内容检索方面。

原子能力



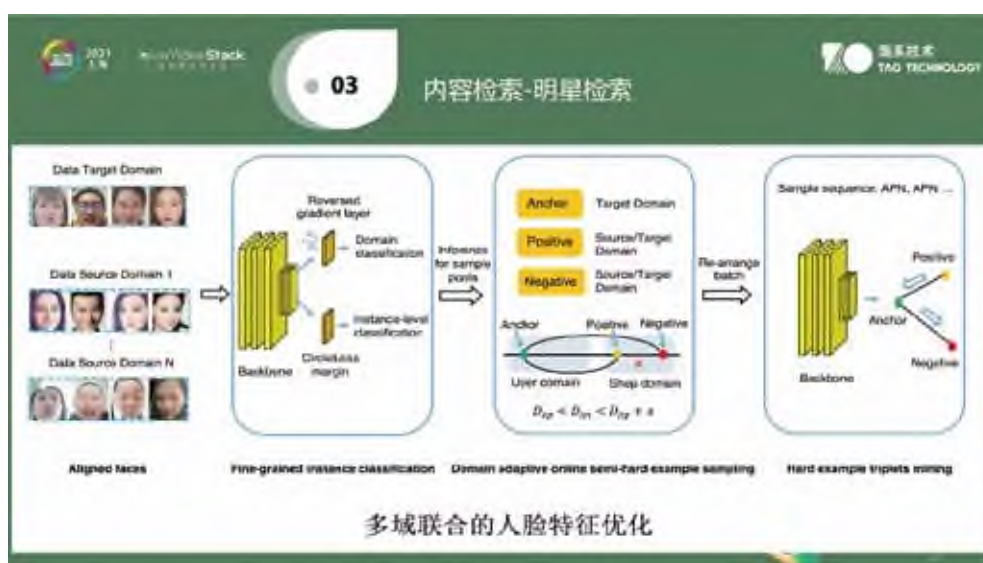
第一需要构建完整内容检索的原子能力。最早在深度学习之前，运用SIFT特征（SLAM中运用ORB特征使之更快）把局部特征构建成为全球特征做检索。深度学习后，全局特征很好提，那么局部特征怎么提呢？我们构建了局部特征算子，整个检索有局部和全局特征。构建后做整个上层检索，以文本搜图片或视频，以图像搜视频，以视频搜视频等各种应用。检索技术是通用的，但运用场景不同，则上层会形成几种产品：直播看点，解决商品搜索过程；内容去重，在整个视频中会有很多重复，调整分辨率帧率或是轻微裁剪，颜色空间变化，本质上内容一致，如何做内容去重；明星识别，识别视频中出现的角色，人物关系；视频推荐，如何与视频检索结合。上述是内容检索中的原子能力。后续会围绕着上层能力做详细介绍。

内容看点



第二是直播看点，内容看点。有很多用过淘宝直播的用户知道回放时会有看点。它是在小的闭集上做商品检索。过程如上图：第一步是数据预处理，一段视频中不是每帧信息都是有用的，将有用帧挑选出来；第二步是通用物体检测，检测出每段具体卖的商品；第三步是时空轨迹/tracklet，在直播商品的过程中，主播卖的商品与时间前后和空间有关，例如卖手机，左手展示，商品展示与物理空间有关；第四步是Query提取，从音频信息到OCR信息；第五步是文本信息与视觉信息结合，进行多模态提商品特征。第六步是在闭环内对商品特征做检索处理。

明星检索



第三是明星检索。在拍短视频或整个直播过程中，出现哪些人，这些人是谁。也许不知道这个人的名字，但我们会给他Face ID，不管在哪个视频中出现，都会被完整提取出来。其核心在于一方面是多域联合，另一方面现在支持的是千万级别到亿级别的，在lost function这一块去做arcface loss。

内容去重



第四是内容去重。分享一下内容去重需要解决的问题，淘宝有分佣机制。当一个视频可以返利时，会有人盗取视频并加入自己的商品链接。视频中会进行裁帧处理，分辨率变化等形变。我们需要将作弊的视频检索出来，以上是内容去重需要做的事情。叫内容不叫视频的原因是现在我们对文本、图片、视频、帖子等形式的去重都支持。

内容语义

讲完了内容检索后，第四部分来分享一下我们在内容语义方面做的事。

视频分类



第一个最常用的是视频分类。无论长视频或短视频都要进行分类，这事情说简单也简单，说难也难，是因为有很多视频在分类过程中，不一定是视觉可分。因此整个分类过程中是跨模态的，会把ASR信息或整个OCR信息联合去做分类。往往分类不是分一级类目（搞笑、美食），一级类目下还有二级类目（美食下有潮汕美食、浙江美食、上海美食），在整个分类过程中，一级类目与二级类目联合进行分类，借助两个分类之间的相关性，尽量减少不一致性。如果一级分类与二级分类相差过大，认为这是有问题的，则会重新调整。通过两级分类联合相互监督使之准确率更高。

视频标签



第二个是视频标签。在整个视频中分类比较有限，一般来说一级类目几十种，二级类目一两百种，除了分类信息，还可以打大量标签。这些标签如何产生，判别什么样的标签是有意义的需要与各自业务结合起来（比如说视频中有一瓶水，里面有两个人。打这样的标签不一定是有意义的）。这需要和各自业务域结合产出有意义的标签。如果是安防产品，会对人或刀具或打架行为关心；如果是电商，则会对商品和出现的人比较关心。这里会有业务上的设计，而有了业务上的设计，跨模态理解最终会把想要的标签生产出来。

内容向量化



第三个是内容向量化。把语义理解后，需要与最终的搜索推荐系统结合，在结合时会有很多结合点。第一步将类目信息或Face ID整个传输给搜推，让其做后续推荐。推荐时那一页不可能都是你喜欢的那件商品，例如我喜欢手机，但我打开搜推系统这一页，不可能一屏全是手机或是我喜欢某个信息。这样做的原因是搜推的需要有多样性和新颖性。如果用户搜了一个洗衣机，那么如何将推荐打散？比如视觉方面，当用户输入文本信息时，会将视频中的洗衣机标签提取，使洗衣机类目打散。

兴趣图谱



第四部分是兴趣图谱。每个视频独立成体系，这些海量视频之间标签的关联性是维度，可以汇集成视频与视频标签之间的关联关系，是标签图谱。另外一方面，一个人看了许多视频，中间的关联性和共通性可以通过标签、属性形成基于个人内容兴趣图谱。针对这一方面，由单个视频上升到群体行为构成整个视频之间的标签图谱或兴趣图谱，上述就是做兴趣图谱的事情。

认识推理



第五部分是认识推理。兴趣图谱构建是个体与群体行为，群体行为分为两个方面，看过的视频与看视频人之间的关系。

训练体系

逛逛算法模型不少。第五部分来分享一下训练体系。

训练体系



讲训练体系的原因是在做标签、内容时，如果类别上到千万级别，会遇到长尾问题，解决现有数据问题时会遇到小样本问题。举个例子，手淘中最不缺的是商品样本；逛逛是做内容的，不一定是商品。我们会发现内容生产者为了点击率生产软色情内容，在手淘中属于小样本。如果要做软色情的识别分类器，会发现手淘中没有很多样本（因为我们不是社区）。所以我们需要有一套体系根据样本分布，如果有海量样本，就需要用监督学习来解决。把所有样本花钱进行标注，标的样本越多，越精准。但另一方面如果标注的都是简单的样本，不一定可以随着样本量上升，精度会线性增长。找出难样本有主动学习方式，同时也能节省标注的经费。长尾的东西多半是小样本，会有半监督、自监督方法、无监督方法做体系，将整体分布训练做起来，会形成自己的训练体系去解决整个在逛逛中遇到的各种问题。

淘系2021技术资讯重点

• 2021双11淘系技术九大技术亮点发布

○ 出品：淘系技术

每一年的双11，淘系技术都在努力让技术变得更好一点；

每一年的双11，淘系技术都希望利用前沿新技术，让双11变得更好逛更好玩；

每一年的双11，淘系技术都致力于用技术的力量，带来一场绿色低碳又不失温暖的双11；

为了每一年双11的卓越体验，淘系技术都离你更近...

至此，请收下淘系技术的这份成绩单《2021年双11淘系技术九大技术亮点》

2021.11.11 TAO TECHNOLOGY 淘系技术九大技术亮点发布

★ 淘宝3D沉浸式直播间

基于全球首个影视级扩展现实的云拍摄技术，
让商家花100元10度电就能搭一场100万成本的3D直播舞台

★ 周冬雨Pick的淘宝拍照

超大规模商品多模态搜索，淘宝拍照让你想买什么就拍什么

★ 国内首款AR试表

毫秒级别地画出手表在用户手上的逼真效果，体验效果行业领先

★ 3D建模小能手Object Drawer

业界首个基于神经渲染的3D建模产品Object Drawer，
推理速度提升10000倍

★ 智能调度低延时系统GRTN

1秒内的直播延迟，让李佳琪自信的喊出：美眉们，321！

★ 智能化淘宝内容审核

基于深度学习网络和卡尔曼滤波算法，轻轻松松10秒完成一个
内容机审，人机高效协作

★ 两极化剪枝 算力黑科技

剪除深度学习神经网络中冗余的神经元，每天节省14.4万度电

★ 新品孵化平台

数字化赋能中小企业，推动新发商品成功率提升100%

★ 智能助手淘小宝

淘宝长辈模式上线，说话搜代替打字搜，咱爸妈也可以畅快的
买买买啦

PAYMENT TAO TECHNOLOGY
团队的努力



• 淘系技术发布业界首个基于神经渲染的3D建模产品Object Drawer，推理速度均提升10000倍

- 作者：淘系技术3DAI团队
- 出品：淘系技术

在China VR 2021大会上，阿里巴巴淘系技术发布了业界首个基于神经渲染的自研商品三维建模产品Object Drawer。目前，Object Drawer已经提供给部分天猫、淘宝商家使用，将在12月对外开放试用接口。届时欢迎大家体验。



团队成员在第二十一届中国虚拟现实大会上展示

在1920*1080分辨率下，Object Drawer的推理速度可以达到200FPS，在手机上可达到30FPS，实现了实时高清可交互的三维模型。与此同时，模型的训练时间仅仅需要4小时，模型大小仅仅需要20M，同时，对于任意视角查看，商品三维模型都达到实拍照片的展示效果，实现了三维建模一直追求的自动高精度还原。Object Drawer不需要依赖特定设备，只需要手机环绕目标商品进行视频拍摄，就可以完成自动重建，所生成的三维建模效果达到高精模型还原度。

Object Drawer技术指标如下图：

首个基于神经渲染的商品建模产品 Object Drawer

- **渲染分辨率：1920*1080**
- ✓ **建模时间：约4小时**
- ✓ **渲染速度：GPU(V100)-200FPS 手机-30FPS**
- ✓ **重建质量：纹理细节1比1还原，PSNR约30**
- ✓ **模型大小：约20M**
- ✓ **类目：支持鞋、包、手办、积木、瓷壶、帽子、沙发、茶几、柜子**
- ✓ **拍摄要求：一般场景，手机拍摄，无需特殊设备支持**
- ✓ **支持应用：商品展示，三维场景搭建、互动，渲染2D场景图/视频**

背景介绍

随着VR/AR应用的发展，商品三维建模需求持续增加，商品建模需求可能达到现在的100倍以上。目前的商品三维建模依赖于手工建模，建模需要几小时到几天时间，费用为数百到数千元，即使是花费巨大的时间和金钱，商品还原度依然存在许多问题。传统的自动三维建模技术，需要对目标商品的形状、纹理、材质进行还原。为了提升建模成功率，建模产品对于建模环境、拍摄设备提出了许多要求。即使是这样的前提下，自动建模的三维模型还原度依然存在严重问题，弱纹理物体比如鞋、瓷壶等无法很好的重建，需要人工修复。同时，高精度材质还原，也需要人工进行大量的工作。传统建模失败案例与Object Drawer建模对比如下图：





传统建模失败案例与Object Drawer建模对比1

近几年，神经渲染技术正在飞速发展，其旨在利用神经网络来model场景和物体并可控制以及交互的渲染出photo-realistic的图像和视频。其中，NeRF及其衍生技术利用神经隐式表达来存储场景的物理几盒以及材质纹理信息，端到端的对于渲染效果进行优化，在novel view合成上取得了重大突破，也给三维重建带来了新的使点。但NeRF相关技术在实际应用中存在许多问题，部分主要问题包括：第一，推理速度、训练速度慢，1帧高清图推理时间超过50s，一个物体的建模时间长达2天以上。第二，细致纹理无法还原。第三，视角鲁棒性不好，部分视角渲染效果不理想。第三，隐式表达无法直接导入图形学工具，不支持显式使用，例如CAD场景搭配设计；同时由于只能还原拍摄场景的光照也使得NeRF模型无法支持环境光照变化的场景应用。以下为Object Drawer针对上述问题的一些解决情况。

推理速度和训练速度

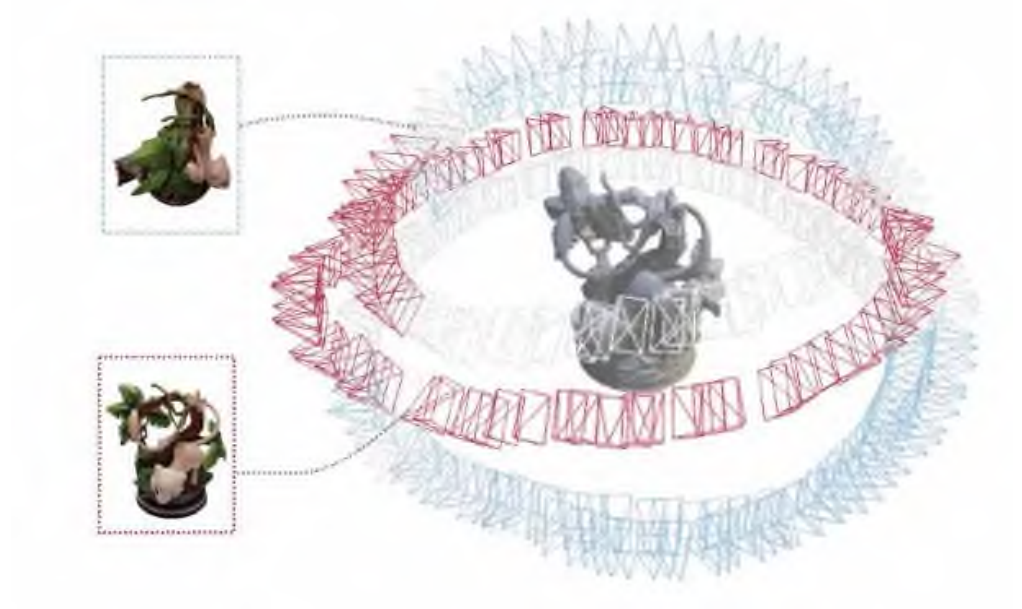
神经渲染的推理速度是近期的研究热点，原始NeRF利用volume rendering得到像素颜色值，渲染单像素通常需要forward一个MLP网络上百次。部分突破性工作，例如FastNeRF和PlenOctree，通过优化网络结构，探索几何先验，预存部分信息，在保证渲染质量的同时显著将1080p渲染速度提升到了100FPS左右，同时模型存储要求压缩到了400M左右。Object Drawer深入分析了神经推理过程中的冗余计算，提出了新的模型表示，相比NeRF实现了10000倍的推理速度提升，模型大小也控制在20M以内。

	推理速度 (V100)	推理速度 (iPhone 12)	存储
NeRF	0.02FPS	N/A	~15M
DONeRF	5FPS	N/A	~15M
FastNeRF	70FPS	N/A	> 1GB
PlenOctree	52~102FPS	N/A	1.9 ~ 0.4G
Ours	200FPS	30FPS	~20M

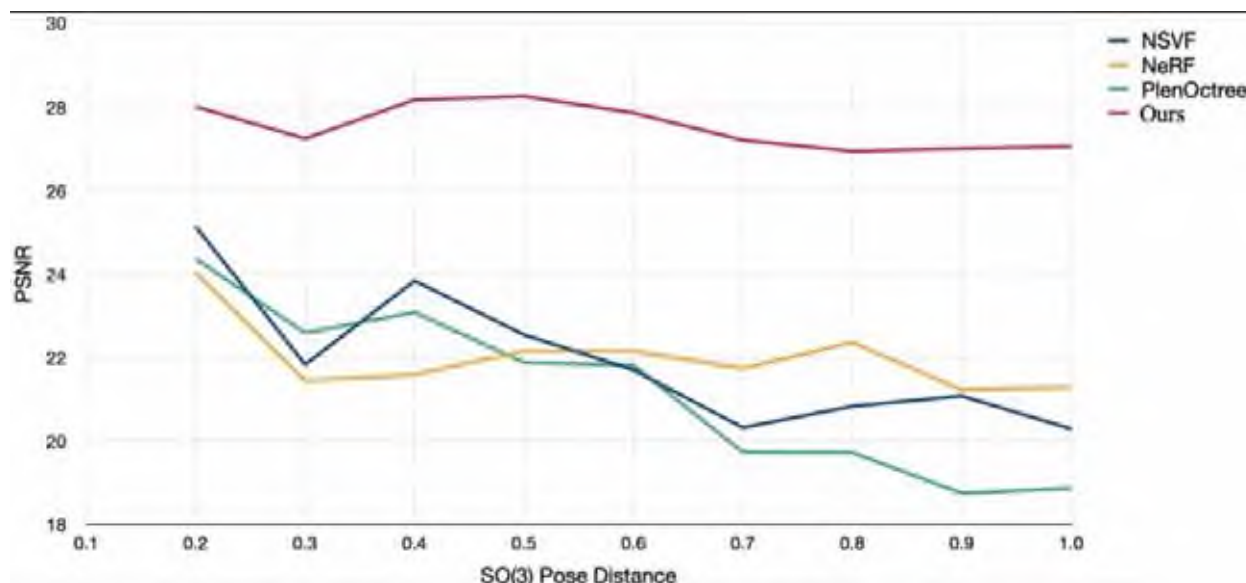
	训练时间 (V100)	PSNR
<u>NeRF</u>	~2 days	26.0
<u>Mip-NeRF</u>	~2 days	25.2
Ours	4h	28.6

视角鲁棒性问题是神经渲染的关键问题之一。拍摄的图片无法覆盖所有视角，当输出视角发生变化和拍摄视角有一定差异时，需要神经渲染有很好的泛化能力，支持新视角图片的生成。Object Drawer，提出RapNeRF技术，利用随机视角增强与平均视角embedding技术，改进了视角鲁棒性效果。

实验表明，随着视角差异的增大，NeRF等模型都出现了PSNR的大幅下降，而Object Drawer的结果，PSNR基本保持不变，case 分析可以看到视角差异很大的时候，Object Drawer也能够给出高清图像，很好的解决了这一问题。



拍摄视角与渲染视角示意，蓝色为拍摄视角，红色为渲染视角



视角差异度变化，PSNR变化

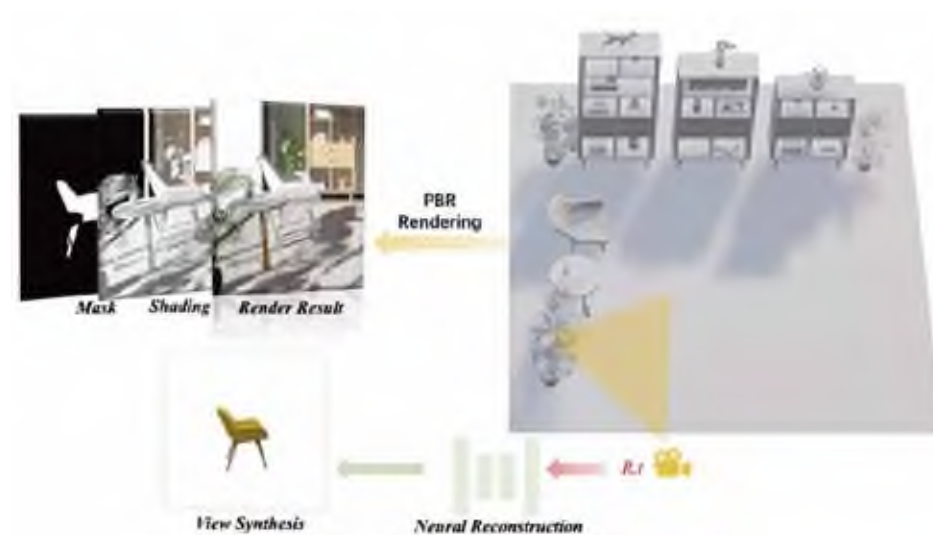
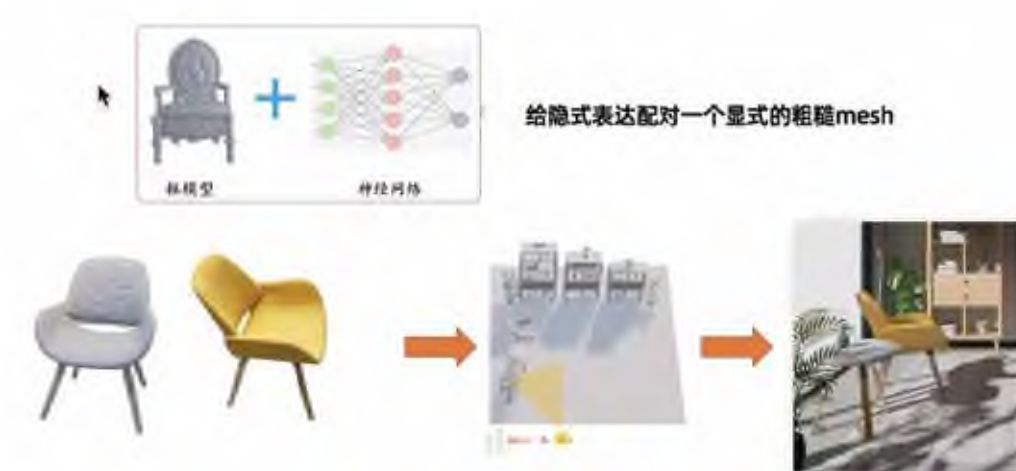


视角差异时渲染效果放大对比

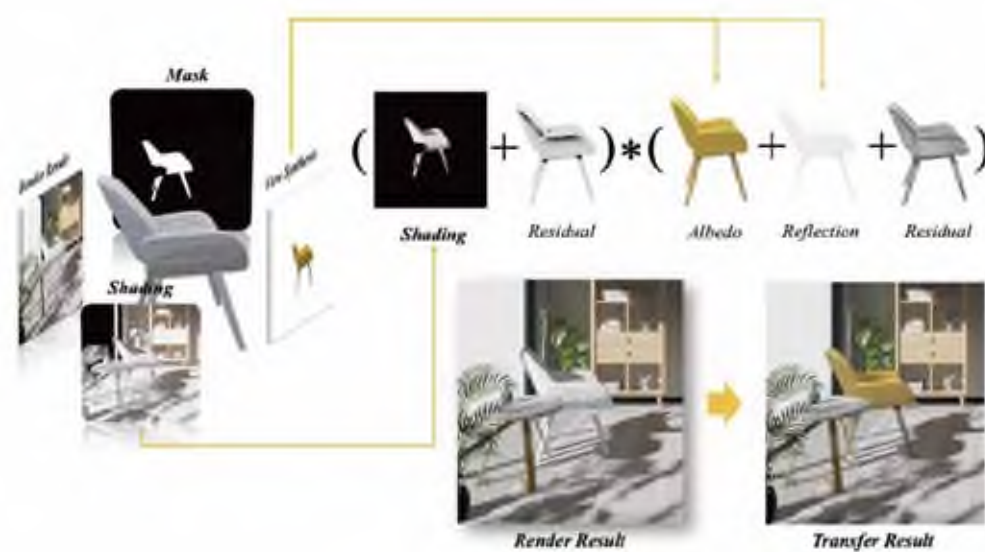
三维模型表示

三维模型，需要能够支持各种三维应用，包括三维布局、渲染出图、三维互动等等。神经辐射场只能支持任意角度的渲染，使得应用受到很大的限制。

Object Drawer提出了将隐式表达与三维网格模型融合的解决方案。具体方案为，从隐式表达中提取模型的显式空间几何表达，建立mesh模型，并且通过texture mapping、材质识别等方法，形成可以应用于现有渲染引擎的三维粗模，三维粗模和神经网络共同表示一个商品。一方面，三维粗模可直接导入图形学工具，如标准高精CAD模型一般用于三维场景设计等现实应用。另一方面，神经渲染可以得到物体任意视角的高清渲染图。



三维粗模渲染



光照迁移效果

光照迁移

为了反应物理渲染作用于三维粗模的光照效果，如反射，阴影等，我们提出LighTNet，将物理渲染反应在物体上的逼真光照效果迁移到神经渲染生成的物体视角图上。LighTNet以三维粗模PBR渲染得到粗糙shading、mask、depth等信息以及神经渲染结果为输入，预测相应的shading和神经渲染的修正值，修正后的shading和神经渲染可以通过本征融合得到具有光照效果的神经渲染结果；进一步用得到的带有光照效果的高清合成图替代物理渲染结果中的粗模投影可以实现逼真的场景渲染效果。在实验数据的结果表明，新方法能够适应各种复杂光源条件以及细节的阴影效果的迁移，视觉效果远超其他替代方案，在3D-FRONT测试数据集平均PSNR为30.17。



光照迁移建模效果示意

纹理细节还原

对于商品三维模型来说，纹理细节还原度非常重要，通常需要达到照片级的还原才能支持实际渲染相关产品。目前的NeRF系列研究虽然可以较高精度的perserve物体appearance，但是无法恢复纹理细节，如布料线条。Object Drawer优化了模型表达，在大幅度加速模型训练时间的同时，第一次做到了高清精细纹理还原，具体的效果如图



原始拍摄图



Ground Truth



NeRF : PSNR 26.0



Object Drawer : PSNR 28.6



Mip-NeRF : PSNR 26.0

不同建模技术所产生的纹理细节对比

Object Drawer对神经辐射场技术进行了全面的改进和升级，实现了商品自动化高精度建模流程，并且支持电商商品模型应用，取得了很好的效果。相信不久的将来，Object Drawer将会推动商品三维模型覆盖率不断提升，为万物3D化带来可能，并加速XR相关技术的发展

• ACM MM论文放榜！淘系技术内容互动算法团队4篇论文入选！

- 作者：淘系技术
- 出品：淘系技术

国际多媒体顶级学术会议（ACM MM2021）论文接收名录公开！淘系技术内容互动算法团队4篇论文入选！恭喜恭喜~

国际多媒体学术会议（ACM MM）是计算机学科公认的多媒体领域和计算机视觉领域的国际顶级会议，也是中国计算机学会（CCF）推荐的A类国际学术会议。ACM MM研究领域覆盖图像、视频、音频、人机交互、社交媒体等多个主题，本次 ACM MM2021 一共收到 1,942 篇论文申请，最终入选 542 篇论文（约 27.9% 的接受率）。

淘系技术部内容互动算法团队，聚焦机器学习、视觉算法、NLP算法，端侧智能等领域，依托淘系数十亿级的视频数据，业务上支持淘宝直播、逛逛和点淘，有丰富的业务场景和技术方向，不断探索和衍生颠覆型互联网新技术，团队成员来自海内外知名高校，近两年参加CVPR竞赛获得4项冠军，累积在计算机视觉顶会期刊（如CVPR、TPAMI、TIP等）上发表论文10余篇，技术成果获得国家科技进步二等奖。

本次 ACM MM2021 会议，该团队一共 4 篇论文被接收，并有相关技术创新点在淘系业务场景中的应用。后文将详细介绍各篇论文创新点以及落地使用。

NO.1

题目

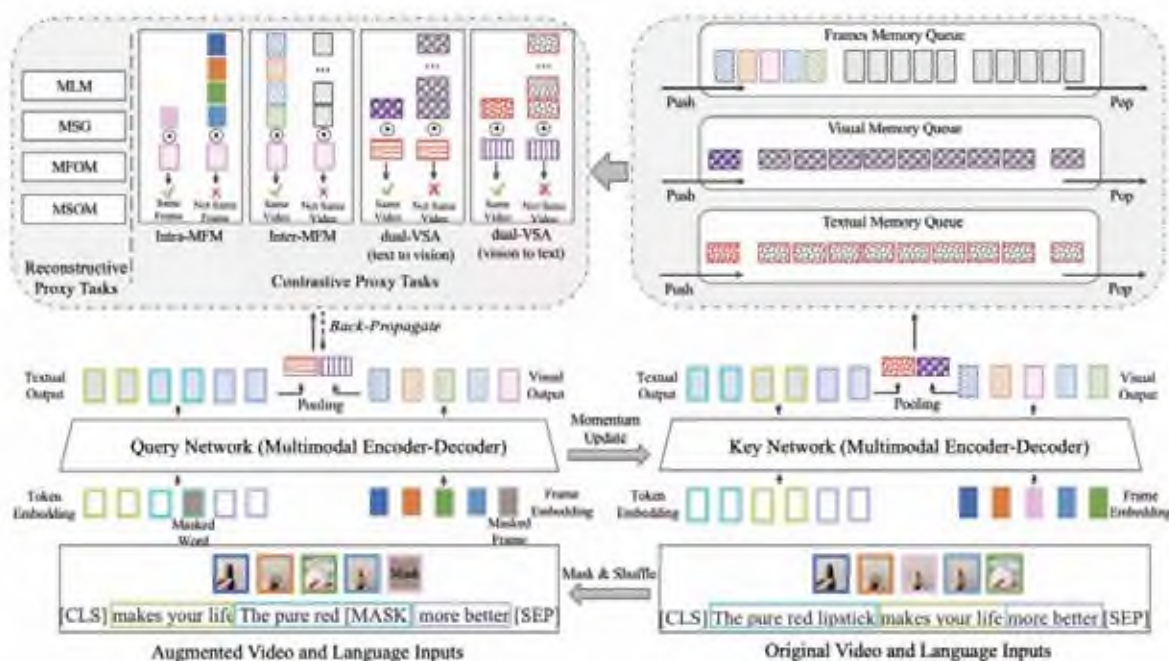
Understanding Chinese Video and Language via Contrastive Multimodal Pre-Training
基于对比多模态预训练理解中文视频和文本

论文创新点&对于行业的影响：

预训练模型在自然语言处理领域、视觉领域乃至多模态领域已经取得了巨大的成功。本文聚焦于多模态领域中的视频-文本的联合预训练策略，尤其针对中文视频和文本。针对视频-文本进行预训练主要存在以下挑战：

第一，和静态图像不同，视频拥有动态的时空序列关系，直接将图像-文本的预训练方法移植到视频-文本领域并不足以捕捉这些复杂的关系信息；第二，预训练模型中广泛存在的视频-文本对齐任务和其它基于掩码的重建任务存在冲突；第三，大规模、高质量的中文视频-文本数据集的缺乏限制了预训练模型在中文领域的发展。

由此，本文提出一个基于重建和对比学习任务的多模态预训练模型VICTOR，并建立千万数量级的高质量中文视频-文本数据集。VICTOR以Transformer为主体，设计出基于重建和基于对比学习的七个任务训练模型。基于重建的任务包括掩码语言建模、掩码句子生成、掩码帧序列建模和掩码句子序列建模四个任务，充分捕捉视频和文本的序列信息和交互信息；基于对比的任务包括对偶的视频-文本对齐、视频内的掩码帧对比学习和视频间的掩码帧对比学习三个任务，在避免简单的视频-文本对齐任务会融合不确定的多模态信息的同时，增强视频内的时空信息融合。VICTOR模型拥有上亿级参数，在构造的千万数量级的淘系视频-文本数据集中进行预训练，并在多个下游任务（如视频文本匹配、视频推荐、标题生成）获得了SOTA的性能提升。VICTOR模型的设计和提出，有效促进了预训练在中文视频-文本领域的进展，并可在多个视频相关业务（如视频推荐、视频分类等）广泛应用。



Victor 模型的总体框架：包含模型设计和自监督任务

论文相关技术在淘系实际场景的应用：

我们将VICTOR预训练的视频特征应用到内容检索、推荐、分类、直播等多个领域。各个应用场景均对比场景中已服务的Strong Baseline，具体来说：

- 1.内容推荐--逛逛内容推荐。效率保持稳定下，3天新发内容占比提升22.81%，冷启动UCTR +4.29%，PCTR +4.72%；
- 2.内容检索--淘宝经验。跨模态检索，保障相关性评测基础下，无结果率由3.23%降至0.95%
- 3.内容分类--逛逛内容分类。图文分类精度相对提升3.94%（60.97%→63.37%），视频分类精度相对提升7.33%(51.99%→55.80%)
- 4.物体检测与匹配 -- 直播看点审核。全品类检测精度相对提升4.83%（89%→93.3%），美妆难例检测精度相对提升8.05%(75.8%→81.9%)

论文阅读/下载链接

<https://arxiv.org/abs/2104.09411>

NO.2

题目

Pre-training Graph Transformer with Multimodal Side Information for Recommendation
用于推荐系统的融合多模态信息的图预训练Transformer

作者

刘勇，杨栗森，雷陈奕，王国鑫，唐海红，张举勇，孙爱欣，苗春燕

论文创新点&对于行业的影响

在个性化推荐领域，尤其是短视频的推荐领域，多模态信息发挥着重要的作用。有效利用item的多模态信息，如文本、视觉等信息，可有效提高推荐的性能，缓解冷启动问题。目前存在的融合多模态信息的推荐模型，都是端对端的基于特定任务的模态融合，消耗资源的同时限制了模型的泛化。另外，在推荐领域，item之间存在着各种相关性（如基于标签的语义相关、基于行为的用户兴趣相关等）。为节省资源，提高模型利用率，同时捕捉item之间的相关性，本文提出一种基于多模态信息融合的图预训练框架PMGT，在捕捉item相关性的同时，指导item多模态信息的融合，并且预训练后的item特征，可应用到多种下游任务，避免了在每个特定任务都要重新融合模态信息的资源浪费和时间消耗。

PMGT首先根据item的相关信息，构建出一个item多模态图，其中图的节点为item，边反应了item之间的关系（如被相同用户交互过的item建立边），每个节点的特征由item的多模态特征构成。对图中的每个节点，我们设计出高效并行的采样方法MCNSampling，从图中采样出若干与其相关的节点组成节点序列，并使用基于多

样性的transformer框架聚合节点特征，缓解模态融合的冗余性。最后使用基于图结构重建的任务和基于节点特征重建的任务指导相关节点融合和节点自身的多模态信息融合。将PMGT在Amazon和MovieLens公开数据集上预训练并测试，和最新的图预训练模型相比，达到SOTA性能。PMGT使用图的方式指导item的多模态信息的融合，并有效捕捉item之间的相关性，让预训练并不局限于item自身，增强了预训练的item特征的表达能力，可适用于多种下游任务和领域。

论文相关技术在淘系实际场景的应用

在淘系的短视频推荐领域，我们基于短视频tag信息建立拥有400万节点、4亿边的视频多模态图，并将PMGT预训练后的特征直接应用到短视频的召回阶段，7天新内容占比提升7%。之后可将预训练特征应用到排序阶段，甚至其它的业务场景（如视频分类），并且可将PMGT作为基础框架，使用特定的任务微调模型，达到效果的进一步提升。

论文阅读/下载链接

<https://arxiv.org/abs/2010.12284>

NO.3

题目

Shape Controllable Virtual Try-on for Underwear Models
(SC-VTON: 针对内衣模特的形状可控的虚拟试衣系统)

作者

高鑫，刘振江，冯尊磊，申成吉，欧开日，唐海红，宋明黎

论文创新点&对于行业的影响

我们提出了一种形状可控的虚拟试衣网络(SC-VTON)，针对内衣模特的试衣任务，使用融合了模特和服饰信息的GAT网络来生成形变后的服饰图片。除此之外，我们在SC-VTON中加入控制点来达到服饰的形状控制。更进一步，通过增加Splitting Network和Synthesis Network，我们可以使用服饰-模特的pair对数据优化模型，同时将任务泛化到常规的2D虚拟试衣任务。我们的方法能够做到精准的服饰形状控制。同时与其他方案相比，我们的方案能够生成纹理逼真的高分辨率图片，并且能够在实际应用中落地。这是业内首个将图注意力网络应用到虚拟试衣任务，同时能够做到精准可控的服饰形变。

论文相关技术在淘系实际场景的应用

服饰是淘系最重要的类目，虚拟试衣作为一种新颖的互动展示方式，为用户带来创新体验，为商家创造新的品牌展示方式。从20年元旦开在手淘拍立淘、扫一扫、云主题等公域场景上线了"虚拟试衣间"产品：提供不同身材的模特供用户挑选，支持几十万件服饰的在线实时试穿。"虚拟试衣间" PV20-30W，UV10W，二跳页人均停留时长2min，平均试衣件数12件。此外，运营同学利用虚拟试衣产品功能，在微博发起"一天试穿500件奢侈品"话题营销活动，曝光2.4亿，讨论量15.1万，产品得到了商家和线上用户认可。

论文阅读/下载链接

<https://arxiv.org/pdf/2107.13156.pdf>

NO.4

题目

TransRefer3D: Entity-and-Relation Aware Transformer for Fine-Grained 3D Visual Grounding
(TransRefer3D: 基于实体-关系可知的Transformer模型的细粒度3D视觉指代定位)

作者

何岱岚，赵禹昇，罗钧宇，惠天瑞，黄少飞，张爱喜，刘偲

论文创新点&对于行业的影响

本文提出了一种基于Transformer的模型来抽取3D场景中物体之间的多模态上下文，从而建模更具判别力的特征来定位被指代物体。

该模型的每一层中主要包括两个模块：

- 1.实体可知的注意力模块。该模块将语言中的实体信息与视觉实体特征相匹配，提取符合语言描述的实体特征；
- 2.关系可知的注意力模块。该模块将语言中的关系信息与视觉实体间成对的关系特征进行匹配，增强符合关系描述的实体特征。该模型在两个细粒度3D视觉指代定位基准数据集上取得了当前最优效果。

论文相关技术在淘系实际场景的应用

细粒度3D视觉指代定位任务目前在淘系业务中没有实际应用，未来可在视频结构化信息提取、智能机器人控制和人机交互等方面有广泛的潜在应用场景。本文所提出的模型可以辅助智能机器人更好地理解人类用户的指示语言与视觉信息的对应关系，从而在真实的3D场景中对物体实现准确的定位，为下游的复杂任务提供技术基础。

论文阅读/下载链接

http://colalab.org/media/paper/mm21_transrefer3d_camera_ready.pdf

结语

获得以上论文收录的淘宝内容互动算法团队，负责淘宝直播、视频和图文以及评价UGC的内容业务的算法研发，利用前沿的人工智能技术在内容业务上的知识挖掘，理解，认知，表示学习，智能剪辑和内容生成等课题和研究方向来打造阿里巴巴内容算法平台。

目前，该团队在大规模多模态预训练模型，多媒体内容的结构化和数字化，融合行业领域运营知识的内容图谱的构建，用户内容消费兴趣表征和认知推荐和内容创意的生成与互动（智能看点，智能摘要，合辑生成，虚拟试衣，3D直播间，虚拟主播等）等技术方向持续深耕。希望通过不断加深对用户在全域的兴趣理解和实时的感知，在内容领域建立起完善的分类和属性的标签体系，细到物品，场景，人物属性和声音风格，粗到内容类型，拍摄手法，泛化到内容的各层次的表征学习等多粒度的内容认知，实现内容的通用表征学习，提升多媒体内容搜索和推荐匹配的极致效率和体验，让淘宝成为消费者购买决策第一阵地，同时热烈欢迎对课题和方向有兴趣的同学的加入。

• CVPR2021竞赛结果出炉，阿里淘系多媒体算法包揽3项国际冠军

- 作者：淘系技术
- 出品：淘系技术

在刚刚落下帷幕的计算机视觉与模式识别领域顶级会议 CVPR 2021 上，各项国际挑战赛的竞赛结果已全部揭晓。

阿里巴巴淘系技术多媒体算法&视频内容理解算法团队，一举斩获



技术域包括图像描述生成、大规模头例级物体识别、多模态视频情感理解以及视频人物交互关系。

作为业界多媒体算法相关领域的领先团队，淘系技术的这支队伍着眼打造“端云一体、跨模态理解”的视频内容感知与理解算法平台；重点构建AR直播、3D数字场、内容智能生产、审核、检索和高层语义理解等技术领域；支持着淘宝直播、逛逛、点淘等淘系内容业务，通过自研的内容中台为整个阿里集团的内容业务提供能力支持。

以下是本次3项国际冠军的竞赛内容详情&我们的攻克方法。

冠军 VizWiz Image Captioning

题目

Workshop: CVPR 2021 VizWiz Grand Challenge Workshop
TRACK: Image Captioning

参赛者

宏黎，宏吉，咏亮，玉琦，少麟，定人

技术领域

图像描述生成

比赛背景介绍

VizWiz Grand Challenge比赛从2018年开始举办，旨在利用计算机视觉技术帮助有视觉障碍的盲人“看见”世界。

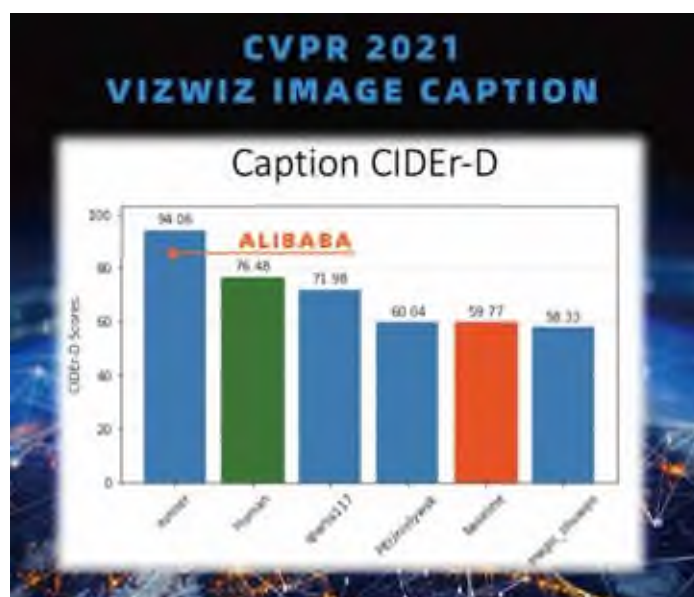
该任务的输入是由盲人拍摄的图像，输出是图像的描述。

与其他Image Caption数据不同，该比赛的数据是由视觉有障碍的盲人拍摄，图像质量比较差，因此任务难度更高。

我们的成绩

我们以 CIDEr-D score 94.06的成绩取得第一名，远超第2名的 CIDEr-D score 71.98。

总分也超过去年冠军 IBM 的CIDEr-D score 81.04。



任务难点

该任务主要有两个难点：

- 1.图像质量较差：含各种室内、室外的场景，同时由于拍摄者视觉障碍，拍摄图像会出现失焦模糊、拍摄不全、遮挡等问题；
- 2.许多图像描述需要理解图像中的文字，不同物体，颜色等信息，需要OCR，物体检测等细节理解能力。

我们通过以下途径解决这些困难

- 1.针对VizWiz数据图像特性，采用swin-transformer抽取图像的grid feature替换object feature，以充分表征不同图像区域的特性；
- 2.考虑到OCR及物体信息会对image caption生成产生正向引导，我们抽取了OCR及目标检测类别信息，作为特征补充；
- 3.并不是所有的图像都含有OCR信息，我们采用多种模型互补融合，用视觉模态模型强化那些不含OCR的数据，用视觉+文本（OCR+物体类别）多模态模型强化含有OCR信息较丰富的数据；
- 4.针对多种模型生成的结果，考虑到最终的衡量指标是CIDEr，我们通过self-cider、ocr maximization 多种策略融合的方式进行结果融合。

可应用的场景

Image captioning需要视觉理解与文本生成，是视觉和NLP任务的结合，可应用于互联网产品的内容标题自动生成，另外也可以帮助盲人和视觉受损用户提升他们对世界的感知能力。

赛事链接

- 1.workshop:
<https://vizwiz.org/workshops/2021-workshop/>
- 2.challenge:
<https://eval.ai/web/challenges/challenge-page/739/overview>

冠军

Herbarium 2021 – Half-Earth Challenge

题目

Workshop: The Eight Workshop on Fine-Grained Visual Categorization

Task: fine-grained plant species identification

参赛者

元年，兰杞，琉潇，有邻，暖雨，济宇，篱悠

技术领域

大规模实例级物体识别

比赛背景介绍

Herbarium 2021属于 CVPR2021 FGVC8 workshop的一项比赛，该workshop针对实例级细粒度识别问题，已经连续举办第八届。

Herbarium 2021 比赛数据集为从多个大型植物园收集的美洲、大洋洲等半个地球的6.5W类2.5M张植物样本图片，用于训练植物识别算法，辅助植物学家进行植物识别，发现和保护新物种。

该数据集存在长尾分布，样本数目最少的类别仅有3张样本，同时，不同植物间视觉非常相似，同时同一植物的不同样本有较大差异，给实例级识别带来很大挑战。

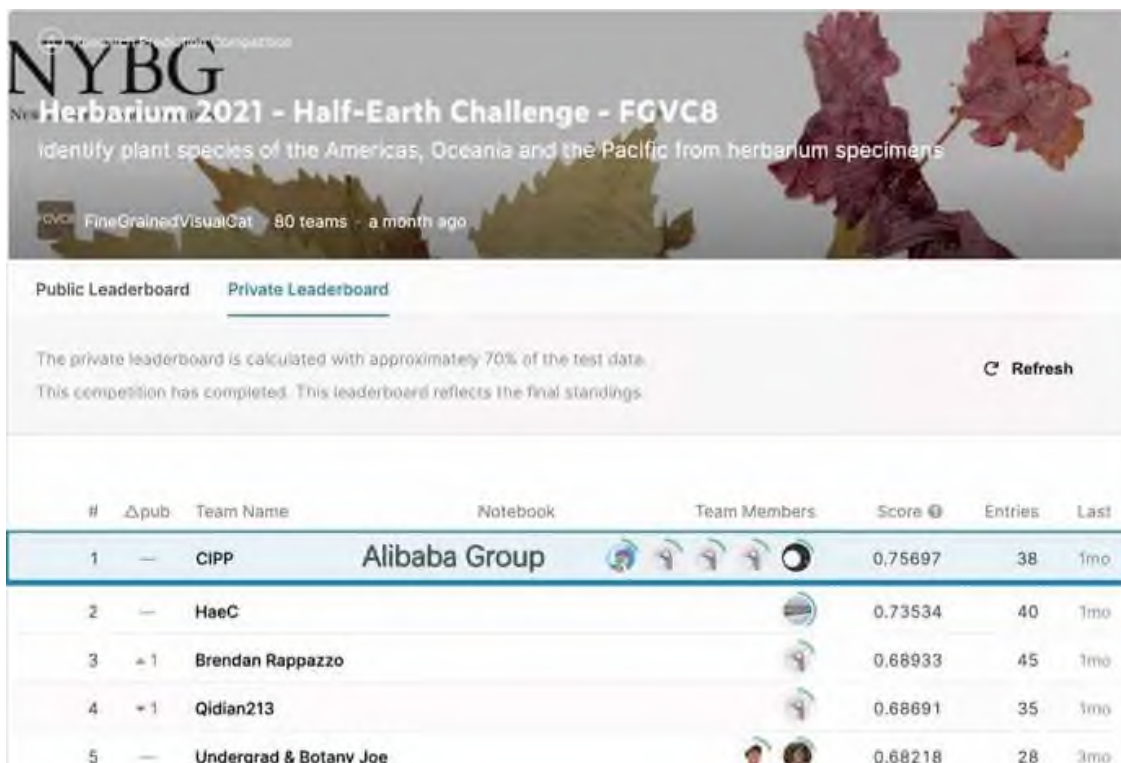
我们的成绩

我们以F1 score 0.757的成绩在该项比赛上取得了第一名的成绩，远超第二名的0.735和第三名的 0.689。

任务难点

该任务主要存在以下两个难点：

- 1.植物种类多、类别细，不同植物间视觉非常相似，而同一植物的不同样本存在差异，导致类间易混淆，区分难度大；
- 2.数据集的样本分布不均衡，存在长尾分布，样本数目最少的类别仅有3张样本，如何提升长尾类别准确率至



NYBG
Herbarium 2021 - Half-Earth Challenge - FGVC8
Identify plant species of the Americas, Oceania and the Pacific from herbarium specimens
FineGrainedVisualCat - 80 teams - a month ago

Public Leaderboard Private Leaderboard

The private leaderboard is calculated with approximately 70% of the test data.
This competition has completed. This leaderboard reflects the final standings.

Refresh

#	Δpub	Team Name	Notebook	Team Members	Score	Entries	Last
1	—	CIPP	Alibaba Group		0.75697	38	1mo
2	—	HaeC			0.73534	40	1mo
3	▲1	Brendan Rappazzo			0.68933	45	1mo
4	▲1	Qidian213			0.68691	35	1mo
5	—	Undergrad & Botany Joe			0.68218	28	3mo

关重要。

我们通过以下途径解决这些困难

将自然场景中实例级植物识别问题转换成大规模细粒度特征表达问题，提出self-attention pooling进行局部特征增强提升特征表达能力；通过引入Imbalanced Sampler和自适应类别loss解决类别分布不平衡问题；此外，基于混合精度的大规模多机多卡训练能力，实现近三百万数据规模下的快速迭代能力。

实现高效万级在线难样本挖掘，极大提升了特征在复杂场景下的泛化能力。最终凭借领先亚军2.2%的优势，一举获得冠军。

可应用的场景

实例级的细粒度识别技术可辨别物体间细微的视觉差异从而实现精细的物体识别，广泛应用于商品识别、动植物识别、行人识别、地标识别等领域。

赛事链接

1.Workshop:

<https://sites.google.com/view/fgvc8/home>

2.Challenge:

<https://sites.google.com/view/fgvc8/competitions/herbariumchallenge2021>

3.Kaggle leadboard:

<https://www.kaggle.com/c/herbarium-2021-fgvc8/leaderboard>

冠军 ActivityNet Home Action Genome Challenge

题目

Workshop: International Challenge on Activity Recognition

Task: Home Action Genome Challenge

参赛者

少麟，廖越（北航），咏亮，叶盈，篱悠，刘偲（北航）

技术领域

视频人物交互关系

比赛背景介绍

Home Action Genome Challenge今年首次在CVPR2021 ActivityNet Workshop举办，由斯坦福大学李飞飞教授课题组主办，比赛提供了一个大规模多视角的视频数据集，通过多模态视频分析，检测视频中存在的人物交互关系。

我们的成绩

我们以准确率76.5%的成绩在该项比赛上取得了第一名的成绩，大幅领先第二名的68.4%和第三名的65.7%。

任务难点

该任务主要有3个难点：

- 1.数据集的日常家居场景复杂，人体和物体的目标检测难度大
- 2.人物关系包含动作关系和空间关系，关注不同的视觉特征
- 3.每一组人体和物体都存在多个人物关系，评估时必须完全正确才计一次正确



Home Action Genome Challenge 获奖证书

任务难点

该任务主要有3个难点：

- 1.数据集的日常家居场景复杂，人体和物体的目标检测难度大
- 2.人物关系包含动作关系和空间关系，关注不同的视觉特征
- 3.每一组人体和物体都存在多个人物关系，评估时必须完全正确才计一次正确

我们通过以下途径解决这些困难

- 1.采用更好的检测模型：我们采用Swin-Transformer和ResNeSt为backbone的性能SOTA的检测模型，并通过多种数据增强策略训练和多尺度融合推理，提升目标检测的准确率。
- 2.强化人物关系的视觉特征：我们设计了融合两阶段和一阶段关系检测网络的方案，首先将Swin-Transformer融入两阶段关系检测网络进行端到端训练，然后改进一阶段关系检测网络，直接提取<人，物>二元组，再通过cascade结构判定关系，给出<人，物，关系>三元组。策略上，我们通过视觉特征判定动作关系，空间位置作为输入辅助判定空间关系。
- 3.基于统计偏置的生成策略：我们在生成最终的人物交互关系组时，采用了融合<人，物，关系>三者共生概率和统计偏置加权的多种策略。

可应用的场景

视频人物交互关系检测，检测视频中动态的<人，物，关系>的结构化信息，未来可应用于视频信息结构化，人机交互等应用场景。

赛事链接

1.Challenge:

<https://homeactiongenome.org/results.html>

2.Workshop:

<http://activity-net.org/challenges/2021/challenge.html>

除上述3项冠军以外，我们也在 Hotel-ID 2021-Hotel Recognition Challenge 比赛中取得第二名，以及在 Evoked Expressions from Videos (EEV) Challenge 比赛中取得第三名的好成绩，在多媒体算法领域内排名前列。

淘系技术多媒体算法参赛团队表示：“随着视频流量在媒体表征方面占比越来越高，视频信息对个人和平台而言，都存在信息过载的问题。视频内容的多维度结构化表示将会是视觉领域的热点研究方向之一。未来我们也会融合文本、语音和视觉等多模态的信息做好视频内容理解，让用户可以更多看到自己喜欢的内容，降低用户信息选择的时间成本，为用户带来更好的视觉体验。”

技术人的通用干货tips

• 程序命名的原则与重构

- 作者：玄苏
- 出品：淘系技术

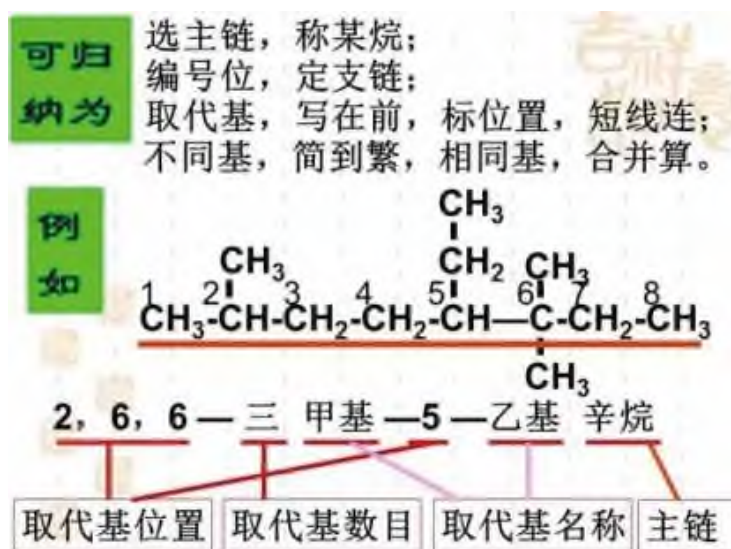
命名是对事物本质的一种认知探索，是给读者一份宝贵的承诺。糟糕的命名会像迷雾，引领读者走进深渊；而好的命名会像灯塔，照亮读者前进的路。命名如此美妙，本文将一步步揭开它的神秘面纱！

命名来源生活



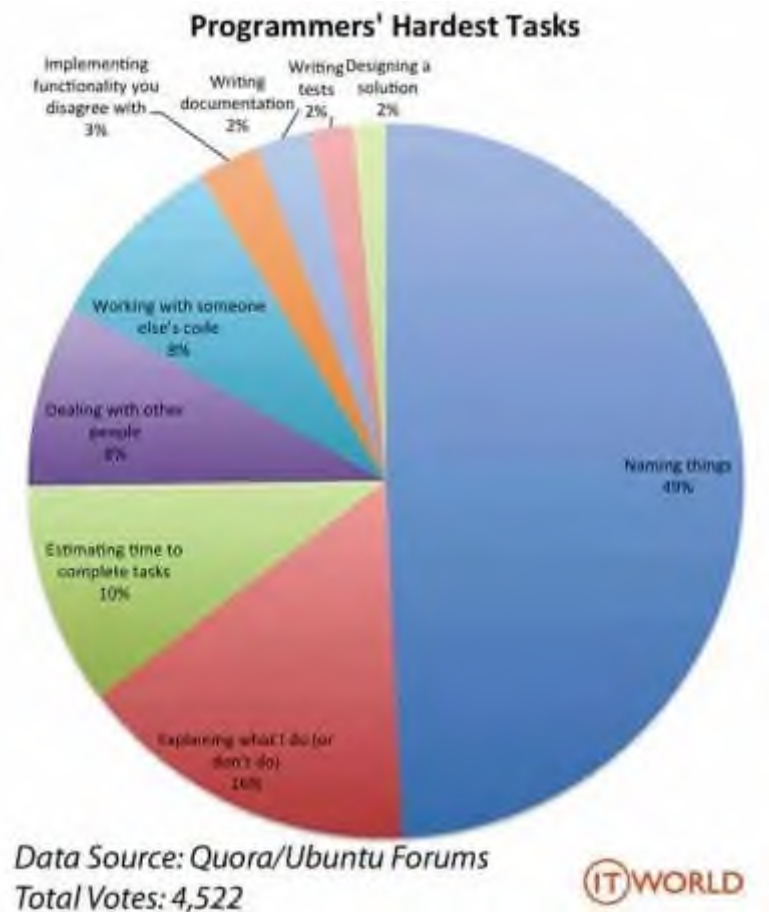
从左到右：正三角形，正方形、正六边形 正表示边长相等，从而得到正XXX的边长一定是相等的。

这些事物的特征比较明显，容易被人们所记忆。但是也有一些比较难以命名，比如化学有机物：



相比于化学有机物，软件世界的事物更加繁多，命名将更加困难。

Phil Karlton (菲尔·卡尔顿)曾说过：在计算机科学中只有两件困难的事情：缓存失效和命名规范。



命名一直是软件领域的难题，好的命名能够信达雅：

译事三难：信、达、雅。——严复

信：含义准确

达：通顺流畅

雅：简明优雅

命名的坏味道

查看下面代码，说出其含义：

```
1 public List<int[]> getThen(){
2     List<int[]> list1 = new ArrayList<int[]>();
3     for ( int[] x: theList )
4         if (x[0] == 4)
5             list1.add(x);
6     return list1;
7 }
```

问题不在于代码的简洁度，而是在于代码的模糊度：即上下文在代码中未被明确体现的程度。

- 1.theList 中是什么类型的东西？
- 2.theList 零下表条目的意义是什么？
- 3.值4的意义是什么？
- 4.我如何使用返回的列表

问题的答案没体现在代码段中，可那就是它们该在的地方。

再看看重命名后的代码：

```
1 public List<int[]> getFlaggedCells(){
2     List<int[]> flaggedCells = new ArrayList<int[]>();
3     for ( int[] cell: gameBoard )
4         if (cell[STATUS_VALUE] == FLAGGED)
5             flaggedCells.add(cell);
6     return flaggedCells;
7 }
```

注意，代码的简洁性并未触及。运算符和常量的数量全然保持不变，嵌套数量也全然保持不变。但代码变得明确多了。

站在使用者的角度，还可以更近一步吗？

虽然getFlaggedCells一名表明方法会返回FlaggedCells，但是返回的数据结构并未表达出来：

```
1 public List<Cell> getFlaggedCells(){
2     List<Cell> flaggedCells = new ArrayList<Cell>();
3     for ( Cell cell: gameBoard )
```

```
4     if (cell.isFlagged())
5         flaggedCells.add(cell);
6     return flaggedCells;
7 }
```

能否看出代码的两处差异点？

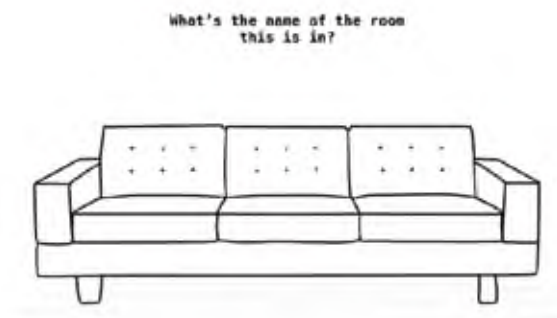
稍微仔细点观察，就会发现有两个点被改进：

- 1.int[] -> Cell：将数据进行建模，赋予其含义
- 2.cell[STATUS_VALUE] == FLAGGED ==> cell.isFlagged()
 - 建模后可以竟然可以将这条语句语义化，可读性增强了；
 - STATUS_VALUE、FLAGGED 都被隐藏至Cell，更加了内聚；

只要简单改了以下名称，就能轻易知道发生了什么，这就是命名的力量。

命名的游戏

首先来做一个游戏，游戏名为“我们住在哪个房间？”，如下会为你提供一张图片，请你说说看这是什么房间：



从上面的图片不难看出，这肯定是客厅。基于一件物品，我们可以联想到一个房间的名称，这很简单，那么请看下图。



基于这张图片，我们可以肯定的说，这是厕所。通过上面两张图片，不难发现，房间的名称只是一个标签属性，有了这个标签，甚至我们不需要看它里面有什么东西。这样我们便可以建立第一个推论：

推论1：容器（函数）的名称应包含其内部所有元素

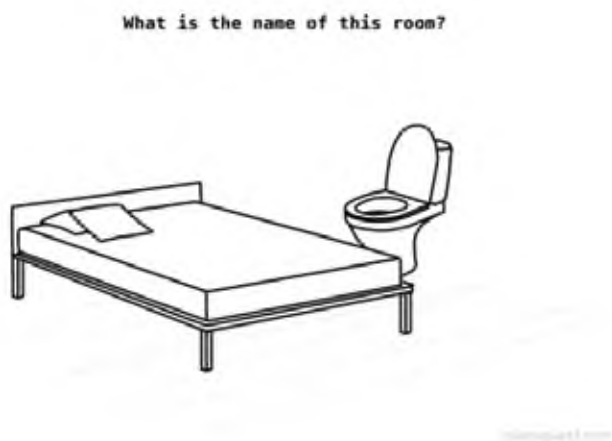
如果有一张床？那么它就是卧室。我们也可以反过来进行分析。

问题：基于一个容器名称，我们可以推断出它的组成部分。如果我们以卧室为例，那么很有可能这个房间有一张床。这样我们便可以建立第二个推论：

推论2：根据容器（函数）的名称推断其内部组成元素

现在我们有了两条推论，据此我们试着看下面这张图片。

问题 3/3

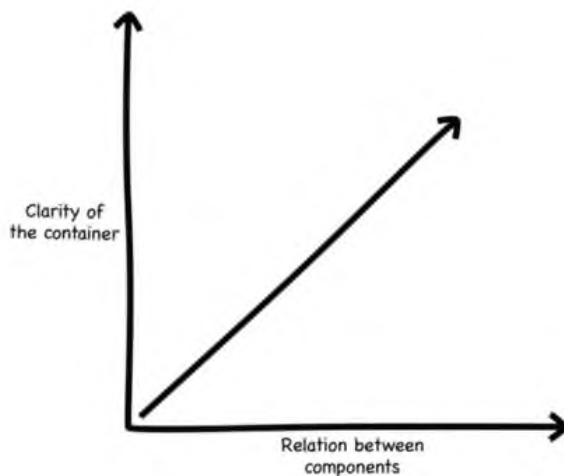


好吧，床和马桶在同一个房间？根据我们的推论，如上图片使我们很难立即做出判断，如果依然使用上述两条推论来给它下定义的话，那么我会称它为：怪物的房间。

这个问题并不在于同一个房间的物品数量上，而是完全不相关的物品被认作为具备同样的标签属性。在家中，我们通常会把有关联的，意图以及功能相近的东西放在一起，以免混淆视听，所以现在我们有第三条推论：

推论3：容器（函数）的明确度与其内部组件的密切程度成正比

这可能比较难理解，所以我们用下面这一张图来做说明：



alexquest.com

如果容器内部元素属性关联性很强，那么更容易找到一个用来说明它的名字；反之，元素之间的无关性越强，越难以描述说明。

属性维度可能会关系到他们的功能、目的、战略，类型等等。关于命名标准，需要关联到元素自身属性才有实际意义。

在软件工程方面，这个观点也同样适用。例如我们熟知的组件、类、函数方法、服务、应用。罗伯特·德拉奈曾说过：“我们的理解能力很大程度与我们的认知相关联”，那么在这种技术背景下，我们的代码是否可以使阅读者以最简单的方式感知到业务需求以及相关诉求？

命名的原则

名副其实

命名应该描述其所做的所有事情(或者它的意图)。

当读者读到上文命名的坏味道里面讲的例子中getThen()，并不能理解 getThen()的意图是什么？是获取什么呢？getFlaggedCells()就能比较准确地表达出来。

```
1 // 糟糕的命名
2 public List<int[]> getThen();
3 // 好的命名
4 public List<Cell> getFlaggedCells();
5
```

```
6
7 // 糟糕的命名
8 private Date userCacheTime;
9 // 好的命名
10 private Date customerStayTotalTime;
```

避免误导

避免留下掩藏代码本意的错误线索。

别用accountList来指称一组账号，除非它真的时List类型。

List一词对于程序员来说有特殊含义。如果包纳账号的容器并非真实一个List, 就会引起错误的判断。所有建议用accountGroup 或 bunchOfAccounts, 甚至是 accounts都会好一些。

避免变量名使用小写字母l和大写字母O。

这样的拼写方式容易误导读者或者让读者花较大的力气去辨别。

有意义的区分

如果同一作用范围内有多个命名，最好让它们之间有区分度。

```
1 public static void copyChars(char a1[], char a2[]){
2     for (int i = 0; i < a1.length; i++){
3         a2[i] = a1[i];
4     }
5 }
```

这里参数a1, a2是依义进行命名的，完全没有提供正确的信息，没有提供导向作者意图的线索。

如何进行有意义的区分呢？

如果参数改为source 和 destination，这个函数的命名就更符合其用途。

```
1 public static void copyChars(char source[], char destination[]){
2     for (int i = 0; i < source.length; i++){
3         destination[i] = source[i];
4     }
5 }
```

```
4    }  
5    }
```

准确使用对仗词可以提高命名的区分度。

命名时遵守对仗词的命名规则有助于保持一致性，从而也可以提高可读性。像first/last这样的对仗词就很容易理解；而像FileOpen() 和 _lclose() 这样的组合则不对称，容易使人迷惑。下面列出一些常见的对仗词组：

add/remove	increment/decrement	open/close
begin/end	insert/delete	show/hide
create/destory	lock/unlock	source/target
first/last	min/max	start/stop
get/put	next/previous	up/down
get/set	old/new	

尽量不要使用info/data为结尾去命名类名或变量名。

info和data的含义过于宽泛，导致没有额外的信息量，反而增加了读者的阅读成本，得不偿失。

风格一致

让同一个项目中的代码命名规则保持一致。

比如：

- 1.每个class的Logger取名为logger、log还是LOGGER，取哪个名字均可，但是需要保持项目统一
- 2.类属性的getter/setter方法的命名统一。getName()还是name()均可，但是需要保持项目统一
- 3.注释的风格统一。

```
1  /**  
2   * 用户的姓名  
3   */
```

```
4    public String userName;  
5  
6    /** 用户的姓名 */
```

抽象一致

让同一作用域内的变量或方法具有相同的抽象。

```
1  public class Employee {  
2      ...  
3      public String getName(){...}  
4      public String getAddress(){...}  
5      public String getWorkPhone(){...}  
6  
7      public boolean isJobClassificationValid(JobClassification jobClass){...}  
8      public boolean isZipCodeValid(Address address){...}  
9      public boolean isPhoneNumberValid(PhoneNumber phoneNumber){...}  
10  
11     public SqlQuery getQueryToCreateNewEmployee(){...}  
12     public SqlQuery getQueryToModifyEmployee(){...}  
13     public SqlQuery getQueryToRetrieveEmployee(){...}  
14     ...  
15  
16 }
```

查看这个类，看看它有几个抽象层次？

通过函数名可以查看：

- 1.getName、getAddress、getWorkPhone 都是获取 Employee 的主要属性，符合Employee的抽象
- 2.isJobClassificationValid 是 校验JobClassification 对象是否有效，属于JobClassification的抽象层次，与Employee 无关
- 3.isZipCodeValid是校验Address的合理性，属于Address的抽象层次，而Address是Employee 的属性，不是一个抽象层次。isPhoneNumberValid 同理。
- 4.getQueryToCreateNewEmployee/getQueryToModifyEmployee/getQueryToRetrieveEmployee 看似与Employee 有关，但是这里暴露SQL语句查询细节，是实现细节，层次比Employee要低。

多个不同层次的方法会让这个类看起来非常怪，就像将晶体管、芯片零件、手机放在一个台面上一样。

```
1 public class Employee {
2     ...
3     public String getName(){...}
4     public String getAddress(){...}
5     public String getWorkPhone(){...}
6
7     public String createEmployee(...){...}
8     public String updateEmployee(...){...}
9     public String deleteEmployee(...){...}
10    ...
11
12 }
```

命名建模

如果在一个项目中，发现有一段组装搜索条件的代码，在几十个地方都有重复。这个搜索条件还比较复杂，是以元数据的形式存在数据库中，因此组装的过程是这样的：

1. 首先，我们要从缓存中把搜索条件列表取出来；
2. 然后，遍历这些条件，将搜索的值填充进去；

```
1 //取默认搜索条件
2 List<String> defaultConditions = searchConditionCacheTunnel.getJsonQueryByLabelKey(labelKey);
3 for (String jsonQuery : defaultConditions) {
4     jsonQuery = jsonQuery.replaceAll(SearchConstants.SEARCH_DEFAULT_PUBLICSEA_ENABLE_TIME,
5                                     String.valueOf(System.currentTimeMillis() / 1000));
6     jsonQueryList.add(jsonQuery);
7 }
8 //取主搜索框的搜索条件
9 if (StringUtils.isNotEmpty(cmd.getContent())) {
10     List<String> jsonValues = searchConditionCacheTunnel.getJsonQueryByLabelKey(
11         SearchConstants.ICBU_SALES_MAIN_SEARCH);
12     for (String value : jsonValues) {
13         String content = StringUtil.transferQuotation(cmd.getContent());
14         value = StringUtil.replaceAll(value, SearchConstants.SEARCH_DEFAULT_MAIN, content);
15         jsonQueryList.add(value);
16     }
17 }
```

简单的重构无外乎就是把这段代码提取出来，放到一个Util类里面给大家复用。然而我认为这样的重构只是完成了工作的一半，我们只是做了简单的归类，并没有做抽象提炼。

简单分析，不难发现，此处我们是缺失了两个概念：一个是用来表达搜索条件的类——SearchCondition；另一个是用来组装搜索条件的类——SearchConditionAssembler。只有配合命名，显性化的将这两个概念表达出来，才是一个完整的重构。

重构后，搜索条件的组装会变成一种非常简洁的形式，几十处的复用只需要引用SearchConditionAssembler就好了。

```
1 public class SearchConditionAssembler {  
2     public static SearchCondition assemble(String labelKey) {  
3         String jsonSearchCondition = getJsonSearchConditionFromCache(labelKey);  
4         SearchCondition sc = assembleSearchCondition(jsonSearchCondition);  
5         return sc;  
6     }  
7 }
```

由此可见，提取重复代码只是我们重构工作的第一步。对重复代码进行概念抽象，寻找有意义的命名才是我们工作的重点。

因此，每一次遇到重复代码的时候，你都应该感到兴奋，想着，这是一次锻炼抽象能力的绝佳机会，当然，测试代码除外。



语境通用化

别抖机灵

如果你使用的命名来自一个比较冷门的语境，比如俗语或者俚语，不知道这个语境的人将很难理解它的含义。

如：用whack来标识kill，wsBank来标识网商银行

使用问题领域的名称

如果不能用程序员熟悉的术语命名，就采用从所涉及问题领域而来的名称，至少维护代码的程序员就能去请教领域专家了。这样至少问题域的专家能清晰理解开发者命名的语境，读者可以询问领域专家或者查询领域词汇含义。

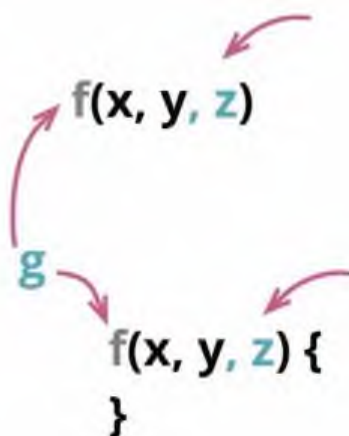
以消息中间件领域为例：topic、message、tag、offset、commitLog

名词	含义
topic	消息的类型
broker	消息处理代理者
message	消息主题
tag	Topic下的次级消息类型
offset	消费者的消费进度
commitLog	消息的存储文件

改名

如果子程序名称、类名、变量 含糊不清或者名不副实时，就需要对这个变量进行改名或者重构。

改变函数声明



好的命名让读者一看看出函数的用途，而不必查询实现代码。

动机

函数名：

改进函数声明的小技巧：先写一句注释描述这个函数的用途，再把这句注释变成函数的名字。

函数的参数：

函数的参数列表阐述了函数如何与外部世界共处，是函数和函数使用者共同的依赖，这其实也是一种耦合

- 1.最小使用原则：函数的参数列表正是函数所依赖的，不会依赖没有用到的信息

例如：一个函数的用途是把某人的电话号码转换成特定的格式，并且这个函数的参数是一个人，那么我就没法用这个函数来处理公司的电话号码。如果把函数接收的参数由“人”改成“电话号码”，这段处理电话格式的代码就能被更广泛地使用。

- 2.根据函数的意图引入函数参数。

如果这个函数的意图只是将电话号码转换成特定的格式，那么只引入电话号码是合适的。

如果这个函数的意图是得到“人”的电话号码格式并且他的号码是通过自身的其他属性合成，那么应该引入“人”。

关于如何选择正确的参数，没有简单的规则可循，需要视具体情况而定。

做法

常用的重构做法有两种：简单式做法 和 迁移式做法

简单式做法：适用于一步到位地修改函数声明及其所有调用者。

- 1.如果想要移除一个参数，需要先确定函数体内没有使用该参数。
- 2.修改函数声明，使其成为你期望的状态。
- 3.找出所有使用旧函数声明的地方，将它们改为使用新的函数声明。
- 4.测试。

最好能把大的修改拆成小的步骤，所以如果你既想要修改函数名，又想添加参数，最好分成两部来做。比较幸运

的是，简单式做法一般可以用IDE工具直接重构完成。

实战：下列函数的名字太过简略

```
1 public long circum(long radius){  
2     return 2 * Math.PI * radius;  
3 }
```

将这个命名改得更加有意义一些：

```
1 public long circumference(long radius){  
2     return 2 * Math.PI * radius;  
3 }
```

迁移式做法：函数被很多地方调用、修改不容易或者要修改的是一个多态函数或者对函数声明的修改比较复杂

- 1.如果有必要的话，先对函数体内部加以重构，使后面的提炼步骤易于开展。
- 2.使用提炼函数（106）将函数体提炼成一个新函数。
- 3.Tip如果你打算沿用旧函数的名字，可以先给新函数起一个易于搜索的临时名字。
- 4.如果提炼出的函数需要新增参数，用前面的简单做法添加即可。
- 5.测试。
- 6.对旧函数使用内联函数（115）。
- 7.如果新函数使用了临时的名字，再次使用改变函数声明（124）将其改回原来的名字。
- 8.测试。

实战：还是刚才circum方法改名的例子

这个简略的函数名先不做修改。

```
1 public long circum(long radius){  
2     return 2 * Math.PI * radius;  
3 }
```

再新增circumference函数：

```
1 public long circumference(long radius){  
2     return 2 * Math.PI * radius;  
3 }
```

逐渐小步地将circum 方法的调用处改成circumference方法，每次修改都运行一下测试；如果测试成功，则提交此次修改进行一下修改，否则返回至上一步重新进行修改。这样及时中间出错，也能准确定位至某此修改，稳定推进重构，间接提高了重构的效率。

变量改名



动机

好的变量命名可以解释一段程序来干什么——如果变量名起得好的话。

机制

- 1.如果变量被广泛使用，考虑运用封装变量将其封装起来。
- 2.找出所有使用该变量的代码，逐一修改。

如果在另一个代码库中使用了该变量，这就是一个“已发布变量”（published variable），此时不能进行这个重构。

如果变量值从不修改，可以将其复制到一个新名字之下，然后逐一修改使用代码，每次修改后执行测试。

- 3.测试

范例

如果要改名的变量只作用于一个函数，对其改名是最简单的，直接使用IDE进行重命名即可。

如果变量的作用于不至于单个函数，重命名的风险就不太好把控，这时需要对变量进行封装。

变量初始化

```
1 int treeName = "untitled";
```

变量被修改

```
1 treeName = "bigtree";
```

变更被读取

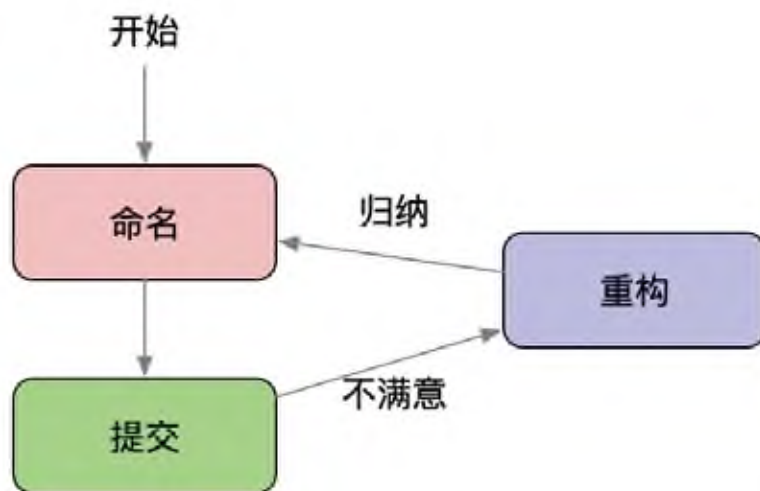
```
1 leftTree = treeName;
```

此时可以考虑采用封装变量进行完成

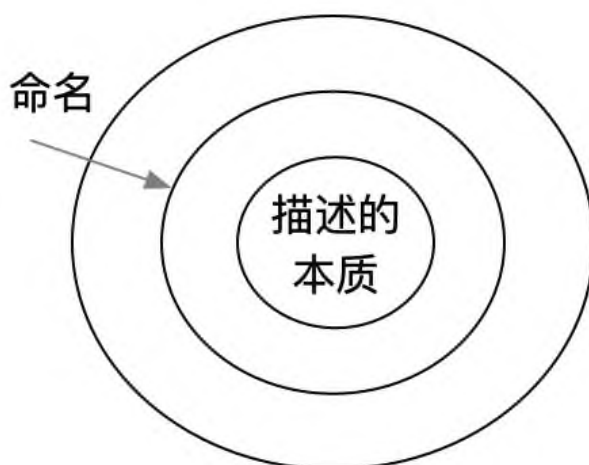
```
1 private int treeName;  
2  
3 public void init(){  
4     treeName = "untitled";  
5 }  
6  
7 public String getTreeName(){  
8     return treeName;  
9 }  
10  
11 public void setTreeName(String treeName){  
12     this.treeName = treeName;  
13 }
```

命名的过程

命名是一个迭代的过程。当你持续很长时间想不到比较好的命名时，不要掉入取名的陷阱，可以先用折中的命名commit掉或者重构这段程序。当你想到更合适的命名，毫不犹豫地去重构它。



命名是一个接近描述事物本质的过程。命名得越好，越容易接近描述事物的本质。。



取好名字最难的地方在于需要良好的描述技巧和共有文化背景。。

结语

好的命名是自解释的，读者不用了解程序实现的细节，就能知道程序实现的意图(契约式编程)。在项目实战中，有时候很难给一段子程序取到一个比较好的名字，这其实是程序在说话--让我干的事情太杂了，导致不知道我是用来干啥的。一般这种情况下，需要重构这段子程序，对齐进行职责拆分，分而治之。命名是门艺术，美在它的简单，美在它的明确，美在它的名副其实。

• 如何用“心流”提升编码工作效率？

- 作者：梧忌
- 出品：淘系技术

“心流”是人在做一件事情时投入到忘我的一种精神状态，常见于运动员、艺术家和科研人员身上。

“心流”如何影响工作？我们可否解决“心流”带来的相关问题？

我在学生时代是个狂热的 NBA 球迷，家里总是堆满了篮球报，工作后我对篮球热情开始退去，直到斯蒂芬·库里的出现。

斯蒂芬·库里改变了篮球比赛的打法。他身高1.91米，在 NBA 中属于矮个子球员，但只要他进入比赛状态，他几乎可以毫不费力地在对方半场上的任何位置命中三分球。他打球的方式让人赏心悦目。



他不仅带领金州勇士队夺得了 NBA 的总冠军，而且还颠覆了所有 NBA 球员和球队的比赛风格——他证明了不需要成为中锋的那样的大个子也能赢得比赛。斯蒂芬·库里之后，越来越多的球员和球队开始练习和多投三分球。

每个运动员都可以通过刻意地训练来找到自己的比赛状态，然后主宰比赛。

我们可以从斯蒂芬·库里身上学习到什么？“比赛状态”到底指的又是什么？

高效编码的秘诀：“心流”

20 世纪 70 年代，积极心理学领域的权威 Mihaly Csikszentmihalyi 定义了一种名为“心流”的状态，这在运动员、艺术家和科研人员身上可以经常看到。

Csikszentmihalyi 研究了是什么让人们感到快乐，他发现，人们在“心流”状态下最快乐。

大多数人都有过这样的经历：忽然抬起头，发现时间已经过去了很久，自己已经完成了一系列的工作。那种轻松富有成效的时间流逝和挥之不去的成就感，令人愉悦。

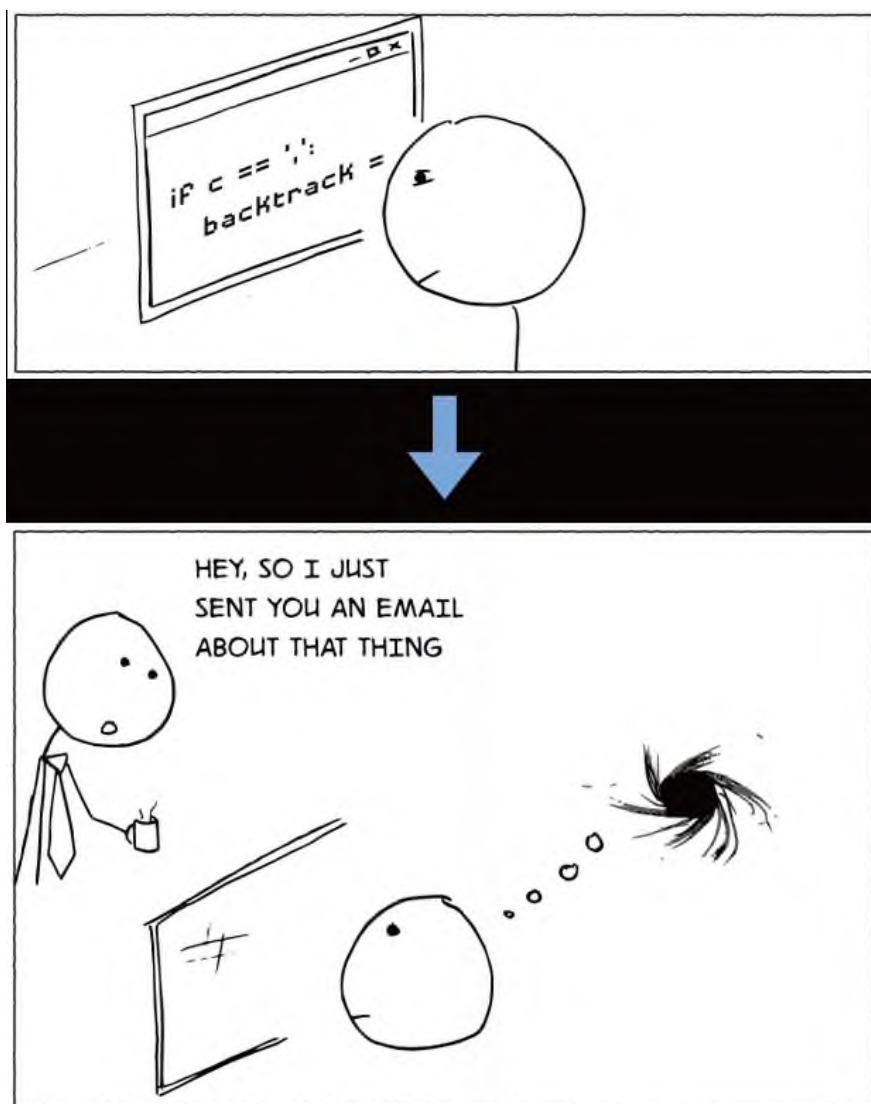
“心流”是高效编码的秘诀。但是要达到“心流”的状态并不容易，来看看阻碍我们进入“心流”状态的几个问题。



影响编码心流的问题

外界的干扰

这是一个重大而又经常出现的情况：突然的外界干扰，会破坏我在编码时的思绪，打乱我正在大脑中加载的逻辑。例如，我会被钉钉或其他应用程序的消息干扰，又或者是到身边来问问题的同事干扰。相信每个开发者都曾经遇到过这样的场景：



外界的干扰

会议是影响“心流”的另一个明显的问题，但可能大家没有注意到的是它们在会议时间之外的其他影响，如图所示：



通常在开会之前，我知道很快就要进入到会议当中，因此会中断掉编码，然后准备会议的头脑状态。会议结束后，还需要时间来消化内容、让自己恢复到正常的脑力状态。

因此，Facebook 的“周三不开会”倡议受到工程师们的普遍欢迎，大家都认为这将有助于提升开发人员的生产力。

但是我们是只关心一周中一天的编码时间吗？有没有想过，在标准工作时间，会议频繁的时候，我们真正输出了多少代码？

人的思维惯性

技术正在飞速发展，总有很多的东西需要研究和学习。我经常思考如何平衡花在学习和编码上的时间。

保持专注并不容易，需要克服种种的思维惯性。我经常陷入到这种拖延的惯性当中：看会儿某乎再开始吧，还可以顺便检查一下邮件……午饭后再开始写吧，毕竟还有半小时就到午饭时间了……

有时候“心流”的敌人就是自己。

需要意志力和纪律才能将全部工作分解为小的、可操作的工作单元，然后才可以开始编码。

解决这些问题的对策

设定编码时间

理解“心流”之后，关键是找到自己的「心流时刻」。可以借助在编辑器上的行为来捕获和分析编程活动的轨迹得出来。



图上的蓝色点颜色越深，代表在此时刻早键盘的输入越多。

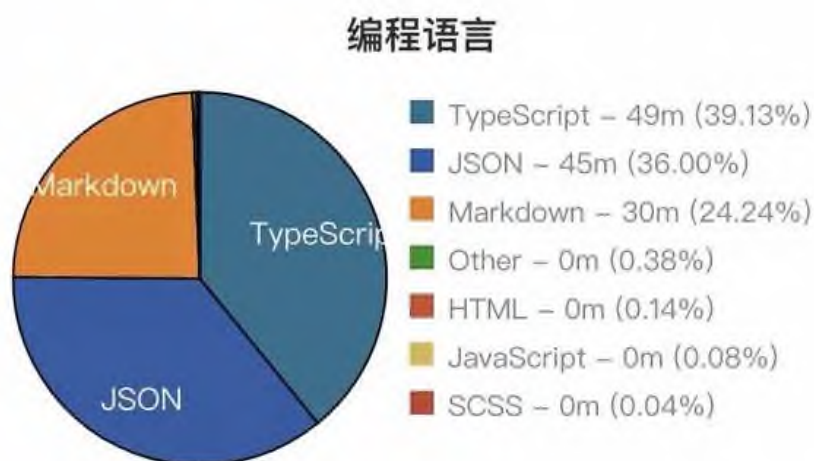
然后，你就可以开始通过工作日来设定自己的编码时间。开会、写代码、阅读，它们需要的状态和脑力是完全不同的，可以自己的情况来计划哪个时间段来执行这些任务。

将这些时间规划设置到你的工作软件（钉钉）上，让它们在这些时间段进入到免打扰模式，不再有应用程序的消息提示窗口，同时它还在提醒你的同事避免在这些时间段约你开会等等。

持续完善文档

会议往往是异步沟通无效的体现。另外，根据 Stack Overflow 的调研报告，几乎一半的开发者在遇到问题时会打电话给同事。为了减少会议和答疑，一个建议是持续地更新和完善文档：更好的文档可以帮助你更快地回答同事的问题，并减少被打扰的可能性。

你知道自己在写文档上的投入如何吗？或许看看你在编辑器上对文本文件的编辑时间投入可以得出一些结论。



高效会议并回顾

更有效的会议可以避免更多的会议。

明确会议主持，制定会议议程，做好会议记录，产出会议结论——这些都是高效会议的一些措施。

另外，有必要考虑一下你是否真的需要参加这个会议。斯坦福大学的研究表明，当会议的参与者只有 5 ~ 8 人时，效率最高。还有 Atlassian 的一篇文章说，近 40% 的员工会在会议期间昏昏欲睡。。。

最后，结合工作日历和编码热图，可以构建会议与编码时间的关系图。可以看看你到底是花了更多时间在开会还是在编码？会议时间的增加是怎么影响编码时间的？

克服思维惯性

每个人都需要一些动力来克服行为和思维惯性。可以问一下你自己：每天在八卦群上闲聊的花费了多少时间？是不是经常走神拿起手机刷朋友圈、短视频？如果可以记录并绘制你的编程活动图，那么：

- 你可以很快地看到你最近一天或一周的编码投入是否高于或低于自己的平均值。
- 你可以看到你每个时刻的编码输出情况，想想那些数据断层是因为你在思考还是走神了？



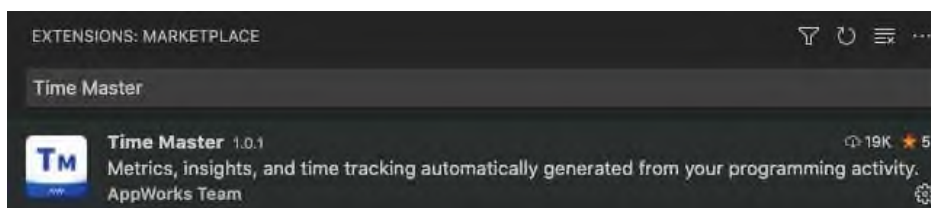
与自己对比来找到最好的自己，关注你的编码数据和设定目标可以提高自己的纪律性和产出。

Time Master

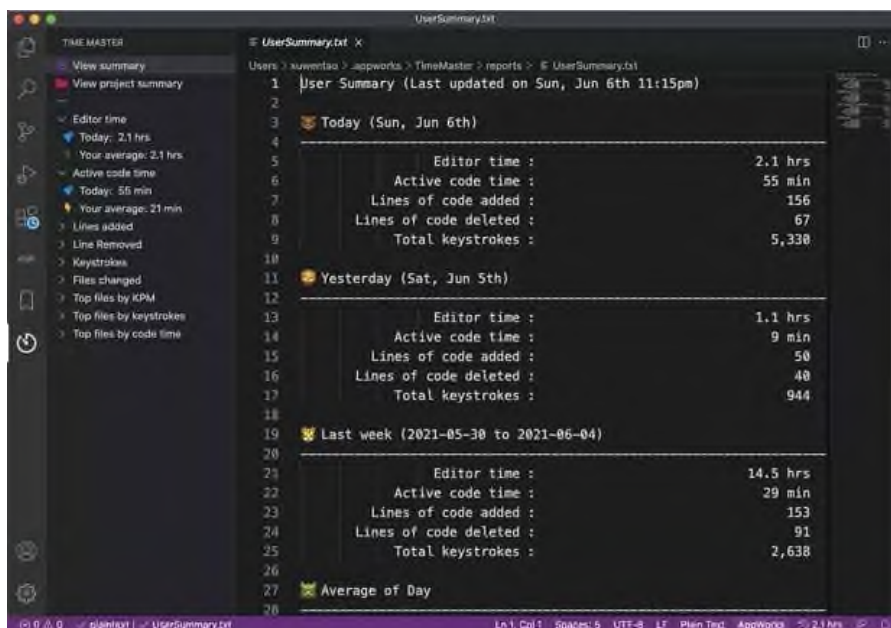
这就是为什么 AppWorks 打造 Time Master 的原因。软件正在改变世界，数字化正在影响着我们生活的方方面面，关于我们如何开发软件的数据也很重要。

Time Master 记录和可视化开发者的编程数据，帮助开发者发现并找到“心流”状态，实现高效编码。

这一切，完全是自动的。你只需要在你的 VS Code 上安装该插件即可：



然后你在编辑器端，就可以看到你的编程活动统计数据：



在阿里内网的线上站点，可以看到编程活动的趋势和分析：



除了帮助开发者进入“心流”状态，Time Master 还有更多的想象空间和其他意义。

越早编码越好？

Joel Spolsky, Trello 和 Stack Overflow 的创始人在他的博客中说道：

Sometimes I just can't get anything done.

Sure, I come into the office, putter around, check my email every ten seconds, read the web, even do a few brainless tasks like paying the American Express bill. But getting back into the flow of writing code just doesn't happen.

Spolsky 分享了一个简单的见解：“「立刻开始做」也许是提升生产力的关键”。

数据可以为生产力的提升提供强大的推动力。为了克服早上的惯性，我们可以做的，最简单的事情之一就是跟踪我们每天开始编码的时间：



为什么要跟踪编码开始时间？

开始编码的时间越早，当天编码的时间可能就越多；）越晚开始，查看电子邮件或社交媒体的次数越多，需要克服的惯性就越大。

你上班的第一件事是编码，还是浪费在查看社交媒体和电子邮件上？一个建议是：应该将工作的第一个小时花在高价值的事情上。

Code First

Code First 是开发人员或团队用来提高生产力的一种工作指导原则。

- Code First 意味着要快速原型制作以了解其工作模式而不是进行宽泛的研究和计划。
- Code First 意味着尽可能地进行编码以进入”心流时刻“，意味着将编码时间置于会议和其他时间之前，优先满足对编码时间的要求。
- Code First 与代码质量并不矛盾，可以与单元测试和严格的代码审查相结合。

一旦可以查看多天一周中特定某天的编码开始时间，开发者就可以设定目标以提高其一致性并践行 Code First。

代码重构率

开发者编写代码，然后重构一些代码，然后再重构一些代码。

就好像我写的这篇文章，我会先写一个版本，然后再次阅读，优化一些语句，添加或删除一些段落；最后我再检查一遍，修正一些错别字。最终我将产出一个经过审核的、质量更高的版本。

代码重构的情况与之类似。首先必须了解：代码重构是很正常的，或者说在研发过程中是健康的，这里讲的“重构”并不是推翻过去的一切。

问题是我们可以从跟踪代码重构中学到什么吗？——分析代码重构的过程或许可以帮助我们更好地进行编码。

代码重构率的定义

代码重构率是指在编写的代码总量（总代码）中被重写的代码量的占有率。

总代码可以分为三类：添加的字符、删除的字符和粘贴的字符。总代码可以通过按键和字符，数据的最小原子单位来计算，也可以使用代码行数来计算，这因编码风格、文本编辑器而异。

代码重构的原因

代码重构有被动的原因，例如：市场发生了变化，导致产品需要更新；产品经理更新了需求；我们发现架构或设计中存在严重缺陷；又或者我们在代码中发现需要修复的错误。

当然也有主动的原因。优秀的开发者通常会快速产出原型，以了解正在创作的软件的知识范畴以便学习，这也是践行 Code First 工作原则的一种体现。

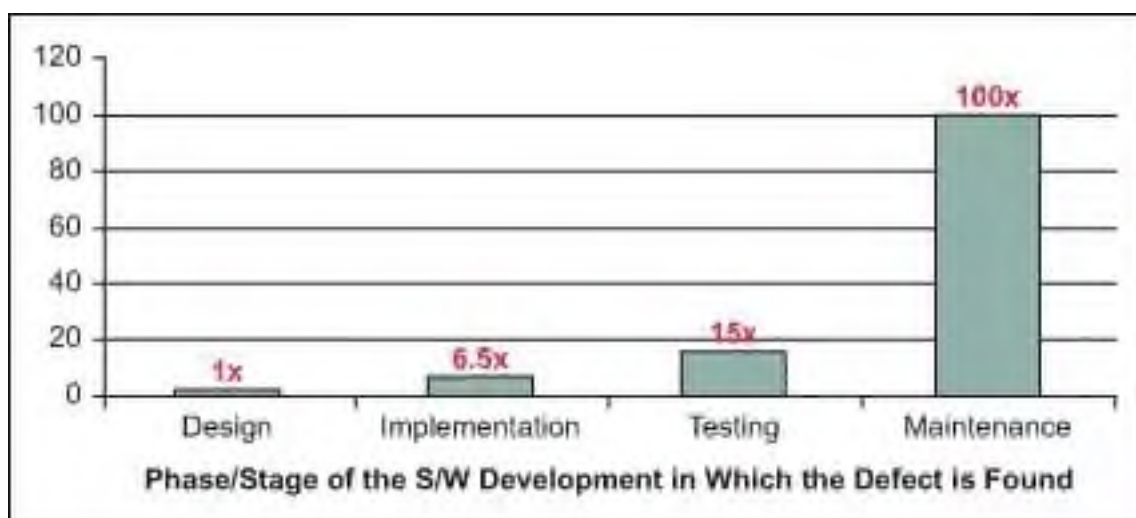
在开源技术被广泛应用的今天，丰富的示例可以帮助开发者快速开始某项编码工作，开发者经常需要集成和粘贴代码，然后在集成的过程中删除一些代码——这也是代码重构的一种原因。

此外当我们改进代码时，例如消除冗余、提升性能或提高可读性，也会进行代码重构。

总结一下：

- 好的代码重构原因：学习、集成、优化；
- 差的代码重构原因：市场变化、需求变化、架构或设计问题、代码缺陷；

代码重构的成本随着时间的推移而变化。在修复缺陷时，随着软件开发生命周期的进展，成本会急剧上升。在生产阶段修复错误的成本可能比在设计中修复它的成本高 100 倍。



最早且修复成本最低的代码重构分析是在提交到 Git 仓库之前，即在 IDE 中进行的代码重构。这就是为什么我们需要追踪编程活动的原因之一。

在代码重构中学习

查看代码重构相关的数据可以挖掘程序或产品的大量信息。例如，软件开发生命周期后期的高代码重构率可能表明存在质量和生产力问题。此外，如果将代码库中所有文件的代码变动可视化，还可以了解开发工作都花在了哪里。

文件

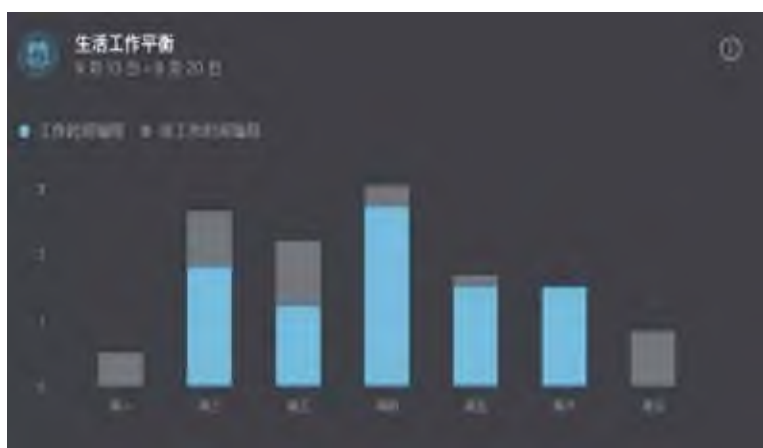
25 mins	...time-master/package.json
14 mins	...material-service/src/index.ts
8 mins	...src/views/quickEntries.ts
6 mins	...src/views/quickEntriesTree.ts
6 mins	...master/package.nls.zh-cn.json
5 mins	...time-master/src/index.ts
5 mins	...iceworks-app/README.md
3 mins	...iceworks-app/package.json
2 mins	...iceworks/CHANGELOG.md
2 mins	...master/src/createTreeView.ts
2 mins	...iceworks-app/src/extension.ts

了解自己代码重构率，与团队成员对比，可以让你看看自己的工作方式是否有改进的空间。

工作与生活的平衡

工作与生活的平衡一种健康的身心状态，两者的平衡可能跟我们在其中投入的时间有关。改善工作与生活的平衡可以提高幸福感和生产力，同时防止过度疲劳。

Time Master 采集了我们的编码数据，因此可以了解到我们在工作时间和工作时间之外的编码时间，帮助我们了解工作与生活的平衡情况。将编码热图和工作日历结合，还可以查看会议时间和变是如何影响编码时间的，从而合理安排你的日程来改善工作与生活的平衡。



例如可以通过几个维度来看工作生活的平衡：在工作时间段内进行的编码时间的百分比、工作时间段内的编码时间变化趋势、又或者跟自己过往的数据或者同事的数据进行对比。上面的示例图表是按照按星期几来显示的，跟同期数据对比会更有意义，还可以深入了解开发者在周末编码投入。

• 如何做好一份程序员的工作汇报ppt?

- 作者：旭伦
- 出品：淘系技术

这个回答分为4个部分：

- 1.首先是分析和拆解问题，为什么对于大多数程序员来说，做好一份工作汇报ppt是件不容易的事情？
- 2.第二部分是针对问题的解决方法，主要是工具和逻辑思维两方面的训练
- 3.第三部分，既然是面向程序员的问题，那么程序员做ppt的优势在哪里
- 4.最后是总结与展望，随着人工智能技术的发展，未来可能做好ppt不再是个困难



为什么对于大多数程序员来说，做好一份工作汇报PPT是件不容易的事情？

大家首先思考一个问题：大家有没有发现，不同人做的ppt水平相差特别大，但是画的流程图、思维导图、UML类图时序图之类的却差异比较小？这其中有什么不同？

另外，大家再思考一下：为什么设计师团队就很少有如何做好ppt的困惑？

再比如，假如你是一个程序员，现在需要画流程图，有Photoshop和Visio两种软件，你会选择哪一种？

我们再拆解一下对于程序员来说，写ppt时遇到的困难：

- 1.找不到好的模板
- 2.不知道该用什么样的字体

- 3.不知道去哪里找图
- 4.不懂如何排版
- 5.在ppt里不知道如何画流程图
- 6.在ppt里不知道如何显示代码
- 7.....

更绝望的是，我们发现，很多ppt高手的ppt里画得美轮美奂的幻灯片，并不是用了什么高级工具或者模块，而是用简单的线条与图标自己绘制的。

比如下面的幻灯片，如果应用模板只改文本框中的字当然很容易，但是想绘制这样的模板出来，需要自己使用各种基本图形、图片、渐变等基本操作来实现，这样的门槛对于一般程序员来说是不低的。



综上，我们可以体会到，程序员做ppt困难的根本原因是，powerpoint/keynote这样的工具根本就不是为程序员设计的，而是为有艺术设计技能的人准备的。

而程序员做工作汇报，需要艺术设计技能童子功这件事情，明显就是个伪命题。所以我们总结起来一句话，导致程序员用ppt做工作汇报困难的原因是：ppt工具不适合程序员用。简称：工具不好用。

针对工具不好用，有两种解决方案

1. 改变工具
2. 改变自己

对于通用人群来说，找到更加好用的工具也不容易，通过训练提升自己的ppt水平，使自己有驾驭ppt工具的能力是最主要的方法。

对于程序员来说，其实是可以考虑换用更趁手的工具的。

下面我们分别介绍一下。

通用解决方法

上一节我们清楚了，ppt不是给程序员设计的工具，靠直觉是做不好的，需要专门学习和反复练习。

当我们学习了之后，可能发现情况比我们想象的还悲观一些，我们想要找字体、图片来让ppt更好看，但是发现字体大部分都是收费的，高清大图也基本都是有版权的。这也很好理解，设计师也是要恰饭的。

这些困难告诉我们，要做好一个ppt，功夫在诗外，平时就需要多积累，没有太好的捷径可以走。

1. 对于powerpoint/keynote的基本操作要熟悉，值得学习一门专门的课程或者看一本比较全面的书
2. 对于字体、图片、国标等资源，平时要多注意收集网址，靠搜索引擎在接到任务时再准备往往来不及
3. 见到别人好的ppt设计的时候多模仿，能够完美还原好的设计，填充进自己的内容，这就比只用模板进步一个层次了

素材是设计师主导的，这不是程序员的特长。但是从排版开始，设计师的技能只是锦上添花，更重要的是逻辑思维。大家都看过不少场面特效一流的大片，但是就是讲不好一个故事。我们排版也是为了目的服务的。

逻辑思维听起来比较虚，我们将其拆解成三点：

1. 结构化表达
2. 用事实说话
3. 用数据说话



我们聚焦在题主的目标“工作汇报”上。既然是汇报，肯定是有目的，要么是立了功希望奖励，要么是遇到困难需要协助，要么是讲解为什么这么做寻求理解与支持。

不管是一是怎样，可以先把结论讲出来，采用下面的结构化的方式：



除了事实之外，还要有数据分析证明你的理由可以推导出你的结论。

这种逻辑推导靠直觉也是不够的，需要专门学习一些相关知识。

比如说，逻辑推理方法有演绎法和归纳法。

演绎法被马斯克的凡事要从第一性原理推导的理论传播得很广泛了。

归纳法的局限性大家也了解得越来越多了。但是归纳法也是有其成熟的理论体系的。比如我们想分析因果关系，有著名的“穆勒五法”。



对于程序员来说，这些方法可能都不熟悉，但是大家要知道，这都是MBA入学考试逻辑题的考点。

讲太多了分散主题，我们就以剩余法为例，它的精髓其实就是那句著名的“排除一切不可能的，剩下的即使再令人难以置信，那也是真相”。不管你是从福尔摩斯，还是柯南，还是唐人街探案中听到的。

发挥程序员特长

有了逻辑框架之后，剩下的就是工具的问题了。终于，发挥程序员特长的时候来了。

ppt现在基本上都是设计师独立完稿，但是，在网页上几乎没有是设计师把设计稿放上去就可以的。前端程序员使用的IDE一般是vscode，也不是所见即所得的设计工具。

所以，为什么一定要用powerpoint或keynote？我们是不是可以用前端技术来写工作汇报？

比如，reveal.js(<https://revealjs.com/>)就是用前端技术写slides的一种选择。

举例来说，reveal.js对于代码的支持就很不错。

我们来看个小例子，代码高亮加上相关动画效果，可是用ppt不容易做出来的：



Reveal.js使用的技术是HTML和CSS，像下面这样：

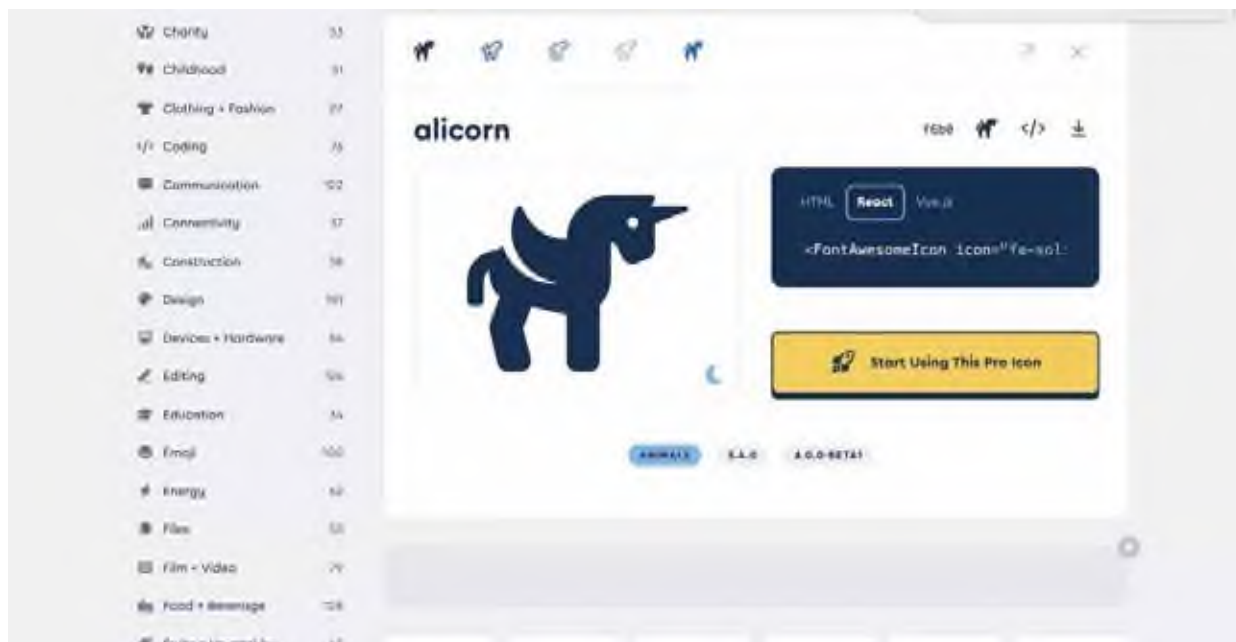
```
1 <section>
2   <h3 style="{color: #ffec3d;}">推荐系统的冷启动</h3>
3   <ul>
4     <li>利用热门数据</li>
5     <li>利用用户注册信息</li>
6     <li>利用第三方数据</li>
7     <li>利用物品内容属性</li>
8   </ul>
9 </section>
```

有了css作为武器，一下子我们腰杆就硬起来了。

比起ppt模板，我们css的各种样式库明显各丰富嘛，比如这两年比较火的tailwindcss：



比如，我们css有图标库：



未来展望

最后我们展望一下未来，ppt要做的事情是将信息可视化，其实只有逻辑思维是机器无法替代需要我们认真学习的，而将整理好的逻辑转化成比较美观的slides，这应该是越来越强的人工智能应该完成的工作。

隔空喊话imgCook团队，能不能搞个pptCook出来？

• 2021阿里淘系工程师推荐书单

- 作者：淘系技术
- 出品：淘系技术

读书有点像机器学习从初始参数开始的局部最小化。

无论是数学、科学、技术、管理，还是历史、文学，读书的过程，就是根据一维的文字流，在立体的思维空间中重新建构。

这种建构的意义，将我们自身和世界连接了起来。

（阿里巴巴淘系技术部资深算法专家 乐田）

上个月，我们推出了2020淘系技术年货《技术人的百宝黑皮书》。

（公众号后台回复“黑皮书”即可获取电子书下载链接）

在书的第三部分，我们开列了来自阿里工程师倾情推荐的本年度书单，涵盖了技术硬核参考、商业思维培养以及文化、科普、工具方法等多个类别，受到不少好评。

以下是关于这份书单的详细扩充介绍，希望能够帮助技术人员精进技术实力，提高思维能力和眼界认知。

（收藏起来，我的懒癌还可以抢救一下）

类别一：技术硬核参考



《深度探索C++对象模型》 [美] Stanley B · Lippman

这本书探索“对象导向程序所支持的C++对象模型”下的程序行为，对于“对象导向性质之基础实现技术”以及“各种性质背后的隐含利益交换”提供一个清楚的认识。



《你不知道的JavaScript》 [美] Kyle Simpson

JavaScript语言有很多复杂的概念，但却用简单的方式体现出来（比如回调函数）。“你不知道的JavaScript”系列就是要让不求甚解的JavaScript开发者迎难而上，深入语言内部，弄清楚JavaScript每一个零部件的用途。本书介绍了该系列的两个主题：“作用域和闭包”以及“this和对象原型”。掌握了这些知识之后，无论什么技术、框架和流行词语，你都能轻松理解。



《深入理解ES6》 [美] Nicholas C. Zakas

深入理解ES6的特性对于所有JavaScript开发者而言至关重要，在可预见的未来，ES6中引入的语言特性会成为JavaScript应用程序的主流特性。



《程序员修炼之道》 [美] Andrew Hunt / David Thomas

《程序员修炼之道》由一系列的独立的部分组成，涵盖的主题从个人责任、职业发展，直到用于使代码保持灵活、并且易于改编和复用的各种架构技术。



《架构整洁之道》 [美] Robert C. Martin

《架构整洁之道》是创造“Clean神话”的Bob大叔在架构领域的登峰之作，围绕“架构整洁”这一重要导向，系统地剖析其缘起、内涵及应用场景，涵盖软件开发完整过程及所有核心架构模式。



《支持向量机导论》 [美] 克里斯蒂亚尼

本书是第一本综合介绍支持向量机（SVM）的书籍，它从机器学习算法的基本问题开始，循序渐进地介绍相关的背景知识，包括线性分类器、核函数特征空间、推广性理论和优化理论，在此基础上很自然地引出了支持向量机的算法。



《代码的未来》 [日] 松本行弘

作者对云计算、大数据时代下的各种编程语言以及相关技术进行了剖析，并对编程语言的未来发展趋势做出预测，内容涉及Go、VoltDB、node.js、CoffeeScript、Dart、MongoDB、摩尔定律、编程语言、多核、NoSQL等当今备受关注的的话题。



《Head First 设计模式》 [美] 弗里曼

一本设计模式入门书籍，用实际的编程案例讲解算法设计中会遇到的各种问题和需求变更，并以此逐步推导出良好的设计模式解决办法。



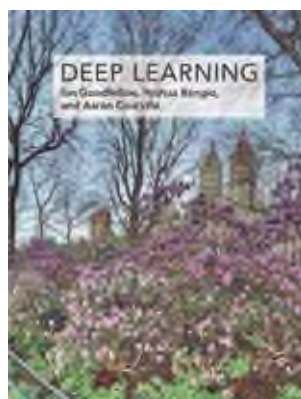
《领域驱动设计》 [美] Eric Evans

全书围绕着设计和开发实践，结合若干真实的项目案例，向读者阐述如何在真实的软件开发中应用领域驱动设计。书中给出了领域驱动设计的系统化方法，并将人们普遍接受的一些最佳实践综合到一起，融入了作者的见解和经验，展现了一些可扩展的设计最佳实践、已验证过的技术以及便于应对复杂领域的软件项目开发的基本原则。



《Effective Java》 Joshua Bloch

Effective Java Programming Language Guide distills the hard-won wisdom of today's best Java programmers into 50 techniques for designing and constructing more robust, high-performance software. Josh Bloch, one of Sun's most widely respected Java developers, focuses on the practical problems virtually every Java developer encounters, offering specific solutions and top-notch code examples.



《Deep Learning》 Ian Goodfellow

The text offers mathematical and conceptual background, covering relevant concepts in linear algebra, probability theory and information theory, numerical computation, and machine learning. It describes deep learning techniques used by practitioners in industry, including deep feedforward networks, regularization, optimization algorithms, convolutional networks, sequence modeling, and practical methodology; and it surveys such applications as natural language processing, speech recognition, computer vision, online recommendation systems, bioinformatics, and videogames.

类别二：商业思维养成



《刷新》 [美] Satya Nadella

任何组织和个人，达到某个临界点时，都需要自我刷新。为了迎接智能时代的挑战，作者提出自我刷新的三个关键步骤：拥抱同理心，培养“无所不学”的求知欲，以及建立成长型思维。



《商业的本质》 [美] Jack Welch

任何组织和个人，达到某个临界点时，都需要自我刷新。为了迎接智能时代的挑战，作者提出自我刷新的三个关键步骤：拥抱同理心，培养“无所不学”的求知欲，以及建立成长型思维。



《规模: 复杂世界的简单法则》 [英] Geoffrey West

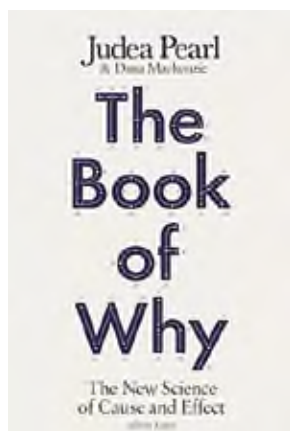
《规模》将帮助你重新思考生命、认识自身、了解你的生活与工作，并告诉你复杂世界其实充满简单的逻辑，只要跳脱思维框架，打破学科限制，你就会重新看清你周遭的一切。



《债务危机》 [美] Ray Dalio

在这本书中，达利欧通过三部分内容阐述了债务危机及其应对原则：债务大周期模型（解释模型）、3个重要案例（深入考察20世纪20年代德国魏玛共和国、20世纪30年代的美国大萧条和2008年金融危机）、48个历史案例和数据对照分析，以备受启发的新方式理解经济与市场。

类别三：文化&科普



《The Book of Why》 [美] Judea Pearl / Dana Ma

Pearl's work enables us to know not just whether one thing causes another: it lets us explore the world that is and the worlds that could have been. It shows us the essence of human thought and key to artificial intelligence. Anyone who wants to understand either needs The Book of Why.



《魔鬼经济学》 [美] 史蒂芬·列维特

《魔鬼经济学》中确立了一个有悖于传统智慧的观点：如果说伦理道德代表了我们心目中理想的社会运行模式的话，那么经济学就是在向我们描述这个社会到底是如何运行的。



《罗马人的故事》15卷本 [日] 盐野七生

还原一个意大利半岛的小国成为世界帝国，持续1300年的辉煌历程，再现一个英雄辈出的时代……读懂了罗马，才能读懂过去的欧洲和今天的世界。



《孙子兵法》 孙武

我国最古老最杰出的一部兵书。



《QED》 [美] 理查德·费曼

一个新的视角来看量子世界。



《枢纽》 施展

中国历史演进的逻辑，中国奇迹持续的原因，以及，该如何认知我们的世界角色。在这个关键节点，每个人都在思考未来。我们在思考未来的时候，最重要的事情是对目标的设定。而如何设定目标，取决于你怎么理解自己、理解中国、理解世界。



《叩响命运的门》 马小平

《叩响命运的门》精选了古今中外最为著名的人文素养经典文本88篇。这些篇章充满生命的智慧和力量，通过对经典的阅读，我们可以和无缘谋面的历史巨人或不相识的朋友进行精神的对话，并在这一过程中实现了文化意义的交流与传承。



《星空的琴弦-天文学史话》 汪洁

这是一部人类探索天文的历史，更是一部人类理性崛起的历史。跟随本书，你将回到过去，在那一个个激动人心的天文大发现的历史现场，和科学家们一起感受当时的兴奋，也体会他们探索的艰辛。

类别四：工具&方法



《金字塔原理》 [美] Barbara Minto

一本讲解写作逻辑与思维逻辑的读物。



《高效能人士的七个习惯》 [美] 史蒂芬·柯维

精选柯维博士“七个习惯”的最核心思想和方法，为忙碌人士带来超价值的自我提升体验。用最少的时间，参透高效能人士的持续成功之路。



《软技能 - 代码之外的生存指南》 [美] John Z. Sonmez

本书聚焦于软件开发人员生活的方方面面，从揭秘面试的流程到精耕细作出一份杀手级简历，从创建大受欢迎的博客到打造你，从提高工作效率到与如何与“拖延症”做斗争，甚至包括如何投资不动产，如何关注自己的健康。本书共分为职业篇、自我营销篇、学习篇、生产力篇、理财篇、健身篇、精神篇等七篇，概括了软件行业从业人员所需的“软技能”。

第二部分

职场成长技巧与解惑

- 职场成长技巧&故事
- 淘系工程师圆桌讨论
- 一线职场&学习问答实录

职场成长技巧&故事

• 从文科生到前端专家 – 在转行时我想过的问题

- 作者：繁易
- 出品：淘系技术

从一篇约稿说起：

在 2021 年的 8 月份，工作中的我接到了淘系技术运营同学的约稿，邀请我谈一下“程序员最重要的能力是什么”。随后我根据自身经历，写下了回答，全文发表在淘系技术公众号的《程序员最重要的能力是什么？》中。

到了九月底，淘系技术运营的同学和我说文章发表出去的反馈不错，希望我能继续写一下具体的心路历程，于是就有了这篇文章。

希望能为和我一样，对编程感兴趣/犹豫转行/正在转行的同学们提供一点经验分享。

我的编程学习经历

我是转行成为程序员的，可以分享一下我的编程学习经历。

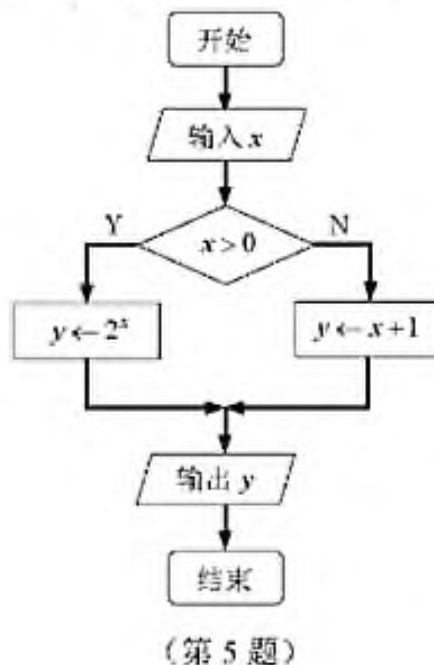
在学习编程之前

在高中时，因为学不会物理和化学，我选择了文科，这也意味着计算机专业类对我关上了大门，不过在当时我也没有想到自己在未来会喜欢上写代码并以此为生。

现在回想起来，与编程有关的只有两件事情。

一是在繁忙的高中生活中，我一直订阅着《电脑报》，并通过它了解互联网的时事。二是当时的文科数学需要解一种流程图题，也是我最喜欢的题型，现在看来与代码流程其实是无异的。

5. 右图是一个算法流程图, 若输出 y 值为 -2 , 则输入 x 的值是_____.



【答案】: -3

从 C 语言开始

升学后, 我就读的是人力资源管理专业。2015 年初, 大一上学期的期末考试刚结束, 距离回家尚有一段时间, 因为闲着无聊便上网络学起了翁恺老师的《C 语言基础》课程, 随后就被带入了到了计算机编程的坑中, 至今尚未脱坑并越陷越深。

首页 > 国家精品 / 计算机



程序设计入门——C语言 国家精品

第13次开课

开课时间: 2021年09月13日 - 2021年11月30日

学时安排: 5-7小时每周

运行至第4周, 共12周

立即参加

记得当时的我，在 2015 年寒假一个月的时间内。开始了每天听课，思考。然后把代码输入手机慢慢运行的过程（那时候还没有笔记本电脑）。当时最开心的就是图片的这一段代码，一次性编译运行通过。



The screenshot shows the C4droid application on a mobile device. The interface includes a top status bar with icons for a play button, headphones, signal strength, and battery level at 66% with the time 13:58. Below the status bar is the C4droid app header with a menu icon, the text 'C4droid', and buttons for '文件' (File), '编译' (Compile), and '运行' (Run). The main area displays a C program with line numbers 1 through 19. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int a;
5     printf("请输入三位数\n");
6     scanf("%d",&a);
7     int b;
8     int c;
9     int d;
10    b=a/100;
11    c=a/10%10;
12    d=a%10;
13    c=d*100+c*10+b;
14    printf("倒序的结果是 %d",c);
15    return 0;
16 }
17
18
19
```

从“年轻工程师”到阿里前端技术专家

往后的半年中，编程的学习也并没有像想象中的那样顺利。学习进度停滞不前。我还清楚的记得当时的自己：“会用20门语言输出Hello World!”，“别人一周就能上手的 Bootstrap 我学了四个月”等诸如此类的事情。

偶然的一个契机，我找到了愿意带我写代码的同学，并磕磕绊绊的完成了第一次 Ajax 操作。那之后的我仿佛开窍了一般，编程世界的大门，终于向我敞开，任我探索和遨游。

直到有一天，我遇到了瓶颈。

2016 年的前端，技术日新月异，当时的我颇有一些手足无措，总觉得时间不够用。于是在参加当年的 JSConf 时，遇见了当时前端圈的名人贺师俊（Hax），于是便壮着胆上去交流了一番。

对话的内容我只记得个大概，在交流结束时我向贺老致谢，他回复我说：“学东西不要有那么强的功利心，而且对于我来说，能帮你们这些年轻的工程师就非常好了。”

在那时候我发现，对编程的兴趣已经托着我从学生过渡到大家眼里“年轻的工程师”了。时间一晃眼已经过去了 5 年，当年的学生，也从“年轻的工程师”蜕变成了阿里的前端技术专家，身份在变、环境在变、技术在变，只有对代码的兴趣没有改变。

从 Node.js 使用者到 Node.js Collaborator

由兴趣驱动的源动力，往往是纯粹且不掺杂利益的，而在软件世界中，兴趣驱动与开源精神是高度契合的。

在 2020 年冬季到来的时候，我遇到了新的瓶颈，想学习更多 Node.js 的知识却不知从何下手。

这一次我的选择是参与 Node.js 开源社区的协作。我开始阅读 Node.js 的源码，解决 Github 上的 Issue 等。在事后提名 Node.js Core Collaborator 时，我统计了一下，两个月的时间内我一共提出了 50+ 的 Pull Request。

现在回头来看，这些与实际工作“无关的兴趣”，最终支撑着我走的更远，现在的我在 Node.js 架构团队，全职负责着 Node.js 的工作，个人的眼界也开拓了许多。

转行的那些问题

转行并非是一帆风顺的。在这个过程中，最常见的问题莫过于如下：

1. 非科班的同学，能学会计算机吗？
2. 计算机专业学了 4 年，你自学的能跟得上吗？
3. 出来后能找到工作吗？
4. 转行的过程中需要做什么呢
5. ……

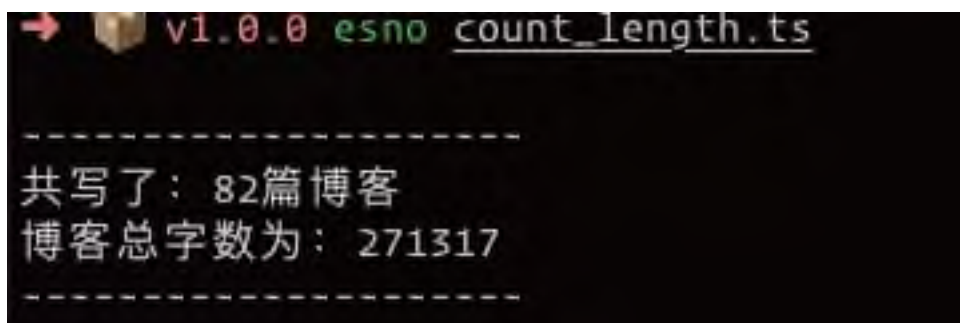
这些问题从我学习编程时，就一直萦绕在耳旁，时至今日依然会有同学找我聊天，然后再问出上述的问题。也是借着此机会，我写一下我的想法。

非科班能学的会计算机 / 自学能跟得上吗？

答案是肯定的：“可以的，没问题的”。

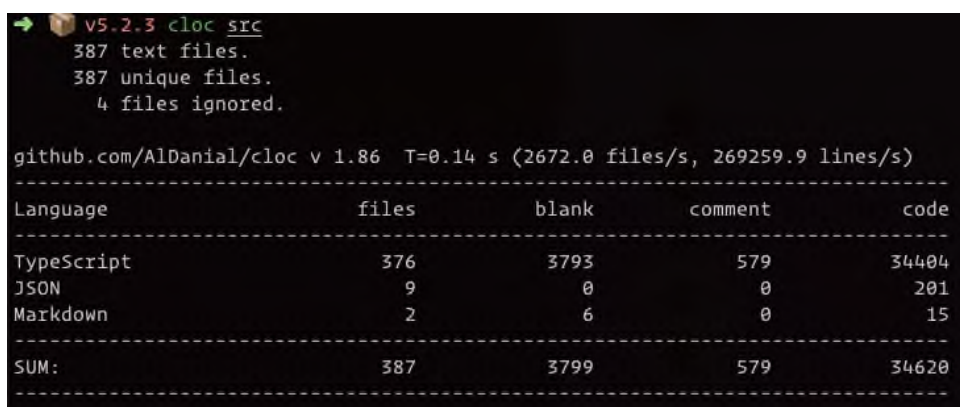
说起来，我从未纠结过自己非科班的出身。在学习编程的伊始，我就想清楚了，如果想要成为一名优秀的程序员，科班 / 非科班的身份根本就无关紧要，真正的关键在于个人的学习效率与投入的时间。

而在大学中，我投入了大量的时间，平均每天编程 6h，这样的生活一直持续到毕业。而在毕业时，我已经完成了大约十几个项目，写过十几万行代码，也有一个十几万字的博客。



```
→ v1.0.0 esno count_length.ts

-----
共写了: 82篇博客
博客总字数为: 271317
-----
```



```
→ v5.2.3 cloc src
387 text files.
387 unique files.
4 files ignored.

github.com/AlDanial/cloc v 1.86 T=0.14 s (2672.0 files/s, 269259.9 lines/s)
-----
Language             files      blank     comment      code
-----
TypeScript             376        3793         579       34404
JSON                   9           0           0         201
Markdown               2           6           0          15
-----
SUM:                   387        3799         579       34620
-----
```



换言之，我也认为，如果一名非科班的同学，学过计算机的课程，写过很多行代码，也可以顺利的完成项目，那么他就是程序员。

出来后能找到工作吗

答案也是肯定的：“可以的，没问题的”。

计算机编程就业环境还是很好的，并不会卡死专业，而是通过过往的经历 + 笔试/面试来决定是否录用你。所以不用太担心这方面的问题。

这个问题我之前在知乎也回答过，所以就不再重写啦，可以看下面的截图。

前端实习真的有这么难找？



子健.Re
node.js collaborator

维尔希宁、ShuSQ、cyan 等 164 人赞同了该回答

-----UPDATE----- 已确认去北京实习。-----

分享下自己的经验，希望能对题主有所帮助。

同大二，某个不知名211的学生，在校内的工作室学习前端近一年。已收到两份前端暑期实习offer。共耗时两周左右。我和题主有很多相似之处，不同的地方可能在于，我的项目经验比较丰富，同时对新技术比较熟悉。

最近的一个项目是一个办公系统，前后端分离，而我把整个办公系统中，我负责的部分做成了SPA，大约二三十个组件，而且从技术选型到实际编写，均独自完成，这点真的很锻炼人。整个项目历经5个月，目前仍然在编写过程中，中途因为设计图变更与ES6的学习，对代码进行了两次彻底重构。所以最终使用的框架和技术是：ES6+Vue+Vue-Router+Vuex+Webpack+ESLINT。同时在做这个项目的过程中，把《JavaScript高级程序设计》和《ES6标准入门》看了好几遍。整个项目，个人超过两百次的commit，以及对各种BUG的解决，Vue，Vuex源代码的阅读和理解（感谢Vue作者 @尤雨溪，创造了如此优雅易用的框架），保证了我学习的高效性。这些经历在我找工作的时候，都有极大的加分。

这一点还得特别感谢 @贺师俊 前辈，他在知乎上某个问题的回答：

“才大二，少做（react/angular/php）hello world级别的事情（除非你能作出点真的有点实际价值的产品），先打基础。”

这个回答指点了我很多。所以从那一天开始，我就决定专研透彻日常使用的Vue框架，实打实的做一个产品。同时继续狠补计算机专业课，争取不给自己拖后腿。（答主是文科生，所以高数，计算机啥的都是从头自学，还好现在有MOOC，帮我实现了学习计算机的梦想。所以很羡慕题主的科班身份.....）

当然，我面试那些大公司，也挂了.....

至于找工作，给Offer的两家公司，都特别Nice，也都知道我大二。

第一家公司，两轮电话面试决定发给我Offer，同时我看公司的官网写的有点凌乱，于是就自己重写了一遍，发给了公司.....现在公司的官网上挂的就是我重写的页面。

转行的同学要如何学习呢

这可以说是社区每天都能遇见的问题了。但授人以鱼不如授人以渔，我说一下我自学新领域所使用的方法。

方法很简单：上知乎，搜索 XX 入门（如计算机入门，前端入门），就会出现一大堆提问，你需要的只是耐心读完其中高赞的一些回答，并归纳总结分析，大概心里就有数了。随后选择其中你觉得靠谱的高赞回答去学习即可。

许多提问的高赞回答其实都回答的很详细也很好，认准教程后跟着学习即可，学习的过程中你又会遇到新的问题，这个时候继续再搜索即可。

这个方法很管用，时至今日，我在学习产品、设计、演讲、游戏等新领域时都是这样做的，在尝试和摸索的过程中，你的自学能力也会得到提升。

我的自学方式

我在刚学习前端时，幸运的在知乎上看到了大漠关于前端自学的回答。

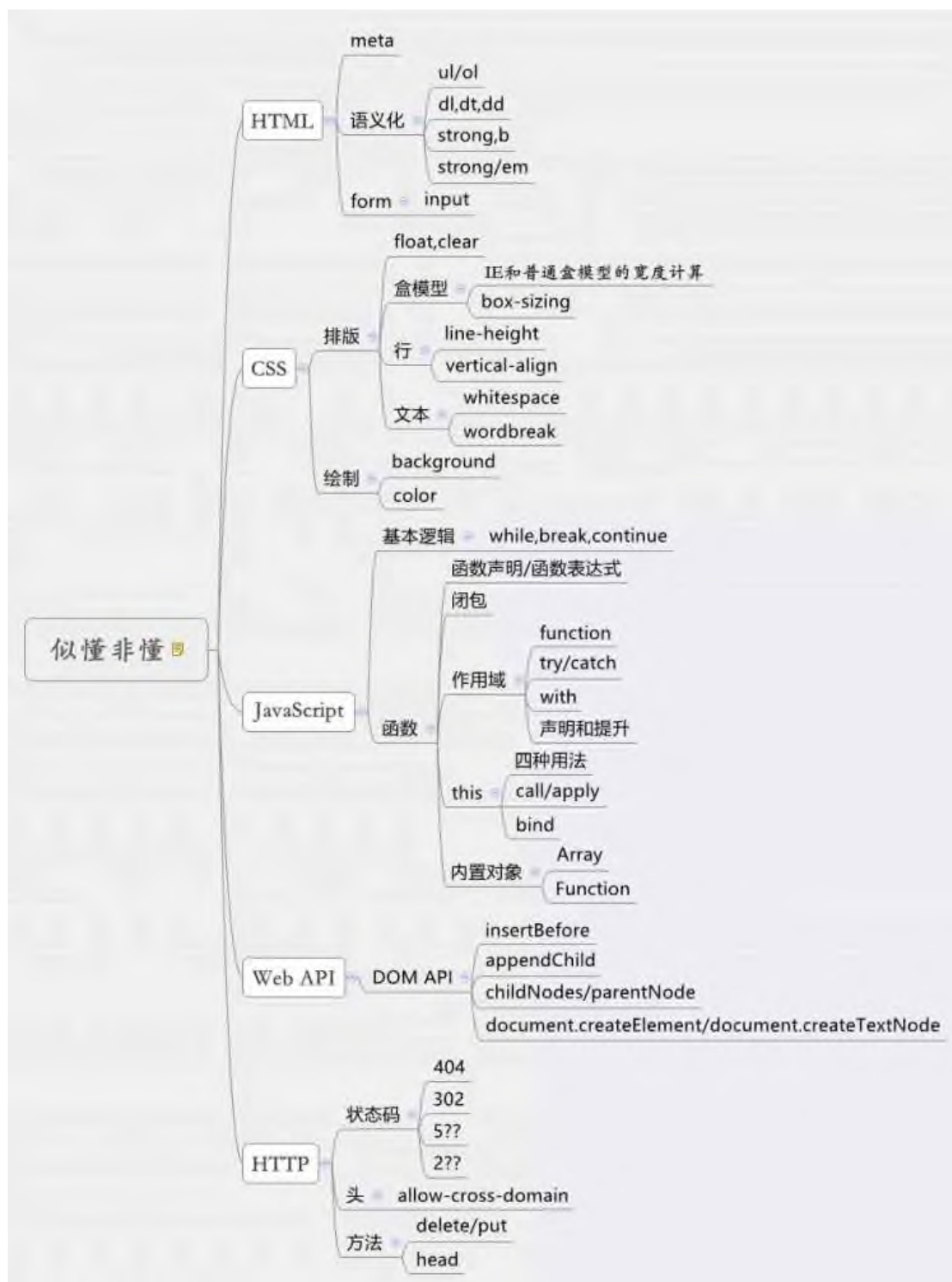
后面我还经历了这样的一过程,我思考过三类问题,并且将他们列在一起:

- 1.哪些知识点懂了?
- 2.哪些知识似懂非懂?
- 3.哪些知识不懂?

接下来有了这样的三份清单之后，就能非常清楚自己知道自己，然后先解决第二个清单中的list，再解决第三个清单中的list。

最后建议，学习这个过程是不断渐进的，整个过程把握：多看、多想、多问和多做。这也是我自己的四多原则。如果你时间允许，多写写东西，总结自己的知识。现多看看规范。

后面也针对自己的情况，画了一张思维导图，把似懂非懂的知识点列了出来。。



后面在寒假里针对这些问题重点攻关，并写成总结的技术文章。

- 02-17 » 寒假前端学习(10)——理解DOM事件流的三个阶段
- 02-09 » 寒假前端学习(9)——理解CSS盒模型与宽高计算
- 02-06 » 寒假前端学习(8)——理解CSS浮动与清除浮动
- 02-05 » 《计算机科学导论》读书笔记（一）：浅析面向过程与面向对象编程
- 02-03 » 计科之路--Start!
- 01-28 » 寒假前端学习(7)——学习JavaScript之this,call,apply
- 01-27 » 迁移至Ubuntu的折腾之旅
- 01-19 » 寒假前端学习(6)——学习JavaScript数据结构与算法（四）：二叉搜索树
- 01-16 » 寒假前端学习(5)——学习JavaScript数据结构与算法（三）：集合
- 01-15 » 寒假前端学习(4)——学习JavaScript数据结构与算法（二）：链表
- 01-14 » 寒假前端学习(3)——学习JavaScript数据结构与算法（一）：栈与队列
- 01-10 » 寒假前端学习(2)——HTML语义化标签探析
- 01-09 » 寒假前端学习(1)——HTML meta标签总结与属性的使用介绍
- 01-09 » 2016，寒假前端学习计划与进度

目前来看，找到自己“似懂非懂”的知识，学习，总结，分享是一个非常有效的学习方式，有兴趣的同学也可以参考试试。

我也一直坚持，学习方案的重点在于学习效果，而不是方案有多复杂和多详尽，适合自己就好。

转行的过程中需要做什么呢

这里我的建议是：

1. 多做项目
2. 多写总结
3. 多打基础

首先我建议多做项目，比起打下“扎实的基础”，对于转行的同学，培养起对编程的兴趣是更重要的。不同于科班良好的学习氛围与节奏，自学往往对自制力的要求更高，这时候培养起对计算机的兴趣，能让自己继续学习写代码才是更重要的。

其次我建议多写总结，总结是一个非常高效的学习过程，如果你要把一件事情给讲清楚，那么首先一定要弄懂这个知识。自学计算机的过程中，多总结不仅可以查漏补缺，也可以让自己在回顾的时候多一份积累的感觉。

最终则是建议多打基础，转行计算机的同学，计算机底层原理，算法等知识接触较少，但这些却实打实的是基础，可以帮助你未来少踩很多坑的知识，因此在学习之余也可以多关注计算机基础知识（至于怎么学习，可以参考上面的回答）。

回头看转行

时光荏苒，回头看已经写了快 7 年代码了。

回头再看自己的转行经历，与其说是文科生转行学了计算机，倒不如说是自己发现了对计算机的兴趣，并持续投入至今。

最后，送上我很喜欢的一段话作为结尾。

「 Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do. If you haven' t found it yet, keep looking. Don' t settle. As with all matters of the heart, you' ll know when you find it. 」 — Steve Jobs

● 淘系工程师亲述经历：如何避免成为“技术工具人”？

- 作者：玖五
- 出品：淘系技术

时间好快，眨眼间，加入阿里已经一年了。这一年发生了很多事，整体上非常地充实且精彩，在一件又一件事情中，我不停地犯错，一路走来，步履蹒跚，也收获到了很多成长。每次结束一件事后，经过短暂宁静的生活便再次踏上新的征程。

之前写过“畅销书《深入浅出Vue.js》作者，在阿里淘系6个月的收获成长”，因此本文主要讲述“后半年”收获的成长。

关于“思考”

- Mentor：你平时周末都做些什么？
- 我：没事的时候通常会看看书，写写代码，研究一些自己感兴趣的东西
- Mentor：可以把一些平时做事的时间换成什么都不做，坐在那“思考”，想一些东西

以上对话来自一次我向导师的询问：“我应该如何再进一步，7和8的区别是什么”（大致是这个问题，原问题记不清了）。

8相比于7来说，在工作层面上“可以调动更多资源”。

之所以需要调动更多资源是因为需要做更复杂的事。

复杂的事哪来？“思考得来”，当然也可以靠组织分配，但谁又能保证那个机遇可以恰到好处地分配给自己。

完全等待机遇的分配，这也违背我内心一直以来所坚定的信念：“让事情因为自己而与众不同”。

完全靠主管分配机遇，这也违背我内心一直以来所坚定的信念：“让事情因为自己而与众不同”。

今天能通过面试进入到阿里巴巴的同学，能力都不差，主管把机遇分配给另一个人绝大概率拿到的结果不会比我差多少，一件事情究竟是因为我去做才拿到好结果，还是大家去做都能拿到好结果，不太好说，这达不到“让事情因为自己而与众不同”的标准。



只有自己凭空创造出的机遇并最终拿到了结果，才符合“让事情因为自己而与众不同”，也更能展现出自己的水平。



所以一个更可靠的发展路径是：

- 1.多思考项目未来的发展方向和现有技术体系的问题
- 2.做判断并按照正确的方向执行
- 3.事情足够大就会衍生出调动更多资源的需求
- 4.时机成熟后，可能会晋升到下一个级别以方便调动更多资源

这会衍生出一个新问题，“如果自己思考的项目未来发展方向和大团队方向不一致怎么办？”



其实不用担心，可以和主管多沟通对焦，如果自己思考得来的方向客观事实上是正确的方向（各方面受益都更高）或者是绝大多数人都信任是一个正确的方向，那么正确的方向一定会替换掉错误的方向。

技术与业务两条腿走路

一定要“技术与业务两条腿走路”。

这是我主管和我导师对我的忠告，当两个经验丰富的大佬不约而同地给了相同的建议，足以证明这句话的分量，这也引起我深思。



自从我开始工作，我都是只搞技术，我对业务其实不太感兴趣，我很崇拜那些知名的技术很强的世界顶级工程师，一直以来我的梦想都是想成为他们中的一员，想成为前端行业技术吊炸天的世界顶级工程师。

但现在我渐渐意识到，不能只搞技术，也要多思考思考自己的业务，这对自己是有好处的，对业务多思考的好处是：

- 1.提前对技术体系做布局，引领技术与项目，避免业务突然变化时陷入被动
- 2.对现有支撑业务“较为成熟”的“有瓶颈”的技术体系做出改革与突破
- 3.避免让自己成为“工具人”

让“技术”和“业务”互相成就彼此，共同成长。业务发展倒逼技术改进，技术改进成就业务，相佐相成。



身为技术同学，可以基于对业务的预判用技术落地项目辅佐业务，业务的成功再反过来成就技术。

千万不要陷入到一个巨大的误区：技术的自嗨，其实并没有多大贡献。

所有伟大的技术创新，都是那些对社会有巨大贡献的技术，伟大技术的诞生，都是基于一个不被满足的“需求”，基本上伟大的技术都是这样被创造出来的（Git、React、Vue又或是支撑公司业务背后的技术体系）。



只有对业务足够了解，思考的足够深，才能知道用户需要什么，才有可能引领未来支撑业务的技术体系，才有可能创造出改变世界的伟大的技术。

一门心思只搞技术，很难做到“引领”和“创新”。大概率只能做到学习现有已被其他人创造出来的技术，学到精通，成为某领域专家，但很难引领某个技术领域。

稳定发挥

发挥稳定型（线性增长）选手比阶段性发挥波峰波谷选手更有优势，稳定（可预测性）本身就是一种优点。

发挥不稳定的选手，缺点是不可预测，换位思考如果自己是主管，有一件很重要的事，你敢交给发挥不稳定的选手么，万一碰上发挥较弱的时间段，就悲剧了。

这个道理，在电子竞技、体育竞技等领域都相通，发挥不稳定是不可能拿到冠军的。



刚进入一家公司，在一个新环境都会有点着急，急于拿到成绩得到认可，这本身其实是件好事，但不要把劲使大了，把自己变成了那个波峰波谷型选手。

放轻松一点，少使点劲，让自己线性成长稳定发挥。

毕竟，职业生涯本就是一场“没有终点的长跑”，大家比拼的并不是短期内谁跑的更快，而是“坚持”，在这条赛道上能跑赢的，不是那些跑得快的人，而是为数不多坚持跑的人，他们能跑赢，只是因为还在跑。

求同尊异

某一刻，我终于理解了这四个字，这要从一件事说起。

最开始，为了帮助团队成员“提升个人写作”、“提升表达能力”、“提升个人技术成长”。

我提出了文章计划，团队成员每个人大概每5个月写一篇文章，同时发明了“贝利体系”作为奖惩机制。

每人每月按照一定的数量自动掉贝利，贝利掉多了需要接受惩罚，发表了文章后奖励贝利，只要保证每5个月写一篇文章贝利就不会达到惩罚线。

具体数值都是我计算好的，并写了段程序自动执行，拿到手里的贝利可以用来兑换一些礼物，有HHKB、AirPods等可以兑换。

后来“文章计划”受到了大家的集体挑战，觉得给大家造成了非常大的压力和负担，“文章计划”就宣告结束。

不过“贝利体系”被我保留了下来，虽然不强制大家写文章了，但依然鼓励自愿写文章的同学，并给予贝利奖励。

这时我对贝利体系进行了一些思考，并重新定位：“衡量体系”，衡量团队成员对“团队建设”、“组织文化”、“横向贡献”的贡献值，助力团队和文化的横向建设。

简单来说，就是所有对团队横向有付出的同学，我都会按照贡献的大小付出的多少来奖励贝利，且有一个排名，排名高代表横向贡献多，我会给予贡献多的一些同学发一些礼品，贝利本身也可以自行兑换礼物：HHKB、AirPods等。

我希望贝利体系和团队横向的事情，例如：团建、招聘、安全生产、写文章、技术分享、对现有产品提改进建议、组织大家健身、组织大家玩桌游等有更深度的结合。

因此，有一天，我提出一个想法：让贝利少的人举办团队的团建，并给予举办团建的人奖励贝利，主要考虑给团队横向贡献少的同学多些机会做些贡献。

且在团建过程中结合贝利有一些有趣的玩法，例如可以按照贝利的多少设定初始装备（贝利高，团建玩游戏时略有优势，但又不失平衡）。

但这个提议被团队负责团建的同事拒绝（因为他认为贝利不客观公正，无法客观衡量谁贡献少）。

当时我觉得这是一个对团队有帮助的好事，可能它暂时不完美，但我会持续优化，我是在“坚持做正确的事”。

而且当场我问同事觉得哪里有缺陷需要改进也说不上来，再加上我觉得自己在做正确的事，在让这个团队变得更好，所以我就和同事大吵了一架，对，我又双叒叕和人吵架了，而且这次格外激烈。



我对自己进行了深度的反思，“贝利体系”打被我凭空创造出来之后，无论是面向用户，还是面向合作方，都不是很受欢迎。

面向客户，团队成员觉得这是一种压力和负担。

面向合作方，团队横向负责人没有与贝利体系合作的动力和需求。

这件事本身不是大事，但做这件事却非常难，贝利诞生到现在一直被大家抵触，被大家无视，还有人觉得这是几个人之间的小众游戏。

但我不愿放弃，我想让我们团队因为我的存在变得不一样，而我又坚信这是正确的事，是一件好事。

为此我和主管聊了两三次，学到了一些知识和做事的方法，总结提炼出精华：

- 1.不要以自己为中心去思考问题，要换位到“团队视角”，“合作视角”全方位立体多维度思考问题
- 2.推进事情要考虑：“共赢”，“利他”



贝利体系诞生以来，所有的“规则”（包括哪些贡献应该奖励，奖励多少）都是我一个人定，大家内心是“不认可”的，因此外在表现就是“你自己玩你自己的，我不参与”。换位思考，每个人都会抵触自己“不认可”的事情。

这一刻我终于体会到，也理解了什么叫“求同尊异”，每个人都不一样，也不是所有人都和自己想法一样，要尊重不同的建议和声音。一件事，只有大家认可了才能赢得尊重和成功，要赢得客户的认可，赢得合作伙伴的认可。

后续：

这件事之后，现有的“规则”我都通过匿名调查问卷的方式投票决定，调查大家认为哪些应该奖励，应该奖励多少，哪些不应该奖励，并根据调查问卷的结果进行了修改。

所有的规则，完全由匿名投票大家共同决定，规则制度“公开透明”，由全体成员“共同参与”。

并且提供了日常的“实名”和“匿名”双通道接收意见反馈，并给反馈意见的同学奖励贝利。

获取贝利和消耗贝利的方式也变得更加的多样化。

且这些新的多元化的获取和消耗贝利的方式都是由团队成员大家共同贡献出来的。

经过一系列的调整，整体认可度相比之前有了很大的改善，贝利体系在向着更好的方向迈进。

年前也按照大家的贝利数量给大家发放了同等价值的礼品，并启动新一轮周期，大家都很开心。

这件事虽然不大，但是它教会了我“如何推进事情”，未来我大概率会打破现有已经“成熟”甚至“固化”的技术体系，为现有技术体系做一些改进，让它变的更好。

那么推进并赢得大家的“认可”和“尊重”与这件事是相同的，通过这件事，我提前得到了锻炼，这是无价的宝藏。

身为PM如何做事

感谢主管的培养和信任，不止给我很多试错空间，还在我犯错后教我如何做才是正确的做法。

“风险”和“进展”及时同步

关于风险同步在《我在阿里半年收获的成长》有提到过，最近又有了新的感悟：不要担心“做的不好”或“不完美”而不敢同步进展和风险，因为“差的信息”比“没有信息”要好很多！

及时同步“风险”和“进展”的好处是：如果真的做错了，会得到及时的纠正和帮助，可以保证项目是安全的，

项目安全永远是第一位。

不要担心大家会觉得自己菜，自己菜不菜根本没人关心，大家关心是：

- 1.项目能不能“按期”、“高质”交付
- 2.你是否在成长

哪怕中途做一步错一步，也比中途“毫无音讯”强无数倍。



即便是中途做一步错一步，但由于及时同步了风险和进度，在不停地犯错中一点点把项目做好，最终大家也会看到自己的成长。

会对自己很放心，下次类似的事情交给自己会让人安心，因为再差再差，自己也不会把事情搞砸。

线上问题如何应对

线上出了事故后，立刻向上汇报，不要自己先闷着头去修复！避免业务方找过来时主管完全不知情，这种情况整体都会很被动！

反馈问题的方式：

- 1.站在用户视角描述发生的问题
- 2.影响面预估（面向用户的影响面，不是判断技术哪里报错，判断不清楚默认当做重大影响处理）
- 3.处理策略
- 4.如果有原因，提供原因

不要把自己当做唯一的资源

当接到一个任务后，首先考虑的是“怎样把这件事做的更好”，“谁来做更合适”，不要把自己当做唯一的资源。

合适的事让合适的人来负责，接到任务后第一个想的是如何把事做好，谁来做更合适，如果自己擅长某一块可以自己去做，如果某一块有更合适的人选，那就应该找到合适的人来做，而不是自己去做。

“悲观”态度给答复

如果评估不准某个功能是否可以按期上线，一律按“悲观”态度给反馈（本质其实是：提升专业性，预判风险，做好预期管理）。

我见过的，除了我还有新手PM也犯了这个错，那就是一句：“我试试”（觉得大概率来不及，但还是和产品说感觉来不及，但我努力试试）。最终没有按期上线时产品就会找过来问为什么没有上线，“不认可”这个结果。

所以，如果评估不准，或感觉有风险，一律给悲观答复。如果一开始有来不及的可能，在一开始就给来不及的“明确反馈”。

技术判断

技术PM最重要的核心竞争力和职责叫做：技术判断。

像双11这种级别的大促，每个功能所涉及的上下游链路都会非常复杂，横跨N多个团队，这就意味着，同一件事，可以有N种解决方案，而不同团队看待问题的视角不同，因此大家给出的方案和倾向性很多时候会有冲突，这时候技术PM要做的就是给一个技术判断，方案1、2、3、优缺点是什么，让高年级同学拍板。

而不是把一个问题抛上去让高年级同学们想方案。

因为信息是越底层知道的越多，越上层对细节的信息越少。

总结

不知不觉，来阿里已经一年了，这一年是我近几年来成长最快的一年，自己的思维和想法，都有了质的提升。

非常感谢舒文把我带到这个团队，以我的学历正常很难进到这个团队，经常感叹自己真的是凭运气遇到贵人。

还非常感谢墨冥（我的主管），这一年来不断地言传身教并给予机会试错，这一年来的成长绝大部分来自墨冥的教导，再次感叹自己的好运气。

相信未来，我会在实战中承担更大的职责，相信未来，我会让我们团队因为我的存在变得不一样。

最后，在舒文身上学到了一个原则，特别认同：

学会坚持，“长时间的积累”永远比“为了短期高收益频繁切换”收益高，无论是“日常做事”、“投资”还是“职业规划”、“人生规划”等。

舒文经常说：“做成一件事情”有很多因素，“坚持”是成本最低的一种。

• 仅9天拿下CVPR竞赛冠军，家里的狗狗都改了作息

- 作者：淘系技术
- 出品：淘系技术

CVPR 2021 Homage Genome Action 比赛，让我们聊聊淘系冠军背后的故事，以及“视频交互关系检测”这个CV领域的未来新趋势。

“薛定谔”的冠军？

“第一次提交榜单的时候，我们队排名第4，和第一名数据还差了十个点。”

“当时只剩下5天就截止比赛了，要是没拿到成绩，我决定装作这一周什么都没发生过。”

正是眼前这个带着点羞赧笑容的男人，一周后带领着他的3人队伍，以远超出第二名8个点的成绩，拿下了CVPR 2021 Homage Genome Action 首届比赛的冠军。



接受我们采访的冠军张爱喜（花名：少麟）

报名就是一场冲刺

他是张爱喜（花名：少麟），来自阿里巴巴淘系技术多媒体算法团队，这支团队支持着淘宝直播、逛逛、点淘等内容业务，在高层语义理解的技术领域一直有着持续的探索。

这次斯坦福大学李飞飞教授课题组重点推动的 Homage Genome Action 比赛，也是高层语义理解的方向，着眼于探索视频结构化的信息与交互，在传统的画面识别基础上，更多地探索视频中人与物的关系推理，从而实现更有效的算法推荐和分发。

作为 CVPR 去年的冠军选手少麟，今年“玩”得更刺激了点：报名时距离比赛截止还剩9天。（正常情况下，一般选手的备赛时间是1-3个月）

这是不是很像学霸考试前说自己没复习的凡尔赛？

“就试试咯，我们参加比赛，一方面是因为这个方向我们在尝试推动业务落地，另一方面，我们也希望通过打比赛来检验一下技术的先进性和通用性。”

彼时，少麟刚刚冲刺了两个多月递交了一篇相关领域的 paper，趁着这股热乎劲儿，搞搞数据集，改改方案，他感觉还能冲一把。

三个和尚有水喝

相比于论文大多数专注于某一个技术点，比赛考验的是多种算法的综合能力和整体方法的创新能力。Homage Genome Action 今年是第一届举办，这意味着没有往年的方案和数据参考，对于创新思考的要求更高。

三个和尚可能没水喝，也可能齐头并进喝。

要在有限的时间内完成冲刺，最关键的是要看比赛的这支队伍如何高效协同，实现精力资源的最优分配。

少麟的队友主要有2个，一个是 paper 的合作者，目前正在北航读计算机视觉方向博士的廖越；另一个是他的老板，大学时拿过ACM金牌，有着一把技术热情的咏亮。

这次比赛提供的是一个室内监控数据集，任务要求准确判断每一组人体和物体之间的动作关系和空间关系。他们梳理了数据集特点和比赛流程后，认为检测和关系判断在比赛中最关键，并迅速做出了明确分工。廖越负责检测模型，咏亮负责模型融合，少麟则负责关系判断模型和全流程整合，三人之间的结果互相依赖，每天上午和下午保持实时跟进，并迅速迭代着各自的方案细节。

“这就像阿里绩效管理OKR的思路，我们确定了O以后，把每个人的 KR 列的清清楚楚。廖越呢，各种想法比较多，我是工程实现能力比较强，咏亮很执着，我们在协商好的 KR 下推进得非常有条理。”

本科北航，硕士香港科大毕业的少麟，一直都是个喜欢突破自己、突破边界的“不折腾不舒服”青年。他原来学的是集成电路设计，在外企捣鼓了一阵儿半导体软件开发后，悄悄瞄上了深度学习。通过自学，16年他开始做计算机视觉方向，从人物静态的脸部识别，到如今动态高层的语义理解，他始终活跃在开疆拓土的最前沿，享受着自己作为一颗嫩芽冲破泥土层的爽感。

距离比赛还剩5天截止时间的时候，他们提交了自己的第一版结果，榜单出来后一看排名妥妥前三开外，和第一名差了近乎十个点。主管咏亮试图宽慰他，“你也拿过冠军，投了 paper，从个人绩效角度而言这个比赛也无增益。”

这段时间，3个人都是近乎没日没夜地投入。一直有着早睡早起习惯的少麟，连着一周多都是5点起床，遛完家里的两只狗狗后，7点就到了公司，一边跑模型，一边打开钉钉回复凌晨来自咏亮的一串问题反馈。



5点被遛完的狗：补个觉，谢谢

“我是队伍的主心骨，比赛也是我提议打的，我不能说不行。”深知队友的付出和努力，少麟表示没有问题，只要在千丝万缕中剥离出方案的改进点，他们依然有希望获得质的飞跃。

一把梭哈夺第一

6月7日，下午3点。坐在工位上的少麟，突然蹦了起来，一溜烟跑去跟咏亮大声汇报：

“我去，我比上次高了接近20个点！”

其他不明真相的同事诧异地看着眼前这个手舞足蹈的男人。原来，由于参赛时间太紧，少麟一直悄悄咪咪在参加比赛，打算要是没希望就做个鸵鸟，把头埋进去当做什么都没发生过。

就在前两天，他们终于找到了方案中的问题所在，他们拆分了验证集，解了一个bug，也完成了最新方案的coding，重新梳理了整个流程结构，完成一系列操作后，单个demo的提交结果显示已提升20个点。

回想起来，那是个比收到最后冠军结果邮件还要高兴的时刻。之后的策略调整就更像打通了任督二脉，每尝试一个策略，数据都会提一点点，喜悦越来越多地飞上少麟的眉梢。

最后一次，实际上也是第二次提交完比赛数据后，他们抱着一把梭哈的心态，少麟和咏亮开心地去公司附近撸了个串，并已然当做过去式在讨论以后的技术规划了。



Home Action Genome Challenge 第一名获奖证书

“视频交互关系检测”这个方向，少麟打从去年10月就开始自发地在研究和探索了。它和单纯的视频理解识别不一样，要求提取画面中结构化的信息，并能够训练机器自主识别人与物实体的交互关系。这个方向目前还处于比较早期的阶段，尚未出现大规模的应用场景。但是该方向已连续在 ECCV，CVPR 和 ICCV 计算机视觉领域三大顶会上举办了workshop，未来在安防和监控领域能够帮助人高效提取需要的信息，极有可能是CV领域接下来一段时间的热点方向。

从淘系业务的角度，少麟希望这个技术的使用，能够帮助算法在商品的推荐和投放上效率更高，计算机能够通过计算和关系判断，智能地将正在直播的淘宝主播或者需要的商品，适时推荐给需要的用户。

“可能用户的直接感知并不强烈，但这个算法技术能够为淘宝直播大大提效。”

技术人的追求：既要又要还要

既是队友又是小主管的咏亮，在打比赛这件事情上态度非常明确：鼓励同学们尽可能参加。

阿里并不是一家可以“躺平”的公司，技术人始终面对着行业环境和业务需求的不断迭代，但任何时候都不能缺失对于技术的敏感度和钻研的追求。对于一名有技术追求的人来说，应该去挑战那些崇山峻岭。

咏亮的团队目前是淘宝多媒体算法分支里竞赛成绩最好的，以少麟为代表，这支来自爱奇艺、小红书、京东、百度等不同文化的小分队，逐渐克服自己的路径依赖和信心不足，慢慢发现，同时做到业务需求开发和技术钻研挑战，并没有想象中那么困难。

蝉联了两届冠军的少麟，特别“凡尔赛”地表示，明年再打比赛就是自己的舒适区了。比赛是方案和技术的综合，论文是更偏单点的创新。下一步地突破，他希望今年能中CCF-A类的一作文章，明年能中 PAMI 和顶会的oral，将自己在业务场景上的思考提炼成基础的学术问题，能够更加形成技术壁垒。

“始终做团队的突破点是很爽的，来阿里之后，发现更难了，因为大家都优秀。如果以后能持续产出顶会paper，成为行业某个领域的标杆的解决方案，就是那种面试的时候，大家会提到说‘哎那个谁写的论文里’的，这种美好愿景想一想就很激动。”

他喜欢《三体》，更因为罗辑这个人物读了三四遍。罗辑这个承载着人类命运重压的普通人，在参透黑暗森林法则的真理之路上，不断突破着认知与精神的极限。少麟认为，不停地探索未知区域应该是种持续的追求，大到人类，小到个人都是如此。

技术人的耐心：盯大熊猫半个多小时

张弛有度，打完比赛，少麟休了一周假，从成都-乐山-西安一路吃喝睡了下去。（家里的两只狗狗表示：谢谢你，我终于不用5点起来被遛了。）

作为一个东北黑龙江人，四川的冷锅串串钵钵鸡、麻辣火锅、跷脚牛肉、毛血旺散发着椒麻的鲜香，瞬间从味蕾治愈到了大脑。

他最喜欢的，还是熊猫基地。

一只硕大的黑白胖子把自己挂在树上，几乎承受不住重量的树枝已经被压成了月牙弯。少麟掏出手机，心里暗暗想着，我可一定要拍到它掉下来的那个瞬间。结果这一站，就是半个多小时。

“耐心这事儿，搞技术的人一定要有的。不然怎么憋得住探索未知的苦闷呢。”



淘系工程师圆桌讨论

• 小项目需要前后端分离吗？

- 作者：淘系技术
- 出品：淘系技术

有网友提问：听说前后端分离是为了让前端人员专注前端，后端人员专注后端，但如果项目比较简单，并且总共也就几个人在开发，前后端分离还有意义吗？

今天我们邀请了 4 名淘系技术的前端以及后端工程师，结合他们自身在项目操作中的感受，给大家分享一些他们对于小项目前后端实际体验的总结，希望能够对你有帮助。

01

淘系技术部 逆葵

小项目是否需要前后端分离，取决于哪种方式能够『时间最快、成本最小地完成当前的业务需求』。

脱离实际场景谈这个问题是没有意义的，毕竟技术架构的选择需要服务于业务。

小项目有多小？它的生命周期有多久？是一个只会存在极短时间的临时系统，还是在可预见的未来有扩展成大型项目的可能？维护这个小项目的技术人员有几个？他们分别是什么技术工种？这些都是需要考虑在内的点。小项目是否需要前后端分离，取决于哪种方式能够『时间最快、成本最小地完成当前的业务需求』。

我们都知道前后端分离是为了让技术人员各司其职，提高效率，在成熟的互联网公司也都是这种运作方式。在人手充裕且技术人员职责清晰的情况下，无论是考虑开发效率、系统可维护度还是系统可扩展性，前后端分离显然都是最合适的选择。毕竟互联网技术发展到今天，前后端已经不是早期那样可以一个人一把梭搞定的了，两种技术已经产生了明显的鸿沟，前端更关注交互和体验，后端更关注数据和并发。

当然，回到背景设定上来，怎么选择依然要取决于你的团队开发人员的实际情况。比如说，如果开发人员只有一个 PHP 工程师，而他刚好对现代的前端框架不太熟悉，平时都是开发 PHP 页面，HTML/PHP/JS 一通杂糅，这个时候强行要求前后端分离显然并不合理。

不过从上面的举例描述来看也能发现，前后端不分离的情况已经是比较落后的生产方式了。今天，以笔者的经历来看，哪怕是学校里的工作室创业，也都是尽量通过多个技术工种协作完成项目。即使一个人身兼数职开发，代码组织方式也基本都是前后端分离的。虽说过度设计是架构中比较忌讳的地方，但是在前后端分离这一点上，已经有充足的实践证明它的优越性。

笔者作为前端，接触到的 Node.js Web 框架都是以前后端分离作为设计原则之一，甚至前后端不分离的经典语言——PHP 中最流行的 Web 框架之一 Laravel 也都是推荐使用现代前端框架 Vue.js 来完成页面开发的。究其原因，现代前端技术的复杂度，从开发阶段的工程架构到提升页面性能的种种优化措施，都不太适合再和后端耦合在一起。

所以，我们的结论依然不变：是否需要前后端分离，取决于哪种方式能够『时间最快、成本最小地完成当前的业务需求』。当然，考虑实际情况，前后端分离是更加普适、先进且合理的方式。

多说一句，时下比较流行的云端一体的应用开发方式，看似是和前后端分离背道而驰，实际上核心原则是一致的。借助 Serverless，前端工程师能够在专注前端 UI 的基础上，更深入地拿到数据的处理权限，某种程度上来说是更为高效的开发方式。至于更偏后端的开发、扩容、服务器运维等，则交给 Serverless 容器来解决。这也是一种前后端分离的实践。笔者认为，在小项目中，可以大胆尝试云端一体开发。

02

淘系技术部 干天

可以从用户体验、开发效率和运维效率三方面来看待前后端是否分离。

我认为在绝大部分场景下，前后端分离优于不分离。

接下来我会从三个主要的场景来分析前后端分离与否：用户体验、开发效率和运维效率。

用户体验

如果你正在做一个门店点餐系统，那么用户对该系统最基本的要求就是界面美观、加载速度快。对于这种侧重于前端效果的项目，如果使用传统的前后端一体项目，界面交互方面，除非有专业的前端程序员，一般的服务端程序员是无法通过 JSP 来实现如果复杂的前端交互逻辑，并且在其中加上丰富的主题和动效的，极大的可能性就是 UI 画的是保时捷，开发出来是夏利。大家可以回忆下 10 年前的网站风格就可以想象了。加载速度方面，因为资源全部放在服务端，首次加载没有浏览器缓存，需要加载大量的 js、css 以及静态资源，这在今天几乎是不可接受的，即使将资源放在 Nginx 服务器，也没办法做到极致加速。如果使用的前后端分离的开发模式，前端由前端项目负责，界面交互方面，可以直接使用现有的前端脚手架来实现，业内已经有专门设计师设计的 UI 框架供选

均实现了丰富的动效和复杂交互，一键引入即可。加载速度方面，前端资源单独发布部署，可以通过cdn预热和极致加速。

如果你正在做一个仅用户服务端接口调用的工具类项目，那用户对该系统的基本要求就是能看懂、能操作成功即可，对于这种侧重于服务端结果的项目，如果功能变化概率较低，完全可以使用前后端一体开发模式，光速开发上线！一套系统解决所有问题。

如果项目是侧重前端交互的，那么最好使用前后端分离，如果项目侧重服务端功能，无复杂交互，前后端一体速度更快。它的服务端就是下单支付这种基本逻辑，很明显，基于传统前后端一体的JSP模式来实现需要有复杂交互的前端页面是很难的，难度一在于页面开发，首先开发html然后由服务端转化为jsp，实现功能的复杂度已经很高了，美观程度更无法保证，这对前端有很高的技术要求。难度二在于流畅性，前后端一体项目加载页面需要经过Nginx打到服务端，如果首次访问没有流量器缓存可能需要多次下载静态资源，页面加载非常的慢，并且服务端没办法解决这个问题。如果选择前后端分离，前端页面可以直接使用现有的前端脚手架来实现，各种优秀的前端框架已经支持了绝大部分的UI组件，集成了动效、样式和交互，绑定服务端数据集即可。加载速度方面前端的资源可以通过cdn加速，实现急速加载。当然如果你开发的是内部使用的工具类项目，前端逻辑简单并且变化很少，使用前后端一体效率可能更高。

对于侧重前端效果的项目，建议前后端分离。

开发效率

前后端一体的项目，仅在前端界面交互简单且变更频次低的服务型项目上占有微小优势，因为此类项目核心是服务，前端仅仅是服务调用入口，只是比postman调用接口稍微产品化一点点的界面而已，不需要投入前端人力，服务端可以顺手开发，这种场景下使用jsp成本最低，效率最高。

而对于任意一个对前端界面以及交互又要求，并且会长期迭代的项目，不管项目大小，使用前后端分离的效率都要比前后端一体要高，前后端分离可以通过定义服务接口并行开发，互不影响，实现双倍提效。专业的人做专业的事，并且专业的领域有更多专业的工具，可以达到事半功倍的效果。

前后端一体的适用场景非常有限，任何一个对前端交互有要求的项目，都应该采用前后端分离的方式去开发。

运维效率

前后端一体的项目，前后端完全耦合，一次部署可实现前后端同时上线，但是一旦服务端宕机，前端页面无法展示，影响全部用户，而且不管是服务端还是前端的迭代都需要重新发布，无法保证前后端互不影响，增大了迭代风险。

对于前后端分离项目，前后端由不同的应用承载，迭代和发布都可以独立进行，实现完全解耦，几乎没有依赖，服务端宕机可以由前端兜底页面交互降低影响面，单独运维极大降低迭代风险，在前端发展到如今，运维工具非常完善，成本很低，相对于诸多风险，单独运维带来的成本增加可以忽略不计。前后端分离在绝大部分场景下的运维效率高于前后端一体。

综上所述，前后端分离在当前的技术环境下已经是大势所趋，其在用户体验、开发效率以及运维效率上均领先于前后端一体的方案，在互联网技术深度发展的今天，行业对技术人员的要求从技术广度转变为技术深度，专业人做专业事，所以，即使小项目也依然推荐前后端分离！

03

淘系技术部 三半

其实真正实践下来，分离也很快的，尝试一下就知道了。

先说结论：

基于现有的前后端技术和运维能力，即使是小项目，也建议前后端分离；当然如果说10年前技术还不成熟，前后端不分离也是项目快速上线的一个比较好的手段。

先谈下何为小项目

- 1.开发周期短的项目（两天内要上线）？
- 2.维护时间短的项目（这个项目用这次紧急用一下，下次就不用了）？
- 3.功能简单的项目（就一个简单的数据库查询，未来不会有需求变更）？

核心问题就是小项目的成本对比？

我的观点：无论项目有多大或者多小，要实现的产品功能肯定是不变少的，基本的开发工作量差异并不大，那么就是前后端分离的运维成本对比了，这个如果有运维现身说法更好，按照我的工作经验来看，运维成本也没太大的差别。

接下来就要谈谈分离的优点了

- 1.必须要说到的就是耦合问题；前后端分离天然地把双方的交互/开发边界厘清了，面向json的数据对接
- 2.大大节省了联调成本，前后端可以并行开发，问题定位更加快速清晰
- 3.可维护性更高，也更加方便后期的重构
- 4.可读性更强等等

这个问题其实挺简单的，会有人觉得小项目怎么简单怎么来，而且先入为主的觉得不分离会更快，其实真正实践下来，分离也很快的，尝试一下就知道了。

04

淘系技术部 饕餮

持续交付且周期越长的项目，前后端分离效果越到后期优势越明显

这个问题往往更多的是看这个项目是否有需要持续交付目的和开发周期长短来确定的。持续交付且周期越长的项目，前后端分离效果越到后期优势越明显。

通常前端和后端所关注和解决的问题是不同的，前端关注的是交互和体验，轻业务关系，需要有随时的调整的交付能力；而后端关注的是数据逻辑，重业务逻辑，相对来说稳定确定一切。

那么是否需要分离，就要看项目交付要求及频次，对项目的影响程度来决定。

一般真实的业务场景来看，前后端分离开发通常是属于较优的选择：

- 1.可同步开发，降低交付延期风险
- 2.完全接口化文档化，减少双方的理解和沟通成本
- 3.提前了解到项目的风险和难度，有更明确的心理预期
- 4.维护，维护，还是维护

对于第一点，在超过1人以上项目，协助能获得更多开发时间，绝对是降低延期风险最有效的手段。其二，前后端分离能较好地解决依赖困局，通过各种工具或者文档执行各自的开发进度，最重要的是，前后端分离后，采用的技术在实现和选择时不被束缚。为后期的维护带来方便。

结语

大多数情况下，前后端分离是更加普适、先进且合理的方式。

你觉得呢？

● 程序员是要专精，还是要广度？

- 作者：淘系技术
- 出品：淘系技术

于一名优秀的技术人员来说，究竟是专精一块技术方向，做到深耕其中所向披靡；还是谋求“什么都能略懂一点”的广度，成为一个全方位的人才？

这其实是一个职业发展和学习规划路线的问题，许多同学都有这个困扰。

今天我们邀请了 4 名淘系技术工程师，结合他们自身在小厂和大厂的经历故事，给大家分享一些他们在技术人员成长中对于【精】和【广】的选择观点，希望能够对你有帮助。

01

淘系技术部 - 应用算法 - 立青

“更早地认识自己和自己的方向，能更快地帮助我做出成绩。”

我想先撇开这个具体的问题，谈一谈一个程序员的技术发展和职业规划。

写代码这件事绝大多数人在喜爱的同时，更多的都是当做一个职业来做的，当然也确实有一部分人真的完全当做爱好，例如前段时间的新闻，HashiCorp 的创始人 Mitchell Hashimoto：“[顶级凡尔赛CTO辞职:写代码才最快乐!管理只会影响我搞研发](#)”；著名的Linux创始人Linus大神也是出了名的热爱编程。对于这样的追求（土豪），抛弃功利热爱编程，我觉得在编程上完全可以喜欢什么方面就追求什么方面，是广泛的发现兴趣探索兴趣，还是在一个方向上深入挖掘都没有问题，大可以今天做前端，明天搞开发，后天攒算法，一人搞定全栈。

但是对于绝大多数人来说，编程更多的是职业发展道路上一个立身的手艺，在众多专业技术方向上挑了一个自己比较喜欢和热爱的。程序员的发展和众多职位的发展一样，每个人都希望自己能够往“上”走：更专业，更能在职场上发挥自己的作用和影响力，从单兵作战做小事，到带队做大一点的事，再到影响一个领域，影响一个行业。这样的发展单单靠自己各方面都懂，都有涉猎，恐怕是不行的。刚毕业的应届同学可以靠自己的知识储备做自己的标签，久经职场的同学必须靠自己在某些领域做出的成绩做自己的军功章。

所以我们越早在某些方向做出自己的成绩，对自己的成长和发展是越好的。我本人是做算法的，算法领域有很多大牛在学校期间就已经找准了自己的发展方向，并做出了成绩，例如caffe作者@贾扬清大神，Taichi作者@胡渊鸣等等。他们的成就就需要[及早的找到自己的方向+不懈的努力+亿点点的天赋](#)……对我们绝大多数人来说更需要及早的找到一个热爱的，希望精通的方向并做出成绩。

我自己的经历其实不是好的榜样，我抱着做机器人的梦想本科学的机械，在学校的时候对什么都感兴趣，既参加过机器人大赛，参加过结构设计大赛，也和同学做过热力学相关的一些机械设计；一个比较有意思的经历是，当时在学校还做过一个上天失败的小卫星。当时学校的微小卫星研究所发射了一颗皮星(体积很小的卫星)之后，在学校办了一个小卫星的比赛，我们做了一个模拟卫星太阳能电池片自动追踪阳光的卫星模型，在答辩前熬夜完成了，然而在最后测试的时候由于线的固定太过粗糙，在运动中扯掉短路了，把电路都烧了……万幸留下了视频支持最后的答辩。对我后来职业选择影响比较大的是临毕业用Kinect做了一个姿态识别控制无人机飞行的项目，初步接触了视觉算法的一些知识。

后来研究生就真正的开始做起了机器人的方向，也是研究生时期做的控制算法和视觉算法的经历让我走上了程序员的道路。在这个过程中自己也是对机器人领域中的SLAM方向产生了兴趣，自己恶补了一些知识，并靠着这些最后走上了这个方向的职业。一方面我觉得自己如果能早一些找到自己的方向并且积累起来一些成果，肯定对自己的发展是要好很多的，但是另一方面每个人也确实需要一个认识自己的过程，但这个过程我想还是越快越好。在这个过程中，我们自己的技术发展就像是一棵树，我们尽可以无限的去展开自己的枝叶，多了解一些不同的方向和知识，但一定记住这是为了让自己的枝头长得更高。

02

淘系技术部 - 应用算法 - 朔明

“大部分的精力还是要用于巩固自己的长处上，你一定要有竞争力的依凭。”

做一件事是要精还是要广？其实相当于赌博里你是要多压，还是要单押。我们的筹码是有限的，当然我们的精力也是有限的，不可能去做所有的选择。那么这时候，问题就变成了如何去组合投资获得最大化的收益。

如果你选择去把所有的筹码压在一个选项上，那么你就一定要去承担选择失误带来一切清零的后果。但是我们也知道，如果你选择到一支潜力项，不费吹灰之力就可以赢到盆满钵满。所以孤注一掷，一定对应的高风险。

另外一个选项就是广博，它带给我们的好处是平摊风险，但是平摊风险的同时收益也会被稀释掉。比较好的一种做法，应该是两者相结合，在该广博的时候广博，在该专注的时候专注。

作为一名算法工程师，我的建议是，广泛的去涉猎相关领域的知识，以及弱相关领域的知识，因为这些知识全都可以作为你自己的储备，作为你专注行业的加成。比如说，你从事图像识别相关算法研究，那么直接的，除了图像相关的专业知识，一些基础的知识例如代数分析等数学理论，计算机原理与体系结构，甚至色彩，摄影美学等知识，都可以为模型设计带来一定的辅助。除基础的知识以外，相关领域比如自然语言处理当中的时序模型就与视频图像分割问题有很多的共通之处；信号处理中很多变换算子能够直接迁移到图像处理当中。这些知识，都可以作为你研究图像算法的加成。

又如，你想不到心理学，消费者行为学以及经济学都可以为我们的工业推荐体系赋能。在洞悉了消费者在购物时的选择动机，心理活动以及当下经济趋势后，算法设计者能够依据这些先验知识有聚焦地设计相关的模块，或是作为趋势因子加入算法当中。用一种更柔软的方式将大众认知融入模型，将表象和理论相结合，从而更好地发挥推荐系统的效能。

世界上很多知识底层是相通的，不要去抗拒学一些基础的知识，很有可能那将是你未来抓住机遇的契机。但是，大部分的精力还是要用于巩固自己的长处上，你一定要有竞争力的依凭。广泛的涉猎，精准地融会贯通，提取出你需要的那一部分，转化为自己需要的能量。

03

淘系技术部 - 前端技术 - 禾鸟

“我个人经历觉得，精进一门技术，不管是对于开发还是其他工作，都是重中之重！”

当我刚毕业的时候，在一家小厂做Flash游戏开发，由于当时开发人员配比严重不足（qiong），所以不但要用AS3去写前端游戏UI功能，还需要用NodeJS写服务端游戏逻辑、SQL处理数据读写操作、HTML/CSS/JS写后台配置管理页面，且一度在公司没有招到测试的情况下，所有的功能测试只能通过自测来完成。

时间一长就有些膨胀了，自我感觉非常好，觉得自己就是全栈，觉得只有更大的舞台才能配得上自己，进入大厂应该是轻而易举的事情，于是开始往网易、阿里、华为一些大厂投简历，Flash游戏开发、H5游戏开发、Web前端开发、NodeJS开发、服务端开发、测试开发各种岗位都投了遍。结果就是被现实狠狠地打了脸，投的简历石沉大海，少有的几个面试也都是一轮游。

后来总结反思了一下，在所有面试中，面试官经常会问这么几个问题：XX原理是什么？XX如何实现？对XX你是如何理解的？对于XX功能是否有更好的解决方案？面对这些问题，发现了自己其实对底层原理一窍不通，所有都还是停留在使用层面。

看清楚了这个问题之后，我暂时放下了其他域的学习，专注在前端领域的学习，从基础的HTML/CSS/JS入手，到lodash/JQuery等常用工具库的使用，再到Vue/React等主流框架的使用以及原理的学习，以及ES6、TS等学习掌握。等熟练掌握了这些技能之后，再深入到浏览器工作原理、网络通信机制、前端性能优化、稳定性安全保障等知识点的学习。一步一步从搬砖菜鸟变成了熟练搬砖工，然后继续朝着搬砖砖家努力。

总的来说，在小厂，老板当然希望能更省钱，巴不得一个人就能干完所有的活。而对于个人而言，精进一门技术，不管是对于开发还是其他工作，都是重中之重！

04

淘系技术部 - 移动开发 - 临境

“不要把自己当做业务研发的工具人。”

这个问题没有标准答案，角度不同，得出的结论会大相径庭。前不久刚从小厂跳到阿里，简单聊下我自己的感受。

小厂会更偏重于业务，因此我们更多的是在实现业务方的需求，日常的研发工作也少有接触特别难或者深的內容。这个阶段不是说基础不重要，但深入学习基础知识可能对你业务研发不会有特别明显的提升。

曾有段时间我为了学习算法知识跑去刷 leetcode，刷完两三百题后陷入了迷茫。这些刷题获得的算法知识在大部分的业务场景中不太能用上，虽然它对于在编程中边界条件判断、减少低效代码确实起到了一定的帮助，但相较于投入的时间来说性价比略低。

同样道理，学习操作系统、计算机网络、计算机组成原理能让你对整个计算机体系有一个更深层次的认识，但工作的人不比在校生，时间比较有限，如果你不是一个特别特别自律的人，投入时间没有看到明显产出很可能会中途放弃。

如果想要在工作的同时更深入学习一些计算机基础知识，可以从你正在做或者感兴趣的性能优化入手。不少性能优化手段背后都伴随着深入的知识，比如之前不少大厂分享的针对 iOS 二进制重排优化冷启动的内容。学习过程中一定会碰到自己不了解的知识，一点点去积累，会比盲目的抱着一本砖头书直接啃更容易坚持。

另外，如果是业务研发，请一定锻炼自己对业务的理解程度，不要只停留于需求来了做完就完了，把自己当业务的工具人没有对个人成长没有益处。

最后，和大家分享一句我和喜欢的话：“书上没有知识，书上只有信息；知识是在特殊的工作和行动中运用信息的能力。”共勉。

结语

程序员要精还是要广，并不是一个有着标准确定答案的选择题。

精于基础，广于工具，熟于业务，永远保持进步和学习的心态，希望各位都能找到最适合自己的技术成长路径。

● 程序员如何在业余时间提升自己？

- 作者：淘系技术
- 出品：淘系技术

在自省过程中，我们经常会问自己这么几个问题，这段时间我尝试了什么新事物、有了什么变化、得到什么成果。

近年来，出现越来越多“自主学习”、“业余提升”的相关话题。

我们经常收到一些同学提问：程序员的工作非常忙碌，如何在繁忙的工作中利用碎片化时间学习或是做自己感兴趣的事情？不管是新的应届生，还是两三年的职场人，甚至工作多年的职场老人都在关心这个问题。到底现在的程序员该如何利用业余时间从而让那个自己达到满意的状态呢？

今天我们邀请了 4 名淘系技术的工程师，给大家分享一些他们的业余提升技巧，以及时间的安排，希望能够为你提供一份参考。

01

淘系技术部 去来

这个话题非常现实，我的方法就是发现工作以外的时间在哪。

程序员的一天其实挺忙的，每天能留出的时间可以分几块：

通勤时间

我的通勤方式是地铁，每天大概需要花40分钟在地铁上，这段时间一般用来读书，一年累计下来能读完2-3本。

有些人可能不太爱看书，其实不同类型的书吸引力确实差很多，找到自己感兴趣的书才能培养阅读的习惯。

本科的时候我自己也是更爱打游戏不爱看书，在研究生阶段偶然看了一本基督山伯爵，才开始喜欢上阅读的。书里的很多故事不好拍成电视剧或电影，这些有趣的故事能让我在代码之外找到更多乐趣。那之后慢慢开始找书看，在朋友圈看看好友分享的，和同事们也有交流看到的好书。

前段时间恰逢建党100周年，组里的几个小伙伴毅安，增群对《毛泽东选集》产生很大兴趣，我也在读中共中央文献研究室出的《毛泽东传》，有很大收获。

午休时间

这段时间不是很长，吃完饭大概还有半个小时，习惯午睡的同学可以趁机休息会，为下午的工作养足精神。

我个人更习惯和同事遛弯交流交流最近生活上的新闻。

晚上下班后

一周锻炼2-3天，在外跑步跳绳，或在家跟着keep练。

没有好身体，心情也不好，其他都干不好，所以对自己的健康还是要多关注一些。

IT从业者白天坐的时间太久，用眼时间很长，晚上尽量别玩太久手机。

健身是给身体充电，读书是给头脑充电，利用一些更加碎片化的时间我会翻一翻知乎和微信公众号上和人工智能相关的技术专栏，看下行业动态。

周末

除了吃喝玩乐，继续阅读，健身还有写作。

阅读是被动接收信息，我还有一个小目标就是自己写文章，打造个人技术影响力的尝试。

去年定了一个小目标，微信公众号的关注超500（超过500就可以加广告，赚流量分成，哈哈）。写了几篇，只吸引了自己朋友圈里的人关注，分析原因：一是内容还不够精彩，二是微信公众号没有公域推荐机制。于是又转到知乎上，通过写文章与答题，目前已经收到1k多粉丝，分析了粉丝来源之后，发现大部分都是来自一两个优质的回答，头部效应很严重，所以再写文章还是要出精品，争取早日一篇自己的10w+。

02

淘系技术部 玖伍

其实提升自己没有秘籍和诀窍，只要愿意花业余时间去学习，再加上长时间的坚持，就可以成为大神。

阅读

我个人比较喜欢读书，喜欢读纸质的书，记得刚开始工作的时候，很多东西都不会，只会写CSS切页面，是一名真切图仔，同时自己又特别想成为大神，然后就每天中午吃完饭在工位上看一个小时的书，下班后也会留在公

司看两个小时的书再回家，就这样每天中午和晚上一边看书一边写Demo，前期的提升速度还是非常明显的，基本上每天都能感觉到自己学会了新知识。

我比较推荐多读一些技术书，特别是纸质书，熟悉我的同学都知道我有非常多的书。一本书从选题表到最终出版，中间会经历很多步骤，出版社专业的编辑也会和作者一起反复的校验和修改好多遍，上市之后再经过读者的认可，这样一本书的内容质量是非常有保障的。根据经验图灵出版的书质量都非常高。

学习资料

学习资料非常重要，要阅读高质量的第一手资料，很多时候我们学习某个技术发现怎么都学不会搞不懂时可能不一定是我们笨，也有可能是学习资料有问题。

我见过很多文章讲某个技术，即使那个技术我事先已经会了，也确实看不懂文章里在说些什么。我也见过很多文章可能作者自己也不是很懂某个技术，他只是把一些其他文章拼凑起来。

不好的学习资料通常内容晦涩难懂且没有把技术讲清楚，而高质量的学习资料通常会很清晰且精准地把一个技术讲透，因为讲解清晰明确，所以学习起来也不会太复杂枯燥。

JS框架、库、工具等，我一般会从官网和口碑较好的纸质书籍中学习。基础知识我一般通过阅读高质量的纸质书籍 + 阅读W3C的规范来学习。Web性能领域我通常在Chrome开发者官网和web.dev里的文章来学习。

具备一定的基础知识后就可以判断出学习资料的质量，这时候就可以关注一些公众号或者明星程序员来获取一些知识。

写作分享

除了学习，我还会利用业余时间写文章，做技术分享等，将自己学到的知识分享出去。切身体会，将自己学到的知识分享出去对自己的成长有很大帮助，有时候写文章的过程中会发现自己对某个知识也没有真的学透。

而且写作和分享可以让自己学会思考并锻炼思考能力，而思考能力其实很重要。

坚持

最后，坚持才是最重要的，我们的职业生涯，其实是一场没有终点的长跑比赛，很多人可能想问怎样才能跑得更快，把这场比赛跑赢。

大部分人跑到中途就主动放弃了，这就是为什么大牛那么少。唯一能决定这场比赛输赢的，只有两个字叫“坚持”。

在这条赛道上跑赢的，不是那些跑得快的人，而是为数不多坚持跑的人。

他们能跑赢，只是因为他们还在跑。

书单推荐

最后推荐一些书单，全都是我自己看过的觉得非常不错的书。

JavaScript相关的书籍：《你不知道的JavaScript》上中下三本、《深入理解ES6》、《JavaScript高级程序设计》

CSS相关的书：《CSS世界》（这类书我自己没有看，但我看张鑫旭博客学的CSS，他出版的书我虽然没看，但凭着对作者的信任，而且作者还专门为这本书做了个[官网](https://www.cssworld.cn/)感觉还是蛮用心的，质量应该是可以保障的）

JS框架相关的书籍：React相关我没有看过不做推荐，Vue相关的推荐一本：《深入浅出Vue.js》（非广告，内容质量和深度确实是目前市面上最好的一本）。

Node.js相关的书籍：《深入浅出Node.js》（只看过一本朴灵大大写的质量还行，别的没看过，所以只推荐这本）。

再分享下其他我看过的觉得不错的书：《算法4》、《Web性能权威指南》（作者是前任W3C性能工作组主席，译者是李松峰老师，虽然这本书出版快10年了，但我感觉还是值得一看的）、《重构》、《码农翻身》、《代码整洁之道》、《软技能 - 代码之外的生存指南》、《金字塔原理》。

03

淘系技术部 岳溪

这个话题非常现实，我的方法就是发现工作以外的时间在哪？程序员的一天其实挺忙的，每天能留出的时间可以分几块

业余时间如何分配，如何保持高效的工作状态？

- 1.首先还是要规律生活，早睡早起，比如晚上尽量逼迫到点就睡，晚上的熬夜必定带来白天的萎靡。

- 2.保持锻炼，规律健身跑步运动，缓解工作的疲劳，也给工作注入更多的体能资本，健身房的一声大喊，工作的疲劳一扫而尽。
- 3.业余时间还是要慢下来，品味生活，比如对某一时期的历史感兴趣，不妨系统研究研究，然后结合了解的知识，再制订一个长期的业务学习计划，总之还是要做一些让能力能够专注的事情，给生活多一些愉悦，需要愉悦。

工作上多线程切换真的会带来效率提升吗？

有时会感觉事情很多，陷入忙碌后容易在多件事情上来回切换，最后的结果经常是一件事都没有做好；普通人真的是不适合多线程工作，有时不妨多想想，要思考，适当的慢下来，分清事情的主次，然后集中精力一件件完成，往往效果会更好；

工作中经常会遇到新的领域，新的问题，高大尚的东东，面对新技术如何学习，快速入门？

还是要具体问题来分析，面对工作的难题，需要深入理解时，我通常会读读经验帖，找一本好书或资料，以解决问题为目标制订系统学习的分阶段目标，阶段目标可以防止总目标lost。

最后来一个团队广告，欢迎加入智能设计应用部，团队每周有篮球活动，定期有足球休闲赛，欢乐多多。

04

淘系技术部 勇剑

“业余时间”对自己提升相比“工作中”的提升，不同的地方是，我们可以有选择性的针对自己薄弱的点去提升。

那么首先要了解自己有哪些需要提升的地方，才能有针对性的去提升。

个人理解，作为技术人员提升的主要方向不外乎技术基础、沟通技巧、推动能力等等。

业余时间的话，主要可提升的还是技术基础，这个也比较好去有针对性的学习。

自我分析

首先需要有一份自己的技术栈大图，有哪些已经掌握的、哪些还不太熟悉、哪些完全不懂的，然后就可以针对自己的薄弱项进行针对性的学习。当然还有一些是我们还完全不知道的技术，这就需要我们经常去关注业界动态了，针对这些，可以视情况去参加一些线下的 meetup，跟大佬面对面的交流。

献上我自己的一份个人技术技术大图：



透过现象看本质

在学习过程中，要注意透过现象去看本质，常说的 What、How、Why 在学习的过程中要常去思考。

不给自己设限

总结来说就是不设限，尝试去寻找自己的突破点，在自己的技术边界不断拓展，而不是不断的去做一些重复的事情。

至于提升的方式那就很多了，首先一手技术资料肯定是各种官方网站，对于各种中间件的学习，**看源码**是最好的方式（github）；想更系统化的学习话，可以**读一些书、专业 paper**，都是比较不错的手段；还可以通过一些社区，跟其他人**一起学习，交流心得**，取长补短。

避免直接网上搜索的拿来主义式学习，另外看多少不代表你会多少，付诸实践、产出结果才是我们的最终目标。

结语

如上，不管是什么职业，我们都需要继续提升。也有很多人觉得不仅现下的生活如一潭死水，工作也没有任何提升。其实不外乎一个原因：想太多，做太少。

解决方法很简单，**放手去干**。

● 程序员如何快速上手一个新项目？

- 作者：淘系技术
- 出品：淘系技术

作为开发人员，我们不可避免地会遇到如下场景，一是接手前同事的项目，二是参与到新的项目组开发。

如果项目不紧急留给我们时间去了解业务还好，一旦项目紧急，则会让我们感觉到压力山大。这个时候必须要有一套行之有效的方案，能够引导我们快速步入正轨。

成熟的程序员，擅长从过往经验里总结出快速上手和熟悉新项目的技巧。

今天我们邀请了4名淘系技术工程师，给大家分享一些他们在接手新项目时的方法心得，希望对换工作或者换业务的你有帮助。

（文章很长，请耐心等待，我们还在文末为大家准备了福利）

不要事无巨细地请教老同学，要有Owner的心态

淘系技术部 - 淘系架构 - 光锥

在接手一个自己不太熟悉的新项目时，可以从以下几点思考以便快速熟悉系统。



熟悉业务

用户是谁、提供的核心功能是什么、系统在上下游的地位是什么。

不涉及具体的实现细节，通过核心功能产品层面的熟悉，能够对项目有一个全局性把握。

熟悉部署结构

最新的代码在哪，测试环境如何搭建，监控告警有哪些，线上如何部署，线上机器分布情况等等，通过自己亲自发布一次代码，观察哪些指标，了解整体的发布流程以及部署情况。

从问题中学习

系统较为复杂时，实现细节太多，直接上手看代码熟悉链路的实现细节效率较低，比较好的方式是通过实际问题，了解一个问题的来龙去脉，通过具体问题的修复过程中，逐步熟悉整个系统，但需要把熟悉的部分整理到整体的认识当中。

就好比玩一款拼图游戏，一个局部一个局部拼好，但心中始终要有一个全景视图，把局部的拼图一点点归纳到整体视图中。

owner的心态

接手一个系统，就需要以owner心态对待。

有些同学习惯性的事无巨细都请教老同学，心底有所依赖，缺少了一份独立思考，成长起来就相对较慢。

遇到疑问要首先要自己做一个判断，不论判断的是否正确，经过一次思考后，对系统的理解将会上一个台阶。

如果是我应该怎么做

在熟悉系统的过程中，可以多问一下，如果是我来设计这个项目，或者由我来实现这个功能，应该怎么做。

原有的项目可能由于历史原因，并不是以最优方式实现，对比自己期望的做法，可以快速了解到系统这样做的深层原因。

通过一次次对自己的追问中，可以更快的理解系统的深层背景，同时增加自身的设计能力。

而自己原有的技能，也能更好的反馈在新的项目当中。

认识各类大中型项目演进历程，有助于你真正理解一个项目

淘系技术部 - 淘系架构 - 家愿

工作多年，大中小型项目均经历过，分享一些我的经验。



首先，谈谈中大型项目是如何演进的。

部分中大型项目是慢慢演进而来的，这些项目一般都是从小项目或者相对简单的架构演变而来的，随着需求的叠加或当前的架构已不能支持用户规模逐步演变而来的。

演进方向一般都是从：单体 -> SOA -> 微服务 -> 云原生。

这类项目因时间周期长，迭代次数多，需求文档和设计文档一般都会存在缺失，一般都是文档落后于实现，且这类项目一般都有一些坑或者历史包袱，很难直接通过文档就直接将项目做到了然于心。

另一部分中大型项目则是近期设计开发完成，周期不是很长，一般是两年内的项目，这类项目因为是近期设计且中大型项目业务或架构都会相对比较复杂，在开发前都是经历了较为完整与严格的需求评审、技术选型/评审的，沉淀的文档和线上代码一般差距不会很大。

当然我们肯定乐意接触的项目都是后者，但是往往并不是事事都能如意，那么针对前者这种文档相对缺乏的项目，我们如何快速切入呢？

下面我们谈谈如何快速熟悉一个项目。

我们首先要了解业务背景，大部分情况下，技术都是为业务服务的，业务优先于技术，一般来讲，我们可以认为业务是一种商业模式，我们了解业务背景，其实本质上是了解这个业务背后的运转模式，基于运转模式进行深入了解。其实变相的就对整个系统进行了了解。



举个例子：比如我们接手了一个Push消息系统（消息推送），那么我们需要先知道Push是什么以及Push的运转模式，Push是什么？

Push可以简单认为是一种触发用户的方式，我们通过Push触达终端设备从而起到一个通知提醒的作用，一般情况下Push指的是无线设备上的应用通知。

Push的运转模式是怎样的呢，整个链路是怎样的呢？

我对Push的整个链路进行了一个抽象：消息请求 -> 消息受理(分发) -> 消息送出 -> 消息到达 -> 消息曝光 -> 消息点击 -> 后续业务转化，其中部分链路可以进行合并（例如消息受理和消息送出，但大致链路遵守这个模式）。

在App Push下，消息送出一般指投递给具体的通道（包括自建通道和厂商通道），那么是如何投递的呢？

我们接下来就去了解各个通道是如何提供服务的，是基于什么技术，我们需要如何使用，这样就把通道送出的大致原理搞明白了。

其实整个思路也是遵循了总分总的模式，我们首先了解项目的整体业务背景，整体的运转模式，然后基于运转模式里的各个点再进行深入了解，然后总结各个点，最后达到对整个项目有个整体的认知的地步。

运转模式其实是对业务的一个抽象，如果接手的项目没有直接的文档，那么可以通过实际上手使用相关功能或者咨询相关产品/运营同学了解，然后再自己归纳验证，整个过程下来，你就对这个项目是做什么的，用户是谁就有了一定的了解，我们接下就可以谈谈如何上手

在了解了项目的运转模式后，我们针对代码如何上手呢？

我们不要一上来就逐行去读源码，这样不仅效率低，而且容易一下子就陷入到细节中把自己绕晕，那么我们应该怎么做呢？

首先我们需要提前了解公司常见的技术，开发部署环境、常用中间件、DB等，对公司的技术有个大致的了解，和市面上相似的技术进行比较，了解每个技术的优缺点，这样我们后面才能知道为什么会用这个技术。

其次我们要寻找整个项目的重点和难点在哪里，针对一个有一定开发经验的同学，我们在了解到项目的运转模式后，我们可以想想，如何是让你来设计实现整个项目，你会怎么做，技术选型会如何选，哪里可能不好设计，哪里可能存在挑战。



在想清楚这个问题后，再来验证当前项目的技术架构是否与预想一致，不一致的地方想办法弄清楚为什么，项目的技术架构如何获取呢？

项目是由各个应用组成的，各个应用有自己的模块，各个应用的职责是什么一般很容易获取到，公司内部一般都能找到应用的描述，但是自己的模块划分怎么获取到呢？

如果有相关文档那当然很容易，但是如果没有文档，我们可以尝试问问同事了解，如果上述两招都不好使，只有代码，那我们只能从代码着手了。

好的设计可以直接从工程的目录结构上了解到应用的模块分类，从命名上知道模块大致的作用，其次我们可以从pom文件、打包脚本、接口类等对应用模块以及提供的服务能力进行初步了解（这里以Java工程为例），同时进行到这一步后，我们可以将应用以及应用内模块的功能进行了一个整理，形成一个文档；这样当我们需要实现一个需求或者需要修Bug的时候，我们可以快速知道这个功能可能需要涉及哪些应用以及模块。

最后就是进行实战了，在我们了解项目的运转模式后，我们对项目的架构、技术有了初步的了解，那么我们可以上手一个小需求了，或者尝试改改小bug，这时候才是我们需要扣细节的时候。

我们接到需求后，思考涉及的应用以及模块，然后去看当前这些模块的实现，搞懂相关的实现，如何搞懂呢？

有一些很好的实践，例如在本地完成工程的搭建后，针对代码进行debug（debug有很多小技巧可以提升效率，例如远程debug、条件断点等，可以现在网上进行学习掌握），了解每步实现的处理和数据变化，从而完全掌握这一个小模块或功能点的实现，这样在心中有数后，就可以愉快的将小需求进行开发了，多做几个小需求后，对应用的细节也就慢慢熟悉起来了。

十分推荐在项目熟悉了解过程中沉淀自己的资料

淘系技术部 - 前端技术 - 阮莹

时逢转岗到淘系一年的时间，正好借此机会来讲讲自己是如何落地一个全新业务和技术栈的项目。

十分推荐在项目了解过程中将自己了解的内容沉淀成资料，来明确和巩固自己的了解情况，同时有助自己和他人后续回顾。整个了解过程由粗到细主要分为三步：



第一步，了解业务

在上手新项目前，如果对该项目所在的业务并不了解，那么先大致了解下整体业务，以及新项目在整体业务中所处的位置，能帮助你后续更快了解新项目。可以重点关注与新项目有关的上下游业务。

过程中可以产出：业务大图、整体业务流程图。

第二步：了解项目/产品

功能&流程：通过试用项目对应的产品先建立对项目功能和流程的初步概念。

技术架构：通过项目现有设计资料并结合表结构、重点功能代码、鹰眼调用链信息等了解新项目的技术架构。

过程中可以产出：项目业务流程图、系统架构图、数据模型、重点逻辑流程图。

第三步：了解技术栈

如果新项目是和前项目完全不同的技术栈，那么你还需要了解新项目所在的技术生态。通过开发一个小功能来了解和熟悉新项目的项目风格和依赖的技术产品是不错。

从小需求开始，尝试编码和逐步矫正

淘系技术部 - 前端技术 - 正期

我总结下自己的经验，大概有如下几个步骤，简单做下分享：



第一步，了解项目架构，按照服务划分模块（预计耗时两天）

接手新的项目，一定是先了解项目架构；后端多以服务划分模块，所以我们以服务为维度对项目划分模块。

一般规范的项目，已经存在有比较完善的项目文档，可以快速了解项目的主要模块，形成大概的印象。

每个模块一般对应了一个git仓库代码，这时候必须记录下对应关系，这样给到某个需求的时候，我们才能知道具体的功能实现在哪个仓库中。

在这个过程中，一定不能只看，要动手做记录，可以是画流程图，可以是记录文档。否则看过一眼之后很容易忘记。

第二步：找准核心业务链路，将模块串起来，走读代码（预计耗时两天）

什么是核心业务链路呢？核心业务链路是指系统提供的主要服务的代码调用链路。

为什么选取核心业务链路研究呢？因为核心业务一般是由核心模块之间调用完成，研究完之后我们能够快速对核心模块有更加深刻的认识。

以手机中台为例，最核心的就是将手机的屏幕数据以视频流的方式传递给页面侧展示，这中间涉及

- 1.session-server模块提供的会话服务；
- 2.用户侧拿到session之后websocket直连device-agent模块；
- 3.设备侧的屏幕数据流采集和h264视频编码；
- 4.网页侧的播放器模块实现播放；

看懂这些模块后，基本也就对手机中台的核心模块有个大概了解了。

同样，这个过程需要进行画流程图加深印象。

第三步：从小需求开始，尝试编码

我们很容易走进一个误区，就是觉得自己对项目还不了解，不着急做需求。

其实一直看而不做，反而印象不深刻，为了学而学，总是收效甚微；相反，带着具体问题去看，逐步踩坑，才能快速上手；因为有些问题不做是不会发现的，比如代码规范问题，不去写，我们永远不知道自己和规范差多远，这是个逐步矫正的过程。

刚开始做的小需求不用求快，而是以规范为主。做完之后有了成就感，对我们也是一种正向激励。

总结

所谓“磨刀不误砍柴工”，淘系工程师推荐大家从项目本质出发，了解项目背后的业务，然后对业务模式进行抽丝剥茧，和应用实现相互对照，最后熟悉项目后，进一步通过完成需求来实现业务，并积极总结沉淀属于自己的理解内容和资料文档。

希望以上分享对大家有所帮助。

● “程序员从幼稚到成熟的标志是什么”？

- 作者：淘系技术
- 出品：淘系技术

任何人从幼稚到成熟的过程，都是从不负责到承担责任的过程。
人生如此，程序员也一样。

我们邀请了9位不同技术领域、不同层级的同学，从以下13个话题中筛选了各人适合的4个话题，针对性展开访谈，聊了聊他们的成长经历和故事，以及他们对职业发展、互联网焦虑、技术敏感度等各类的话题的思考。希望对正在成长的你有所启发。

话题表：

- 你的花名 & 工龄
- 你目前主要负责的技术/产品是什么？
- 当时为什么选择了加入淘系？
- 你在淘系工作的这段时间有哪些成长，是如何达成的？
- 你认为青年工程师最应该培养的素质是什么？
- 你觉得身边哪个人对你启发最大？为什么？
- 你有互联网的年龄焦虑吗？如何看待？
- 你觉得程序员从幼稚到成熟的标志是什么？
- 你觉得淘系对年轻人的成长提供了哪些支持？
- 你感觉自己工作中最困难的时候是什么？如何克服的？
- 你是如何保持学习的？
- 给想要在技术这条路走下去的年轻人一些建议？
- 你觉得程序员如何保持自己的技术敏感度？

01

前端 | 空堂

5年工龄

我实现了从「需求实现者」到「促使目标达成者」的观念转变。



你目前主要负责的技术/产品是什么？

手淘弹窗搭投平台 Poplayer 内的弹窗编辑器（xEditor）与多端弹窗渲染引擎（xRender）。同时还负责淘系前端技术公众号的运营，欢迎关注：）。

你在淘系工作的这段时间有哪些成长，是如何达成的？

最大的成长是刷新了对「完成工作」的认知。以往我比较满足于自己「工匠」的角色，认为只要能在规定的时间内，最大限度的保障架构设计和代码的质量，满足用户的需要，并提出一些专业可行的建议，就可以算是出色的完成了阶段性的工作。

但去年双 12 和我的 TL 元泉一起经历的一个项目让我意识到，仅仅是「较为理想地完成了被分配的开发任务」还不足以为一段工作画上句点。一阶段工作的完成应该是以「项目下线为标志」。这意味着，作为开发者，工作并没有在项目成功上线后结束，在项目在线的整个生命周期里，都应该尽可能地关注，思考，实践通过哪些途径，手段可以帮助项目更好的达成，甚至超越既定的目标。

这种从「需求实现者」到「促使目标达成者」的观念转变，让我在每一次项目中能够拿到「更多的经验」，也有

机会创造出更显著，更大的价值。令我受益匪浅。

你认为青年工程师最应该培养的素质是什么？

「敬业心」和「方向感」。按时完成任务是敬业的一种表现，但除此之外，最大限度的保障代码质量，为代码负责，不断学习，不断探索更优的需求实现方式也是敬业心的一种表现。我建议青年工程师尽早做好「终生学习」的打算，时常将自己的技术打磨锋利，这样才能在工作和生活中找到一个稳固的立足点，始终保持心态的平和。

我发现一些工程师会逐渐迷失在源源不断的需求中，最终磨灭掉工作的激情。这对于雇主和自身来说都是十分可惜的，因此我建议这些工程师要尽早规划自己的职业发展方向，技术领域专家？行业领域专家？转行？无论思考的结果是什么，在工作中有一个清晰的发展目标总是比浑浑噩噩、固步自封好太多。如果目标与当下的工作内容脱节严重，建议和主管聊聊。如果实在想不出未来该如何发展，同样建议和自己的主管聊聊，和更多优秀的同事聊聊，聊这些问题没什么大不了的。

你觉得身边哪个人对你启发最大？为什么？

TL 吧，和他一起经历了一个项目，转变了我对工作的一些认识，前面有提到，就不赘述了。

02

客户端 | 来一

5年工龄

所有来自「事」的困难都可以用态度解决，所有来自「人」的困难都可以用换位思考解决。

你目前主要负责的技术/产品是什么？

目前主要负责 Weex2.0 的技术开发，致力于打造高性能、优秀的开发体验的跨端技术产品

你在淘系工作的这段时间有哪些成长，是如何达成的？

- ①学习到了在阿里的一些做事方法，目标感，推进项目落地，拿结果。
- ②要沉下心来做事，不浮躁，但行好事，莫问前程，最终的结果也是水到渠成。
- ③要善于树立自己的个人品牌，以及不要给自己设限。

主要就是通过不断的参与项目进行历练，以及学习身边的老板/师兄的做事方式，多观察周围比自己优秀的同学



你觉得程序员从幼稚到成熟的标志是什么？

- ① 技术能力——遇到技术问题不慌乱，有自己标准的排查思路，解决后要能想清楚：为什么会有这个问题、怎么解决的、以后怎么避免。
- ② 技术预判——能在特定的场景下有技术选型的能力，涉及到包含但不限于业务的需求、各技术方案之间的利弊等的综合判断。
- ③ 持续学习——长期来看，专业技能任然是立身之本，要清楚意识到这一定是未来投资回报率最高的事情之一。
- ④ 终极——程序员等到了所谓的年龄大限之期时，不焦虑，自信自己的技术能力、技术视野等，那一定是到了高阶，自勉。

你觉得身边哪个人对你启发最大？为什么？

是我刚进公司时的老板——舒文

- ① 做事态度与换位思考——印象最深说过的话是，所有来自“事”的困难都可以用态度解决，所有来自“人”的困难都可以用换位思考解决。仔细想想，这句话是真精辟，工作/生活中都适用。
- ② 阿里做事方式——要有持续的目标感，推进事情落地，快速拿结果的能力

03

前端 | 永霸

7年工龄

保持好奇心，如果想要做，就立马去做。



你目前主要负责的技术/产品是什么？

负责淘系基础产品（首页/详情/店铺/交易）前端开发

你觉得淘系对年轻人的成长提供了哪些支持？

第一、淘系复杂的业务场景与用户体验的极致追求，为技术人提供非常好的场景与机会去锻炼自己的能力。古语有云，无根之木，无源之水。业务就是我们的根，我们的源。当市面上常规的技术方面已经无法满足业务需要的时候，就要求技术去探索与设计的更好的方案。

第二、淘系有一堆业务与技术领域的顶尖人才，为年轻人的成长做了非常好的榜样力量。在我们遇到技术难题、成长困境的时候，可以很容易找到比我们有经验的前辈请教。正所谓站在巨人的肩膀上，站的高看的远。以我所在的淘系前端团队为例，在跨端、搭建、工程化、ServerLess、前端智能化等各个领域内都是国内最顶尖的存在。

你是如何保持学习的？

保持学习我认为主要是学习意愿与学习条件问题，前者回答为什么要学习，包括工作需要、生活需要、兴趣等。我认为最重要的是保持好奇心；后者回答要学习但条件是否允许，个人建议是如果想要做，那就立马去做好了。

不同的学习目的，我的学习方式不太一样。如果是工作需要，我个人偏向追求最佳的解决方案；如果是生活需要，我个人可能比较佛系，能 work 就行；如果是兴趣使然，那么自己开心就好。

在学习形式上，主要有两种：第一、从书本上学习，主要是为了体系化的补充自己知识短板；第二、跟身边优秀的同事学习，时刻保持好奇与谦卑。

你有互联网的年龄焦虑吗？如何看待？

暂时我没有太多焦虑。这个行业大家可能普遍比较关注的是 35 岁危机。

我是这么理解这个问题的，在半导体行业有个非常有名的摩尔定律：【集成电路上可以容纳的晶体管数目在大约每经过18个月便会增加一倍。换言之，处理器的性能每隔两年翻一倍】。摩尔定律反映了互联网领域技术迭代更新比较快，特定技术的经验积累，随着时间发展可能会逐步失效。这就要求行业从业者需要不断的学习以跟上摩尔定律的发展，否则就要落后。但随着年龄的增长、家庭的责任以及自身身体素质的变化，在学习意愿与学习条件上的平衡可能会被打破。当出现这个情况的时候，会导致缺乏安全感，出现焦虑。如何应对呢？我想可能是需要在工作过程中，逐步积累对应的工作能力，包括定义问题与解决问题能力。同时问题的范畴也可以逐步变大，例如，具体问题->领域问题->行业问题。

04

算法 | 伯研

10年工龄

成功在于持之以恒的坚持和努力，时间的积累才会带来量变到质变。



你目前主要负责的技术/产品是什么？

淘宝人工智能推荐

你在淘系工作的这段时间有哪些成长，是如何达成的？

首先是对团队的工作职责有了深入的了解。其次对团队的同学进行了全面的了解和熟悉，和团队同学构建相互的信任，这点是最重要和宝贵的事情。最后在业务技能方面熟练掌握了目前业务线上使用的技术框架、算法模型和业务流程，在和相关同学的交流和帮助下，深入参与了一些几个推荐业务的研发，并且融入了自己以往的技术积累和经验，带来一些创新性的技术方案，给业务带来更好的用户体验。

你有互联网的年龄焦虑吗？如何看待？

有的时候会有一些年龄焦虑。因为过几年自己也会很接近35岁，目前的层级距离自我预期还是有一定的差距。每每有这种焦虑的时候需要自我好好的调节，努力做好工作，创造更多的业务价值，提升自己的综合水平，等待晋升的机会，心态也是还好的。

你觉得身边哪个人对你启发最大？为什么？

三桐老师对我启发很大。坚守在一个技术方向，多年持之以恒，不断提升团队技术架构和产品用户体验，给我的启发是“成功在于持之以恒的坚持和努力，时间的积累才会带来量变到质变。”

05

算法 | 陆轶

3年工龄

保持技术敏感度，需要持续学习和多与同行交流。



你目前主要负责的技术/产品是什么？

手淘首页商品信息流推荐，排序大模型方向。

当时为什么选择了加入淘系？

我是硕士毕业就加入淘系至今，一直从事推荐算法工作。当时觉得淘宝作为国内最大的电商平台，坐拥海量数据，非常适合从事推荐算法工作，我们的工作在服务数亿消费者，优化算法模型可以创造巨大的商业价值，非常有成就感。

你感觉自己工作中最困难的时候是什么？如何克服的？

最困难的时候应该是我的师兄离开阿里的时候。以前在学校的时候有导师、刚进入工作时有一个很好的师兄指导规划方向，师兄离开后自己突然转变需要去规划技术方向，完成了一个大的角色转变。后来在主管的帮助下，不断学习去思考技术方向，判断技术是否能给业务带来价值，在这个过程也获得很大的成长。

你觉得程序员如何保持自己的技术敏感度？

我觉得最关键的是两点，持续学习和多与同行交流。

技术发展日新月异，会不断涌现出新的技术、新的算法模型等，要保持持续学习，主动吸收学习新的信息和技术成果，持续进行自己知识体系的迭代。多与同行交流，保持一定的求知欲，从同行的角度多方面了解最近的技术方向，结合自己的知识体系进行判断。

06

开发 | 十段

1年工龄

技术不应该只做业务支撑，而是要以技术的视角去思考如何帮助业务成长。

你目前主要负责的技术/产品是什么？

淘宝直播直播机构&商业化后链路承接相关业务



当时为什么选择了加入淘系？

淘宝是阿里的核心产品，想在一个大平台上服务更多的用户，让更多的用户用到我做的产品功能。

你觉得程序员从幼稚到成熟的标志是什么？

可以在适宜的场景和时间，选择当前最合适的技术方案。大炮打蚊子是不可取的，螳臂挡车也是不可取的，选择当前时间最合适、成本最低的技术方案才是架构师该做的事情。

年轻的时候，总是想要把技术方案设计到极致，但是过于在乎技术方案本身的做法，也会导致花费了很高的成本做的事情却达不到回报。互联网时代下，各厂的产品日新月异，每时每刻都有新秀在崛起。合理选择技术方案也会让产品迭代的更快，可能在快速变化的环境下，也会让产品更有竞争力。这是我觉得这是一个程序员从幼稚走向成熟的标志。

你觉得身边哪个人对你启发最大？

龙甲，让我很好地理解了技术与业务之间的关系，明白了技术人员在业务成长过程中应该怎么做。技术不仅仅只能做对业务的支撑与建设，同时对业务也可以以不同的视角带来多维度的提升。技术和运营以及产品看问题的视角是差别很大的，技术在业务迭代、增长的过程中，可能以独特的视角发现其中的问题和优化点。

举个例子，我现在在做的商业化后链路承接，产品可能主要侧重于以产品化的形式去发现更多的商业模式以及提效点。但是技术的视角则不同，比如我们可以提供更多的用户、直播、商家、商品特征，帮助广告推荐引擎让流量变得更加精准。技术不应该只做业务支撑，而是要以技术的视角去思考如何帮助业务成长，这是对我比较大的一个启发点。

06

开发 | 赤杨

11年工龄

青年工程师面临的问题，本质上是成长和走出舒适区的问题。



你目前主要负责的技术/产品是什么？

网关与基础网络

你在淘系工作的这段时间有哪些成长，是如何达成的？

以前从事音视频工作，来淘系后切入到QUIC协议领域，掌握了QUIC协议相关的知识、上手了基础网关相关的工作，突破了自己的知识边界。

你认为青年工程师最应该培养的素质是什么？你在淘系工作的这段时间有哪些成长，是如何达成的？

乐观积极，勇于突破。

青年工程师面临的问题，本质上是成长和走出舒适区的问题，需要在技术、业务、管理等层面上实现突破，建立认知的立体框架。虽然工程师的本职是为公司提供技术支撑，但这个过程会存在诸多超出本职工作的事务。比如有时候研发工程师遇到问题，需要跳出自己模块，联系上下游的其他模块进行解决；有时候研发还要承担项目经理、产品经理的工作。

这些过程，是一个局部层面的工作在水平和垂直方向的衍生，包括了对系统局部到整体的认知突破、对团队技术到管理的认知突破。因此面对这些问题，青年工程师首先要能够积极面对，敢于跳出舒适区，从而实现自我的提升。

你觉得身边哪个人对你启发最大？为什么？

在同喵吉、哲良的工作过程中，发现他们经常做一些超出本职的事情，比如某项目申请商标时，发现商标被注册，同时法务告知商标持有方不愿转让，喵吉主动打电话与商标持有方沟通，并最终达成了商标购买。

这个事情对我感触很深，很多时候我们都会碰到障碍，从而导致方案、进度受阻，这种突破自我界限的精神，恰好是破壁所需的基本精神。

还有一个就是，哲良是怎么保持这么瘦的，我感觉研究哲良保持瘦的秘诀，说不定淘系每年的绩效都暴增几十亿图片。

08

算法 | 琦舞

5年工龄

技术终归是为业务服务的，业务起飞了，才能证明技术的真正价值。



你目前主要负责的技术/产品是什么？

主要负责首页猜你喜欢短视频全屏页推荐业务

你在淘系工作的这段时间有哪些成长，是如何达成的？

我是社招过来的，在淘系工作的时间不长。虽然只有几个月的时间，但最大的成长并不是技术方面的，而是业务owner意识和责任感，你在做出某些决策时，若想到这个决策有可能会影响到几十万几百万甚至几千万用户，你就会更加谨慎和用心。每次实验，若是效率提升的实验，可能伴随着用户短期乃至长期留存的提升，但若是效率下降的实验，可能伴随的是用户的流失，这是我们不得不小心对待的。

你感觉自己工作中最困难的时候是什么？如何克服的？

最困难的时候是算法策略在上线初期没有效果，有点焦虑，但经过认真分析实验数据，复盘反思，不断尝试新实验新思路，最终拿到了理想的线上结果。

给想要在技术这条路走下去的年轻人一些建议？

如果真的热爱技术，就请热爱下去，紧跟技术潮流，做好活到老，学到老的准备，因为IT行业的更新速度之快超乎想象，我们必须让自己的知识储备不断更新不断升级，才能紧跟时代步伐，不被社会淘汰。除此之外，大家可以与业务再近一些，技术终归是为业务服务的，只有业务起飞了，才能证明技术的真正价值。

09

测试 | 荣木

7年工龄

多项工作并行的情况下，要合理高效的分配时间。

你目前主要负责的技术/产品是什么？

测试开发工作 / 手机淘宝（连接消费者与商家的消息业务）

你感觉自己工作中最困难的时候是什么？如何克服的？

多项工作并行的情况下合理高效的分配时间，并能积极拥抱变化，保障最高效的输出，例如你在测着需求、思考着用例设计、敲着代码，突然爆发了个线上问题；又或者你需要同时参与A/B/C项目，A、B、C项目里面原本定的计划发生了改变、某个高优先级项目临时被阻塞了、某个项目线上出故障了需要紧急响应。



- 1) 及时调整优先级 & 合理评估时间：可以建一个list，并根据各个项目情况与风险甚至线上故障问题积极应对，值得注意的是：根据“墨菲定律”，事情所费时间大部分情况比你预期的多，任何事情都会出差错，而且是在最坏的时刻出差错，因此需要考虑backup time。
- 2) 及时上报风险：受限于经验或领域知识的影响，可能我们并不能做出最好的决定或选择，出现风险较大的问题或者可能存在风险时需要及时向上级或团队反馈，依靠团队的力量做最正确的决定。
- 3) 思考提效策略：拿出一部分时间做复盘总结并投入精力思考过程中的处理措施是否合理，哪些环节可以提效，哪些环节可以更合理的使用时间和精力。例如，从质量角度来看，自动化测试、问题监控与定位是一个非常好的思考主题，由于不同的业务形态差异，当前并没有一个绝对通用的方案，需要根据不同的项目、不同的业务场景分析，选择最优、最合理的策略。从PM或管理者的角度来说，策略安排显得尤为重要，例如：推进阻塞问题的解决、多方协同作战等等。
- 4) 申请资源：当前策略安排下没有足够人力投入时，应该及时反馈，申请资源援助，使用团队的力量解决问题。

5) 保持良好积极、乐观的情绪：紧急项目连续作战、面临各方面的压力同时可能会存在一些负面消极情绪，应当及时调整，可以约好朋友吃个饭、聊个天、打个台球、玩个游戏分散下负面情绪的注意力。

你是如何保持学习的？

1) 保持入行时的好奇心：当前互联网新技术日新月异、百花齐放，以一个初学者的心态对技术保持好奇。例如：PC自动化测试框架 -> 移动自动化框架 -> 智能化自动化 -> ...，这中间需要 we 与时俱进，为学习时刻准备着。

2) 了解行业动向：参加行业大会、技术沙龙、阅读知名博客、国内外高质量论文资料，帮助我们更好地思考、找到正确的方向

3) 拥有技术信仰：相信技术会让生活变得更好，相信技术能够创造更大的价值以及自我价值的实现。

给想要在技术这条路走下去的年轻人一些建议？

1) 锻炼更好、更强的团队合作精神：能够和团队成员打成一片，和谐的氛围能够聚集漫天星星，发挥最大力量；

2) 拥有对技术的渴望：尝试新技术、新事物，通过不同的渠道（论文、大会、博客、技术沙龙等等）了解技术动向、行业关注的热点问题，沉淀技术、升华自己

3) 有时间可以了解/做一些工作相关性不大或者一些艺术修养方面的知识/事情，一方面可以使自己生活多一些色彩，另一方面可能会发现一些新的观点、视角来看待技术问题

4) 快乐工作、认真生活，拥有积极乐观的情绪

结语

成长意味着改变，成熟是一种选择。

正如来一同学说，“所有来自「事」的困难都可以用态度解决，所有来自「人」的困难都可以用换位思考解决。”

淘系同学在面对技术和业务的难题和困境时，展现出了乐观的个性和开阔的格局，他们不拘陈规，勇于创新；他们积极交流，保持谦逊。

一直在成长的人，不会畏惧年龄危机。

共勉。

一线职场&学习问答实录

• 为什么阿里巴巴超级喜欢java开发？

- 作者：风胜
- 出品：淘系技术

我在阿里12年，使用Java从业15年，经历了国内互联网发展的各个阶段，也见证了Java语言从兴起到成为世界排名前三的开发语言。我使用过C、VB、PHP、ASP、C#、Python、Java等语言，对于各种语言的优缺点有一定的体会，下面结合我在阿里的经历，给大家讲一讲阿里为什么这么喜欢用Java开发。

两个原因，

- 一是淘宝初创时期的最佳选择，
- 二是技术传承和商业发展的必然选择。

为什么Java是淘宝初创时期的最佳选择？

淘宝是2003年创立的，当时为了快速上线，购买了一个基于LAMP架构的PHPAuction网站，最初的几个月跑下来还是很好的满足了需求，但随着用户量和交易量的飞速增加，各种各样的性能问题就爆发了出来，不得不考虑换一种更加高效和稳定的架构。这时候Java就进入了候选列表。J

Java是一门很好的语言。它简单易学，没有指针，不需要手工管理内存，与平台无关，文档工具齐全，生态系统完善，性能和稳定性也非常好，学习门槛也低，有其他语言基础基本可以无师自通，这对于人才的培养与招聘都是非常有利的。

Java拥有大量杀手级应用。如：Eclipse、Idea、Struts、Spring、Hibernate、iBatis、Tomcat、JBoss、Hadoop、Hive、HBase、Elastic Search、Netty及Apache工具箱，这些应用或工具大都久经考验，开箱即用，遵循业界标准，还开源免费，不管做什么类型的网站开发，基本都能找到相应的解决方案。

Java适合做高并发网站开发。回到2003年代，门户网站还处在鼎盛时期，网页三剑客是程序员争相学习的技术之一，Ajax和jQuery还没开始流行，PHP、ASP(.NET)和JSP是互联网常用的网站开发语言。PHP和ASP对支撑门户网站这种网址导航和新闻资讯类业务是绰绰有余的，但对于涉及商品、营销、交易、支付、物流和售后

这种全商业流程的高并发电商业务，稍显力不从心。淘宝早期的实践也证明PHP要支撑大流量高并发的网站要付出的代价很大（具体历史可参考《淘宝技术这十年》、《尽在双11》等阿里出版的书籍），而Java生态里面拥有大量成熟的解决方案，往往付出较低的研发成本就能解决棘手的问题。

综上所述，Java是当时最受欢迎的语言之一，开发者社区活跃，拥有完善的企业开发框架，非常适合大型互联网企业协同开发，而且也被世界上主流的大规模网站普遍采用，eBay和Amazon就大量使用了Java语言，淘宝选择Java也就是一个必然的结果，所以才有了后来在Sun公司专家的主导下淘宝全面从PHP转向Java，一直延续到现在。

为什么说技术传承和商业发展的必然选择？

经过10几年的发展，阿里已经有拥有数以万计的Java开发人员，在Java技术栈有着非常深厚的积累，并推动着电商行业的技术不断发展，在Java界拥有广泛的影响力和号召力。

阿里自研了大量Java高性能中间件。包括RPC框架HSF/Dubbo、MVC框架WebX，消息队列Notify/MetaQ、分布式数据库访问层TDDL、分布式缓存Tair、分布式文件系统OSS、分布式网关TOP/MTOP、分布式日志监控鹰眼、大数据离在线分析引擎ODPS/FLink，金融级数据库OceanBase等，很多中间件在阿里云均有商业化版本，部分中间件也回馈给了开源社区。

阿里推动了互联网低成本建站方案。淘宝早期为了快速支撑高并发业务，采用了昂贵IMB小型机、Oracle数据库和EMC存储，后面发现性能和扩展性无法满足业务高速发展要求，成本也是一个不可承受之痛。后来才有了广为人知的阿里去IOE运动，通过采用Linux虚拟机取代IBM小型机、定制MySQL取代Oracle数据库、自研Java分布式存储取代EMC存储。去IOE奠定了阿里自研技术的发展路径，支撑了阿里电商业务的高速发展，也孵化了世界排名第三的阿里云计算平台。

阿里内部研发生产几乎都跑在Java语言构建的工具之上。从需求管理、代码托管、集成发布、运行容器、数据存储、监控预警、日志分析等全流程工具大部分都是基于Java自主研发。阿里这个商业操作系统全身上下都流淌着Java的血液，喜欢Java是必然的，也必须喜欢Java。

另外，阿里服务端开发以Java为主，但也在大量使用其他语言，阿里系的各个App大量使用了C和C++，在运维场景和部分中间件中大量使用了Python和Go，在面向合作伙伴提供的SDK也大量使用了.Net、Ruby、PHP等技术。

综上，我们对于 java 还是有比较多的实践经验，也很荣幸成为大家讨论和学习的对象，欢迎加入我们，也欢迎各类技术交流~

• 阿里P7、P8、P9级别需要多少年的工作经验，还是因能力而定？

- 作者：舒文
- 出品：淘系技术

P4：懵懂学生入行记 (2008.10–2009.12)

2008年10月，入职阿里巴巴日文站，以前端岗加入UED团队。日文站 (<http://www.alibaba.co.jp>) 的业务源于阿里巴巴国际站 – 提供在线的平台化服务，专注撮合中日贸易。

在那个前端的鸿蒙年代，页面重构工程师和 Javascript 程序员还是两个细分职位：在阿里，前端、交互、视觉共同划属用户体验部门。

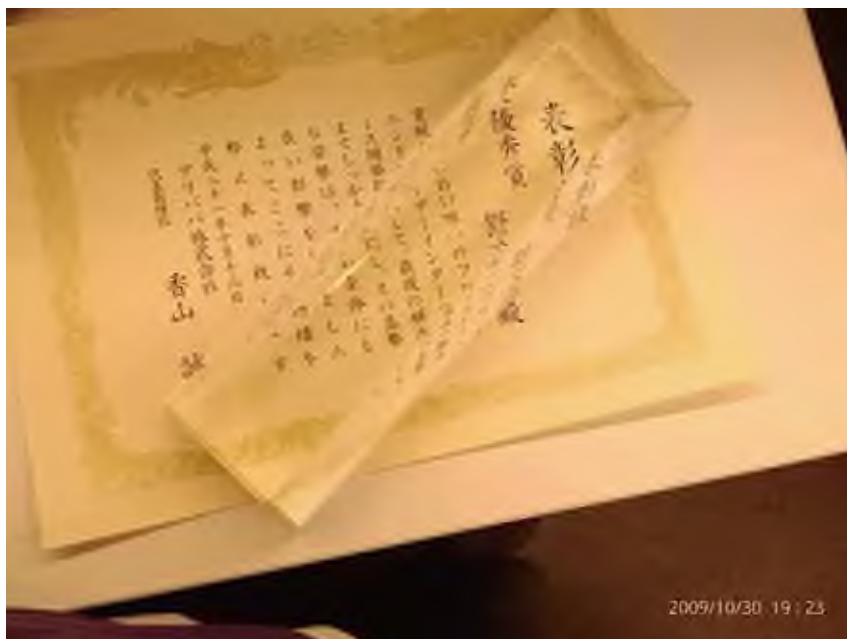
而在部分公司，有专门的重构工程师。

在日文站亦如是：现实情况是杭州和东京的团队，具备写JS能力的工程师均不多。不少同事专注在 HTML-CSS 领域，且专研极深。举个例子：在日文站，所有的 HTML/CSS 代码必须通过两个Lint工具：HTML文書の文法をチェックし、採点します、[The W3C Markup Validation – 这让代码的规范性更好外，也能获得更好的搜索引擎分析&索引权重。

随着业务发展，需求场景多且复杂起来，对Javascript开发能力要求就越高。同时，也激发了我的技术热情。除了看书，我迫切想参与其他事业部的前端交流会。感谢时任主管的支持，我开始主动跨部门沟通，将他山之玉和学习所得有了更多结合。慢慢地，在团队内部提出一些技术方案并逐渐应用到杭州/东京两个团队 (特别感谢最早时代的 D2)。

在那个阶段，我几乎参与了所有较为复杂的项目。结合当年的热门大作《高性能网站建设指南》、Yahoo 网站性能优化黄金法则，负责了主站的性能优化，取得了一些不错结果。

2009年10月获得了日文站优秀员工，年底顺利晋升到新的层级(P5)。



P5：从热血到成长(2010.01–2011.06)

2010年开始，业务略微调整，参与“日本买家获得”(Buyer Growth)。这个业务本质是通过平台技术优化(SEO、SEM等)获得更高的搜索引擎权重、提高 Landing 到转化。

在这期间，随着被认可度提升、持续的项目历练，我的自信心也得到了锻炼。除了参与更复杂的项目外，我开始主动对业务和技术体系做出一些提案(Proposal) – 包括参考 Google Webmaster Guidelines 进行技术优化、参与了各种 SEO 项目，提出了现在回想起来各种或靠谱或不靠谱的提案。非常感谢历任主管的支持和包容，让一个楞头青能够没有后顾之忧地敢想敢做。

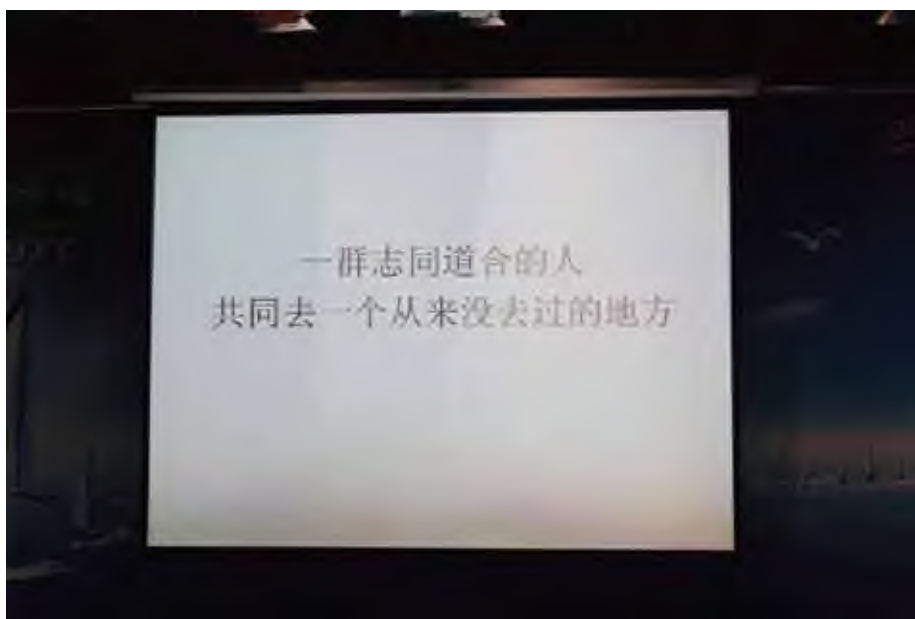
日文站初版克隆自<http://Alibaba.com>，加之外包参与、跨东京和杭州多团队开发等原因，全站前端框架混杂，包括但不限于YUI/Mootools/PrototypeJS/ExtJS/TBra等前端框架&类库混杂在同一业务甚至页面。新人上手、多人协作、性能优化的成本很高，但历史原因积重难返，大家苦不堪言。

结束一个大项目后，时值“愤青”的我对全站的代码做了个调研和分析，说服了主管们，决定启动一个重构项目——旨在将全站混杂的代码统一到 jQuery。并大体确定了执行方案：随业务迭代上线、项目成员1人，我兼任 PM、开发/自测。

在之后，一个如同打了鸡血般的前端程序员，白天黑夜的翻历史 PRD、熟悉代码逻辑、阅读各种类库的手册，陆续将各业务线的脚本重写了一遍，边做业务项目边重构，整个项目持续了半年多，直到 11 年 Q1 完成。

现在回想起来，有些事情真需要一股冲动，正如买房、结婚。如果主管让我再多考虑半天，我兴许就怂了。当然，人生也可能走入另外一条主线。

2011年日文站业务合入 B2B，我也参加并通过新事业部的年中晋升(至P6)。



2011年阿里巴巴日文站誓师大会

P6：内部创业 + 前端的无线爆发 (2011.07-2014.06)

两段非常宝贵又特别的经历。2011年下半年主动选择加入阿里内部的创业项目：打造一款面向个人消费者的云产品。团队不大，20多人，各路专家云集，各负责人级别都很高。很享受这段工作旅程，我第一次参与桌面 Hybrid 端项目，第一次开发 SPA Web 应用，第一次参与跨桌面软件 /PC Web/H5 的项目，第一次领略敏捷管理的魅力。在来自美国的技术 Leader 带领下，我接触到很多新领域，极大地开阔了技术视野。

“创业失败是要承担风险的，没有加薪，没有晋升，而且还是 996，直到项目成功”。第一次面试这个项目前，业务负责人，阿里工号 16 的虚竹告诉我。

不算幸运的是，最终这个项目没有取得如预期的成功，业务有了变化和调整。

但幸运的是：这段经历让我亲身体验了一把“创业”，也感受了技术以外的方方面面艰难。这个过程中形成的产品和业务思维，对我日后带来了深刻影响。当然，也在我之后几百次想离职创业时，这段经历让我能够静下心来思考创业的内核 - “为什么要创业？我有什么资源、能做什么事情、创造什么价值？”



湖畔花园办公场地

2012年7月，加入到天猫的前端团队。主管给予了充分的信任，让我负责天猫 H5 首页的研发，开始接触淘宝前端——一个更成熟完整的技术体系。

2012 年底，@三七 加入天猫带领前端团队，作为国内前端领域最早的拓荒者之一，他在随后的两年时间给天猫的前端技术带来体系化的变革——从模块化到工程化再到生产环境的 NodeJS，并引入了 Mobile First 理念。在他的支持下，我带领了一支前端小组陆续参与多条业务线的开发工作。我自己也加入集团级技术建设，主导跨端 Web 的项目，推进了前端体系的移动化建设。

除专业技术外，花了比原来更多的功夫在团队工作上，包括不限于团队氛围、风格 and 文化的打造，学会在内网分享中学习借鉴前辈大拿的管理方法论。在阿里的文化中，非常强调传承。一个人牛不算什么，让一群人厉害才算厉害，最好是让下属比自己还强。

时间花在哪儿，结果就会在哪儿。团队内一些业务尖兵逐渐冒出头来，在后来几年成为天猫甚至大淘系的中坚力量。

这个过程既痛苦又快乐。痛苦在于要做越来越多的超过自己能力层级、远离舒适区的工作，快乐的则是能明显感知到自己在过程中的快速成长。

很幸运，在 14 年年中，顺利通过晋升(P7)。

P7：从双十一中打将出来 (2014.08 – 2016.01)

2014年注定是难忘的一年。在这一年，我开始负责营销活动的业务，并担任 2014 年双十一前端技术 PM 工作。

营销活动是一个顶有趣、富有故事的业务：

从技术上：它可以极致简洁到切一个页面上线就行，也可以复杂到如双 11 一般——它是阿里技术的年度大考。

从业务上：它既可以简单到直接把现实生活的促销活动进行虚拟世界的进行流程投射，也能如互联网史上的春运般整合零售生态和供应链。

我和多个团队在天猫制定了各个维度的技术规范，如内控标准、外包规范，有惊无险地渡过了14年双十一。



在 PC 时代，营销活动的研发模式，对于前端来说，实在过于“简朴”：“5到6位正式前端常年带着数十个外包，根据运营需求开发成百数千的页面，通过一个叫做 TMS 的运营系统预留坑位给运营同学填数据，交由后端应用推上线（CDN）。”这类业务活动频次高、页面量大、协同成本高。某种意义上讲，对前端技术挑战并不高，为了更高效的研发，我们能做的是不断提高组件和模块复用率。

感谢TMS，它帮助前端们快速迭代出一个又一个页面，支持了集团的业务，即使在今天，它的设计思路都能称得上精巧犀利。

发生变化来自 2014 年双十一前，线上发生了一次令我终生难忘的离奇故障（涉及敏感信息，略过细节）：当时因临时未排查出原因，VP 现场观摩面，我和 CDN 运维团队的同学们能做的只是不断重启集群应用缓解问题。

我作为时任前端 PM，也因技术方案选择失误受到批评责罚。（时隔多年，大家也偶尔调侃，责罚不冤，幸而那天的问题提前暴露而未发生在 11.11 号当天，否则后果难料）。

那年双 11 最终还是完美谢幕，GMV 571 亿，移动端成交 42.6%，很多人敏锐的感觉到：移动化时代到来了。

之后技术复盘，我们针对问题做了多轮讨论和推演，决定启动一个代号“斑马”的产品项目：基于我们对营销的理解，运营的实际需求、过往的技术痛点，设计一个高效的页面系统来支持运营快速发布活动上线，更重要的是它是完全的基于 Mobile First 设计的系统。

正所谓“一饮一啄，莫非前定”，因为过去一年的打磨锤炼、痛过后的反思，我们很快把方案框架敲定，并和多个业务搭档共同推进项目实施。

项目于 15 年初立项，我代表团队“激进”的将阶段目标定为“支撑 2015 年 3 月的春季大促”。这种既要支撑日常高频的大促业务，又要同时从 0 到 1 做平台研发，不啻于“飞行中升级引擎”，困难很大。但我背后的思考是：“这就如同创业，没有一往无前的决心和一鼓作气的气势，等想多了细了，多半就怂了怕了”。

我负责整个系统及 PM 工作，团队培养出来的中坚力量们承担各个功能模块研发。为实现我们心中所定计划，整个项目组如同创业般激情，为了一个共同的目标奋进。春节假期不少同学还在主动提交代码，节后迅速返回公司继续项目。

很幸运，克服了重重困难，“斑马”项目顺利上线，并成功支持了 2015 年第一个 S 级大促。基于新的系统，我们共同把营销活动送上新的轨道：因模块化带来的效率提升让我们完全解除了外包依赖、因具备跨端能力运营们可以单次同时进行 PC 和无线活动、我们设计并推动了 CDN 构架，单在 15 年双 11 就节省了数千万预算。

随着技术迭代，“斑马”这个技术产品，在后面几年成为阿里的基础运营平台之一。

2015 年 5 月，我参与了 P8 的晋升提名，结果未通过。Leader 和我沟通：“成长很快，能力和规划出色，业务成果需要时间验证”。我很坦然接收了建议，晋升答辩中的技术评委的建议中肯不失公正，让我受益很大。

“但行好事，莫问前程”——在随后的时间，“斑马”的能力更为完备。我再次担任了 2015 年双 11 的前端 PM 工作，业务也顺利上线。那一年，双 11 当天 GMV 912 亿，移动端成交 68%。

16年1月，新主管告诉我，绿色通道通过，晋升至P8。



双十一令牌/虎符

P8：立足业务，敢想敢造 (2016.02 – 2019.07)

一次偶然的机会，看到一张图：如遵循民意苹果手机会变成什么样？给我带来一些对业务和技术的思考：“页面发布系统本质上是一个通用的工具平台：前端/运营/设计师都是它的用户，每个角色都有对它的功能需求。但每个新增功能在解决一类用户需求的同时又增加了平台复杂度，降低的客户的易用性，丢失掉另一类用户的民心。”

“这兴许是一个产品的轮回宿命”，我内心有些敬畏地想。带着这个敬畏，时间到了2017年，随着斑马的功能日趋复杂，我决定将运营操作平台和底层技术能力进行剥离：

- 将斑马定义为运营系统，专为运营打造极致的页面搭建。
- 将底层能力抽象成服务平台（代号“天马”），提供开放的研发标准、搭建方案、页面渲染能力。

随着时间推移，“天马”的开放能力不断完善，越来越多的事业部基于它构建面向业务的平台系统，带来的技术回报是：技术复用减少从0到1的研发&硬件成本、标准统一使跨业务协同变为可行。

而在技术的另外一边，业务发生着悄无声息的变化：16年双11GMV 1207亿、17年双11则是1682亿。每年双

十一在如此巨大的基数下还能持续业务高增长，个性化算法、推荐技术的大规模应用起到了非常重要的作用。在全局效率上，大数据优于人工干预逐渐得到了共识。在我看来，这些变化意味着时下大家认可的大促会场模式，也将带来新的变化。

2018年2月，我在内部做了个分享，描述了对营销活动未来的预判：

- 业务化：面向“活动”组织，而非页面。
- 产品化：构建常态化、心智化的大促产品矩阵，而非单纯是模块级的抽象。
- 智能化：投放算法化和规则化，而非运营的动态化。

基于以上预判，我和团队的小伙伴，计划联同业务、上下游技术团队共同构建一个面向营销活动的平台产品（项目代号：方舟）。

很感谢历任主管实仙/四虎的信任，愿意帮我顶住重重压力，经过多轮沟通，我们达成了一致共识。

和过去几年稍有不同的是，我逐渐把包括不限于方舟的一些重要产品放手给团队中的骨干/TL，让他们去规划、去驱动、去变革。我更多做了辅导性、资源性、业务判断的全局性工作。

随着时间的推移，越来越多的结果开始凸现：

1. 斑马逐渐成为集团级的通用平台，数万小二同学用它发布管理业务。
2. 方舟作为业务平台交付成为大促体系的核心部分，并成功应用到双11期间，在效率、效果和交付质量上取得瞩目的结果。
3. 天马通过“开放共建、出海上云”，为集团多个BU提供基础的技术搭建服务。
4. 在业务中，团队中沉淀出一个个技术产品：渲染引擎、API聚合网关等，也把曾经仅服务于内部的产品开放到商家、ISV。

在2018年中，提名参加晋升面试，结果未通过，技术委员会的评委们建议更多参与“集团级的技术建设、更高格局的视野”。

我很快走出了沮丧，毕竟在阿里，常胜不常有失败倒常伴。更重要是，主管 @四虎 的开导让我开始思考更全局的未来。

随后的时间，我继续投入了2018的双11前端整体工作，推进了多个端技术方案落地天猫。感谢阿里前端技术委员会主席圆心的举荐，我加入前端委员会，担任搭建技术方向的Sponsor - 旨在站在集团视角，定制标准、融

合并打通搭建技术体系，在更大范围内赋能业务。

在集团内各个前端Leader的支持下，我们很快跨多个事业群达成了共识，制定完善了技术标准，基于一体化方案启动了多个层面的项目。

2019年5月，参加了年度晋升，顺利通过P9晋升。感恩。

P9：不止于前端

现在，仍然有太多的事情值得去深入：

1. 以双11为代表的大促前端体系仍然是业内最具技术挑战的业务场景之一，包括不限于客户端容器技术、服务端渲染(Node)、框架与组件体系、跨终端技术等综合应用。
2. 阿里有数以亿计的消费者，如何为我们的消费者构建一个好玩有趣的互动购物体验，是我们这个团队一直需要探索和改进的。
3. 在未来，源自于前端的搭建技术，不仅能支持小二还能服务生态角色，不仅能支持国内还要服务全球。

除以上，我也深度参与阿里前端技术委员会的工作：

在阿里的前端体系，除了搭建方向以外，有相当数量的跨事业群共同建设的方向&项目，包括不限于Serverless、IDE、智能化、中后台、数据可视化、工程化、Node技术等。这些技术方向或者始于前端，但又不止于前端，共同为打造行业领先的技术生态服务。

有期待有更多的优秀的同学加入阿里，让我们的智力和努力触碰到数以十亿计的用户。

一家之言 for 技术人

最后分享一些过往多年的心得：

关于工作：

- 1.想做好一件事情前，获得上级的认可非常重要，这是把一件事情做成、做好的催化剂。

- 2.很多时候，源自“事”的困难都可以用态度解决，来自“人”的困难可以用换位思考解决。
- 3.对于刚参加工作的同学，如果家中无矿，啥也别说，努力就对了！毕业五年内的表现，可以决定人生很多事情。
- 4.如果你和领导发生分歧：拿出你的数据/理由/态度尽最大的努力去说服他，如果充分沟通还形不成共识，那就先听他的。这不是媚上，而是“在必须达成一致的前提下，相信更有概率做出正确决定的人。”
- 5.在工作中，如果某个时间突然发现和往常不一样，陡然压力变大/身心疲惫，别害怕，因为，这有可能是你在成长、突破瓶颈前的黑暗期。
- 6.专业技能是立身之本。从长期来看，它是性价比最高的投资标的之一。
- 7.结合资源，尽最大的努力、用最好的态度，做好手上的工作，也是一种创业。
- 8.不要因为背靠大树久了，就误认为自己是颗大树。对客户&用户的尊重和价值创造是我们很大的护城河。

关于生活：

- 1.技术人读书工作挣钱养家，一步步成长同时也会一岁岁变老。相比不少行业，互联网行业自身有赛道上的优势，但如果单纯以同比其他行业略高的溢价按年/月出售自己的时间和技能，这不应该是我们做的。坚持不断的思考、通过技术优势带来叠加价值，在过程中不断成长，这兴许会更好。
- 2.学会奖励自己。如果经受了辛劳、痛苦和压力走了过来，再不好好的、肉痛地奖励自己一把，哪对得起过去和未来的自己。
- 3.除了在工作中，人生处处皆可学习：学理财、学打球、学拍照、学游泳、学健身、学拍短视频、学习怎么把生活过得更好。
- 4.最后，咱们技术人可能很忙，有可能没法“work-life balance”，这是一种取舍。但健康的身体一定是最宝贵的一笔财富，没有之一，不能舍。谢谢大家。

● 去阿里淘系做技术怎么样？值得2021届毕业生去吗？

- 作者：空骋
- 出品：淘系技术

校招的时候，淘系不会因为你不是这个方向来限制你

那会儿毕业的时候，我发现自己图形学方向就业形势严峻。对口的公司不多，主要是用在VR和AR方面，于是我就拓宽了自己的求职范围。

面试淘系的时候，我是属于跨方向和跨语言的，之前的实习经验也只写过一些客户端上的脚本语言。我印象很深的是，当时面试官不会刻意刁难我不懂的领域，他主要根据我在学校做过的项目和发过的论文，询问我一些比较开放的问题，比如项目实践中自己的思路和想法等等。

语言技能本身并不重要，这些都是可迁移的，更重要的是未来的可塑性。我觉得一些思维想法和热情很关键。

当时也是BAT的 offer 都集齐了，最终选择了淘系后端开发。一方面是我在杭州念书，喜欢杭州想要留下；另一方面是我比较看重淘宝这个平台，毕竟是航母级别的APP，这个结构和系统，我想要深入地学习和了解。

第一年的时候，我在淘宝做“网关&技术支持”

刚入职的时候是很亢奋的，一个现成儿的、成熟的系统摆在了我眼前，千万级QPS，学校里面从未接触过这么庞大的系统。所有的淘宝业务请求都要经过这个网关，它的核心关键是稳定性保障。

初期我的大部分工作就是维护它，小心翼翼那种。接到需求的时候我都非常谨慎，牵一发而动全身，一个小改动就会影响很多东西，我经常要拖着我的师兄和同事们一起来看变更的影响面。对当时的萌新来说，冲击力很大，见识得也多。

另外，我也负责这个系统面向全集团的答疑。没想到的是，答疑这活儿，很快让我自己陷入了迷茫。

咨询量和问题量很大，我总是尽快尽好地回复大家的问题，结果我发现自己工作效率极为低下，忙里忙外毫无产出…

阿里对新入职的人都会配一个师兄，从工作到生活，帮助全方面的适应和学习。师兄很快就发现了我的这个问题，他提醒我，要学会规划自己的时间，安排事情的优先顺序。在他的启发下，我做了这样4件事：

1. 设置固定的答疑工作时间
2. 仔细盘查所有咨询问题，对于重复度高的问题，配置自动回复机器人
3. 整合和重构已有的用户帮助文档&答疑话术，沿用至今
4. 思考咨询问题背后的通用需求是什么，如何从根本上解决

这样，我既帮业务解决了问题，又解放了自己的劳动力。

还有很多这样的時候，比如当初刚来的时候师兄给我讲了团队结构的演进和每个人负责的模块，这种“人”的视角其实很有意思，让我更深刻地理解了目前的产品为什么是这个样子，和我能够发挥的作用。

我非常感谢淘系的师兄给我的指导和帮助。

第二年的时候，我开始做从0到1的事情

这个转身还是挺猛的。19年4月之后，我们新拉了一个团队出来，做一个全新的研发平台。

没有任何历史包袱，我们需要系统性地去思考 and 搭建一个新的东西，不止是技术开发式的自嗨。这个东西的场景、价值和意义，在构建的时候就需要去不断思考，甚至延展思考业务未来的衍变。

我们探寻过，很多复杂的问题业界都没有特别明确的或者成熟的解决方案，社区上一些开源的东西也没有实践验证。淘宝是一家商业公司，重视的是业务价值，我们看重的是这个东西是否真的可以用。

太多问题扑面而来，这些事情都是我在旧系统里做维护不会考虑的。

我之前一直烦恼，上班以后似乎没办法系统地学一块东西，都是零零碎碎 get 各种技能。主管和师兄告诉我，不用太纠结这个，现在的学习目的是为了解决问题，是在不断的实战和锻炼中成长，学的东西可以立马得到实践和检验。而且从更高阶的学习层面来说，通过零零碎碎的学习，长成自己的一块知识系统，才是真正的消化吸收。

如果说在淘系的技术发展前景，对我来说，现在做新平台研发的经历，让我慢慢确立自己以后想要走架构师的方向。

从我最开始负责的模块开始，从我去划分我代码里的功能点和接口开始，我一点点学会拆解和抽象一个需求，将

它延展成小块去解决。随着我能力的提升，我的视角和边界会不断拓宽，需要抽象的问题也会越来越复杂。研发平台-需求-写代码-测试-底层的资源交付，全链路的问题和视角是非常繁琐的，这也是我想要不断去精进自己的方向。

在淘系，这样的机会是非常多的，能够去交流和讨论的前辈和同事也很多。我也会看集团内部的ATA技术社区，看阿里学习的网站，大家沉淀的专业经验和知识都非常宝贵。

最后，想对学弟学妹们说的话

我来淘系3年了。总的来说，这3年的成长符合自己的预期，更高兴的是找到了自己喜欢和亢奋的方向。

如果回到我挑offer的那一年，就像是题主目前的选择境况之中，以我的过往经验，我觉得主要是取舍问题。

首先，是怎么获得这些机会？这些不用细说，提升自己的硬实力。

其次，是怎么取舍平衡这些机会？最重要的还是遵从自己的内心，每一个人的情况不一样，有的人最重视地点，有的人最重视薪资等等。你如果明确知道自己最想要什么，就按这个去选，当然更好的情况是你有一个长远规划，知道自己的发展方向。

另外从过来人的角度说，薪资这个事儿要稍微放长远看，格局大一点，校招的起薪差距远不如三五年后的差距大，多多和师兄前辈交流，有些东西是进来以后才能看到的。

最后，我所在团队是淘系技术部-淘系架构团队，负责一站式serverless研发平台研发建设，不断为业务提升研发效率和极致研发体验。平台已平稳支撑淘系互动、淘宝人生、金币庄园、闲鱼、特价版等多个业务的6.18、双11、双12、春晚等多个大促活动。如果你对云原生、Serverless/FaaS、K8S或基础架构感兴趣的话，欢迎加入我们！邮箱：zhiheng.gao@alibaba-inc.com

其他团队也很好，欢迎大家来淘系技术，共同交流进步~

● 面试官如何判断面试者的机器学习水平？

- 作者：友闻
- 出品：淘系技术

许多面试官掉进了坑里，对于候选人主要关注研究新进展、模型优化方法等等基础知识的考核，也关注了对于调参技巧等等技能考核。

但实际上公司最关注的是三个问题：

一是候选人是否能够解决当前的问题，这里面对于实际问题的理解和抽象能力尤其关键；

二是团队合作，候选人是否可以与其他同学形成合力，共同实现目标；

最后一个候选人入职后会有绩效、晋升的需求，候选人成长潜力如何，是否能和团队一起共同成长。

这3个问题都是需要关注的，在某些方面会比机器学习的基础知识更重要。

下面我会回答这样几个问题：

- 1、面试如何设计？
- 2、问题解决能力如何考察？
- 3、团队合作能力如何考察？
- 4、成长潜力如何考察？
- 5、候选人如何准备面试。

1、面试的设计。

面试需要的是尽快完成对候选人的考察，从而判断是否录用候选人，对于面试官来说，需要考虑的是是否接纳候选人成为未来的伙伴。

需要注意的是，面试题不宜经常变动，用相同/相似的题目面试不同候选人，会看到不同候选人的差异，有助于快速做出正确的判断，也就是大家常说的结构化面试。

面试的环节一般分为几步，自我介绍、算法/项目问答、编程题/逻辑题、最后总结。

面试首先是看到候选人的优点，而不是寻找缺点，首先的自我介绍环节，对候选人进行全面的了解，同时让候选人放松下来，以便正常的发挥自己的水平。

算法/项目问答，一个投递岗位是算法工程师的同学，如果没有合理的理由，算法的基本理解是需要过关的，对于低级别候选人，要现有深度再有广度，许多面试官刨根问底，也是为了考察候选人的思考深度。

编程题/逻辑题，为什么算法工程师也会这么重视编程题和逻辑题呢？目前的许多候选人算法背景差距很大，隔行如隔山，面试官与候选人钻研领域有时候并非匹配，考察的复杂度就会大大增加，而编程题和逻辑题考察每个候选人都可以使用，面试官非常喜欢这种题目。

最后的环节建立在前面环节的基础上进行考察，如果在知识和技能上失分太多，这个环节可以忽略，如果前面的环节表现比较好，可以和候选人深入交流。交流对未来的想法，对自己的优缺点的认识，从而建立更完整的对候选人的认知，看看是否匹配公司需求，避免做出错误判断。

2、问题解决能力如何考察？

对于问题解决能力的考察，关注的不是问题是否被候选人完整解决，而是得到答案的思考路径和方法如何。

具体关注三个能力：

一是读题能力，是否能够正确定义问题；

二是分析能力，识别问题中的关键点和难点，确定解题思路；

三是应用能力，是否能将已有方法在具体情境中应用，真正掌握了先进技术。

一种方法是逻辑题或者编程题，首先关注候选人是否能够通过沟通和互动，对问题建立正确的理解。其次是看候选人面对问题的时候，是能够进入问题情境，就事论事的分析问题，一步步建立对问题理解，还是说急于把未知问题套到一个已知问题的框架内解决，或者是在思考不完整的情况下轻易下判断。考察的时候一定不要用题面清晰，解答复杂的成题，这样的题目候选人即使能解决也是背下来解题思路。

我曾经用过这样一道面试题，12个球，其中有一个坏球，坏球和好球重量不同，用一个电子秤把坏球找出来，至少需要多少次。

这是一道很老的IOI题目，有的同学学过天平称球，就想套用天平称球，发现不适用，还是跳不出来思考框架，

这样的同学在面对实际问题的时候，也缺乏灵活的解决办法。

还有的同学着急去探索最优解法，而实际问题中大部分情况，我们只需要可行解，然后在可行解基础上不断优化即可。

这道题如果是忘记脑中的各种套路，从题目本身思考，得到可行解甚至次优解并不难，也可以排除一些不可能的情况，逼近最优解。如果能够对于问题的本质进行思考，从而给出解题的完整思路，因为考试时间问题无法做出最优解，也是可以获得高分的。

另一种方法是把同学拉到一个具体的算法情境中来看，就像是有个高赞回答提到的现场调试算法，但现场调试需要时间比较长，题目的设计也需要下功夫，还需要有个题库，避免漏题，公司里有专门的同学负责此事的话可以考虑。更通用的方法是对于同学应该比较熟悉的领域，构造一个实际场景，观察同学的思考角度和方法。

比如同学对图像识别比较熟悉，可以问问同学，一个每个类目目前只有10条数据，大约100个类目的家具分类问题如何解决比较好？回答的时候既包括模型选型，也包括标注任务设计，迁移学习理解，实际问题中样本不平衡、噪声样本等问题的解决。

学术界对于这些单一问题都有相关的解决方案，但是在具体情境下的决策能力，还是非常考验候选人这些方法融会贯通、灵活运用能力。

3、团队合作能力如何考察？

团队合作能力在面试中很难考察的很深入，最好是候选人实习一段时间，才能完整判断合作能力。面试过程中，主要关注候选人沟通中的一些特质体现。

第一，候选人在面试环节中是很自我的在思考、表达，还是尊重面试官、尊重题目。

第二，候选人的沟通能力如何，是否能够快速理解面试官问题，有疑问的时候是否能用沟通解决疑问。

第三，候选人表达能力如何，能够简明扼要的说清楚自己的观点，还是说表达缺乏重点，效率很低。

4、成长潜力如何？

成长潜力体现在自控力、主动性、学习能力几个方面。

自控力和主动性的考察，主要是体现在简历考核中，候选人是能够主动探索自己希望学习的方向，还是被动的接受学校的培养计划。当自己确定了努力的方向，是否能够按照自己的期望进行努力，还是说很容易放弃。

学习能力体现在题目考察中，读题的时候理解题目速度如何，分析题目时是否有很强的逻辑性，能够快速找到问题的关键。

成长潜力的考核是贯穿面试整个过程的，面试后需要尽快进行总结，从各个细节中形成完整的判断，候选人是否成长潜力合格。

5、候选人如何准备面试？

准备面试分为2个阶段，一个是长期积累，另一个是短期冲刺。

学习能力、学术能力、解题能力都不是短期能够突击的。

希望大家建立正确的学习观念，面试的目的是双方相互了解，进行匹配，通过一些短期手段，即时通过了面试，后续的工作也是很痛苦的。

一是长期能力培养。

leetcode是很好的工具，大家不要简单的去满足题目通过了，而是是否能够独立分析题面，理解题意，解题的时候都有哪些方法解决，优缺点如何，解题方法选型依据如何。坚持这样的思考过程，做了几百道题，编程和解决问题的能力自然提升了。

读论文、实验复现也是如此，首先关注论文的核心贡献，思考论文面对的问题背景如何，你有什么想法，论文中设计了什么样的解决方案，为什么这么设计。

主动性的培养可以尝试主动去参加比赛，以及各种活动，并不断改进方法提升成绩。

二是短期冲刺。

冲刺的目的是完整的展示自己，避免面试的短时间无法利用好，展示自己的优点。

首先是整理好简历以及相关内容的了解，说清楚自己的核心优点，简历内容中相关的支撑点都有哪些。

其次是对自己的项目，看看相关资料，说清楚难点和自己的价值，也对于领域内其他人的相关工作可以简要说明。

最后是多演练，找同学模拟练习，表达熟练了，情境熟悉了，自然面试表现会好不少。

最后，我们团队也在招人~

商业机器智能部基础算法团队，负责淘宝、天猫等电商平台相关算法研究和创新，在KDD、ICCV、CVPR、NIPS、AAAI等国际顶级会议上发表多篇论文。

在3D机器学习方面，我们探索2D视觉算法和3D几何及视觉算法融合的先进技术，在3D建模、AI设计等方面处于国内领先水平，期待推动家居行业的技术革新。

• 简历中怎么写「项目经历」最好？为什么？

- 作者：空聘
- 出品：淘系技术

作为阿里淘系面试官，对过去3个月我模拟面试过的30多位同学的面试情况总结，和大家聊一下在简历中如何书写项目经历，以及找到属于自己的亮点。

本篇回答以前端技术岗位为例，其他岗位可参考书写逻辑。

你负责的业务是什么？（学会发现问题）

我接触过不少同学，有说在小公司，被业务需求压着。既然大家都说在做业务，那么，正看到这里的你，能不能5分钟说明白，你负责的业务是什么？

这个问题我在活动群的 github issue 活动中，带有业务理解标签的题目里经常会问到，可是大部分同学都没有说到位，甚至答非所问。

这里谈谈我个人对业务的理解，或许没有普遍意义，所以仅供参考。

业务最核心的要素是业务本身的价值

一家公司，或者一个部门，做的事情有许许多多，零零散散。也有很多事情合到一起，促成了一件大事的时候。那么，我们是把那些零散的事情都看成业务？还是只把那一件大事看成业务呢？我认为都可以。决定权在于这件事是否逻辑自洽，以及是否具有独特的价值。

接下来让我们拿着一个例子来说，假设你在开发一个营销活动页，这个页面能够给公司带来3000人的新用户，这些人有可能会购买公司的产品，从而带来收入。

这里明显可以感受到，营销是一个业务线，他的商业逻辑是投放页面 -> 拉新回流 -> 商品销售，价值在于新用户的触达，以及商品销售利益。基于这两点，我们就值得投入精力，因为做的越好，公司业绩越好。

那么，做个页面就是亮点了？

当然不是，但是亮点已经离我们很近了。如果你想要有亮点，那你需要保持思考。在上面的例子中，我们有许多

可以优化和验证的事情。

- 1.营销页每天换内容，怎么快速替换？
- 2.营销部门人越来越多了，页面每天要10个，一个人怎么做得完？
- 3.前端的人也越来越多了，改个组件不能只靠复制黏贴，怎么管理？
- 4.拉新回流效率具体有多高？新人真的有买我们的商品吗？这么多人投入，都是要工资的，卖出去的商品能够发我们的工资吗？
- 5.转化率低了，怎么才能提升？
- 6.这个按钮写错个样式到了右边，居然点的人特别多？那下次是不是都应该放右边？

上面列举的几个问题，估计很多同学日常都有做类似的事情。但问题是，这些事情是你想做的，还是产品让你做的？这些事情能诞生什么出来呢？

- 1.运营配置后台与投放策略
- 2.营销搭建体系
- 3.工程化研发套件
- 4.业务埋点与数据分析系统
- 5.数据仓库与数据分析后台
- 6.A/B test系统

至少在我看来，如果面试的同学上来自我介绍的时候，能够讲一下上面例子中遇到的问题，之后再说做了下面对应的某一个系统，那么，这就是绝对够分量的亮点。只可惜这样的同学少之又少，大部分同学是因为产品说要做就去做了。

所以，你真的想过业务是什么吗？有为业务想过什么吗？有了你，业务有什么不同吗？

可以开始写代码了？(学会思考的方式)

好了，假设我们思考了一下，想了点东西出来，接下来我们可以开始写代码了.....吗？

做一个有亮点的技术产出，可不是撸起袖子就能快速干出来的，当然，如果你是个天才，那请自便。如果和我一样是普通人，那么请先做好技术方案设计。而设计的第一步，就是做一个ppt工程师，画图。

图，是思想的结晶

在上面提到过的github issue活动里，大部分同学的业务大图或者技术架构图，都没法说明白先表达的意思。

几个最典型的问题是：

- 1.思路混乱：下面几个框在写业务的系统，上面画了一个vue或者webpack的框。
- 2.层级混乱：底层写的是native容器，上层画了个api gateway。
- 3.答非所问：要求画业务大图，结果画了一堆前端脚手架的关键字，或者画成了流程图。

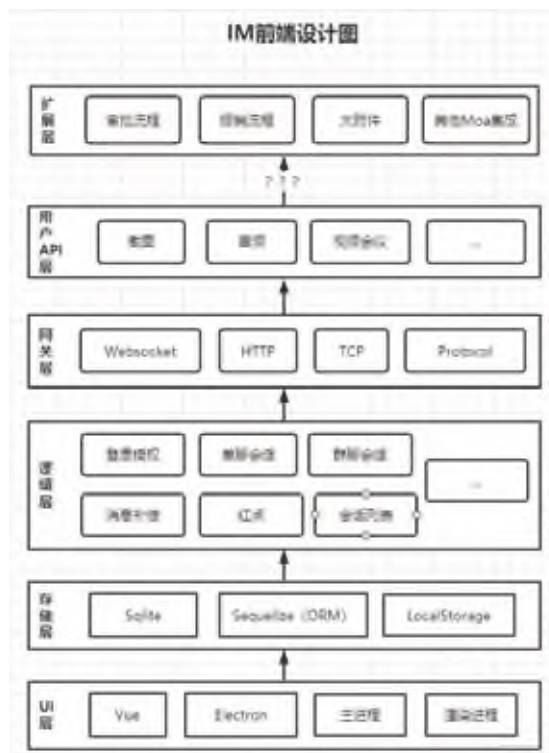
如果看到这里，不明白画图是干什么的同学，可以去查一下架构图是什么，以及如何做程序设计。这经常是被大家忽略的事情，虽然很多同学在大学里学习的时候，都学过相关的课程，但是估计大部分都还回去了。

怎么画好一张图？

这里不做具体的展开，毕竟我自己也不是画图高手，每次画图也是迟迟不知如何下笔。只给到几个建议，供大家参考。

同时，以一个模拟面试同学的案例来做参考。

原图：



第一步，先想明白这张图要表达什么？

这位同学说他参加过很多技术会议，看那些分享的ppt里面的大图，都很酷炫，自己平时也有总结(这点非常好)，但是总画不出那种图来。面试过程中我问了这位同学，这张图他想表达什么，答案是他想说明白消息通信业务的技术方案。但是，这张图并不能表达出一个技术方案来。

这张图第一个问题是不够完整，他只有一条主链路，对于IM这样的复杂技术产品，主链路只是冰山一角，如果真的只做了主链路，那么代表思考不够，早晚会线上故障。

第二个问题在于含义不明与层次混乱。最下面的UI层有个箭头指向存储层，那是指渲染进程会去调用localStorage？那再向上2级的网关层呢？UI层会调用网关层？这里显然逻辑是不通顺的。

第二步，图里的每一个大块必须是同一个领域或类似概念的，每一个框都有意义

在这个问题上，这位同学做的还是很好的，但也还是有些小问题，比如UI层里的两个进程。这两个框显得意义不明，在没有描述的情况下，至少我是不明白他想表达的意思，而实际在沟通过程中，他也觉得这里挺奇怪的。

第三步，画完回顾一下是否描述清楚了第一步里的核心逻辑

很多时候我们一气呵成画了一张大图，结果一不小心容易画成一张流程图，把怎么写代码的思路也画到图上了。这就会导致图上有些地方是模块划分，而有些地方则是细节流程，整体就很失调。这只能通过反复的回顾和思考，进行自我调整了。最后，我给出当时模拟面试时，对于这个业务的粗略设想：



知道原理有什么用？(技术如何赋能)

有了大图，我们终于可以开始实现亮点了……吗？

现实很残酷，哪怕我们想出了一个饼，并不代表我们能吃到嘴里，从图变成面饼，我们需要太多的中间步骤。

而摆在技术人面前的问题是：如果有面粉了，你会揉面吗？你揉面的技术能保证烤出来的饼好吃吗？

知其然，而后使其然

我认为这就是为什么我们要了解原理。曾经有一位模拟面试的同学，在最后互问互答的时候问了我一个问题，怎么看待面试造火箭，平时拧螺丝？我觉得有点冤枉，因为一面大部分问的都是怎么拧螺丝，以及螺丝的型号，二面开始也就问问怎么造飞机，但是真的进入工作状态，阿里的场景里，至少在我们团队的场景里，我们就是在造火箭，只是造火箭的时候必须要拧拧螺丝，没螺丝你敢上？

有同学又不服了，我会拧螺丝，和我需要知道用什么螺丝有什么关系。那么上面那个烤饼，你能告诉我放白芝麻好吃还是放黑芝麻好吃吗？我相信大厨一定能回答上来，他甚至连小麦原产地都会和你掰扯一下。为什么到了同样需要匠心的编码领域，我们就不用关心用什么螺丝了呢？

当时我给这个同学举了个实际的例子：简历中有提到上传，那你不能当场告诉我，这个上传是服务端http接口配合你上传，还是用阿里云oss？用oss是服务端每次加签，还是用sts，还是前端直接加签？http上传你用什么contentType？用form表单组件提交还是自己通过xhr发送？如果需要登录鉴权怎么办？如果出现跨域问题怎么办？两种场景都有，都要实现，怎么封装组件？

什么？你说你要百度一下？你要百度一天？那我为什么不聘用那个不用百度的人呢？一天的工资算上5金这些成本，月薪20k来算，估计也得有小2000了，如果我把这2000增加到一个懂原理的大神手里，我们岂不是双赢，为什么要等你去搜索呢？只是个简单的上传文件功能，也就是页面里的一个豆腐块，这么小的螺丝，里面却有大大的学问。而日常工作中我们遇到类似的问题有非常多，具体可以参考我上一篇文章的解读，这里就不重复了。

任务的拆解

对于平时愿意学习的同学，到这一步可能开始陷入迷茫了，我之前也遇到过类似的困惑，那就是：要不要造轮子

我们经常会发现好像什么都能做，比如：你有的，我改改也能实现；社区有个差不多能用的，要不要直接用；好像大图上都有差不多的，那是不是拼拼凑凑就可以了，这个方案是不是没什么好做的了。

从我个人来说，每次画图我都会陷入这样的思考，还常常会钻牛角尖，为了整点差异化，故意换一些思路去做，这样能保证这个饼是我的。但最后我都会绕出来，这得益于上面画图的第三步，每次画完我都会重新回顾一下我真正想做的事情是什么。我认为这也是是否造轮子的一个评判标准：从业务的价值出发，思考真正核心的目标，并且为之努力，如果有现成的轮子，能满足业务核心诉求，那就放手去用。

首先，现实往往是这样的，当我们放手去用的时候，会发现这个轮子好像不那么好用，或者这个轮子没人维护了，又或者业务变化太快，轮子自己觉得顶不住了。机会自然会来到身边，而触发这些机会的，是我们不断的站在业务的视角去思考问题，业务的变化一定比一个平台化的轮子要来得快。

其次，真正核心的系统一定是紧贴业务，而且很难大范围复用的，好的技术架构在设计的时候，讲究的是够用即可，过度设计大部分就是没用的设计。在之后的迭代中，会随着业务的不断变化，被带着自我进化，那最终的产物也自然是和业务形态非常贴合。所以，我个人在选择的时候，一些核心的轮子，该造就造起来，但这些轮子一定是带有业务特色的，比如我会去造一个业务组件库，但是我绝不会去造一个antd。

最后，随着事物的演变，分久必合合久必分，单一业务用的好的系统一定是在更高的视角上抽象、整合的，在整个过程中，每个人的成长就会是我们想要的亮点了。或许在简历上你写下的是一个已经废弃的系统，但是它的灵魂在你心里，也存在于把他整合了的系统里，这种亮点在个人介绍的时候，一定是能侃侃而谈的。

从1到10能做什么？（思考方式的抽象）

终于，我们经历的各种抉择，投入了大量的时间，把一个亮点做出来了，完成了美好的从0到1，可有时候我们会发现的问题：从0到1看上去有很多要做的，做完了，从1到10还能做什么？

这个问题我个人也没有太多话语权，因为这两年总是在做从0到1的事情，甚至和我老板也聊过这个，总感觉自己没有个确定的事情。从0到1做一次挺爽的，一直做，不会一直爽，却只会让人觉得心慌，毕竟谁能保证永远能想出从0到1的事情呢？

而静下来反思之后，我发现事情并不是这么一刀切的，谁能说明白现在做的事情是0到1，还是1到10呢？这里的边界其实并没有那么明确，但抽象看，他们都是同一个套路

业务/技术思考 => 发现痛点 => 产出方案 => 拆解实现

伴随着这个闭环，业务永远在变化，而变化又会带来新的问题，只要保持一个思考的状态，没有必要区分具体再哪个阶段，因为你总能找到可以实现自我价值的地方，发现属于你的亮点。

• 我想学习后端，后端的学习路线是怎么样的呢？

- 作者：罗集
- 出品：淘系技术

初阶

对应岗位

开发工程师、高级开发工程师、技术专家

技术基础

网络：DNS、TCP&IP、浏览器、HTTP

操作系统

Unix/Linux命令行：grep, awk, sed, lsof, curl, wget, tail, head, less, find, ssh, kill；正则表达式

操作系统如何运行

进程管理

线程与并发

基本前端知识：

HTML、CSS、JavaScript

编程语言：

Java, Go, Kotlin、Bash、C/C++

算法与数据结构：

性能分析、基本数据结构（列表、树等）、排序、查找、图

数据库：

索引、事务、分库分表

编程工具

版本管理：

Git、版本控制系统、代码托管服务

构建工具：

Maven、Gradle

编辑器：

VSCode、Emacs、VIM

IDE：

IntelliJ IDEA

相关学习资源

1、陈鸣译，《计算机网络：自顶向下方法（原书第6版）》

2、龚奕利，贺莲译，《深入理解计算机系统（原书第3版）》

3、谢路云译，《算法（第4版）》

4、陈昊鹏译，《Java编程思想》

5、李道兵等译，《Go程序设计语言》

6、《ProgrammmingKotlin》

7、刘晓霞等译，《MySQL必知必会》

8、《Pro Git》

9、许晓斌，《Maven实战》

10、李松峰等译，《JavaScript高级程序设计（第3版）》

11、Cameron Newham等，《学习bash（第二版）》

中阶

对应岗位

高级开发工程师、技术专家

编程框架：

应用框架：spring、play网络框架：netty、持久化：MyBatis、HibernateRPC：gRPC、Dubbo

JVM执行系统：

类结构与字节码、类加载机制内存管理：内存区域与溢出、垃圾收集器与内存分配策略、性能监控与故障处理工具）并发：内存模型与线程、线程安全与锁优化

消息队列：

kafka、rocketmq

存储：

redis、mysql

工程测试：

单元测试、集成测试、功能测试安全：OAuth、HTTPS、CORS、内容安全策略持续集成

相关学习资源

- 1、Brian Goetz等，《Java并发编程实战》
- 2、张卫滨译，《Spring实战》
- 3、何品译，《Netty实战》
- 4、周志明，《深入理解Java虚拟机》
- 5、Neha Narkhede等，《Kafka: The Definitive Guide》
- 6、黄健宏，《Redis设计与实现》
- 7、《计算机程序的构造和解释》

高阶

对应岗位

技术专家

编程范式

函数式编程、面向对象编程、泛型编程、类型系统、装饰器模式、代理模式

设计原则：

DRY、KISS、S.O.L.I.D.

分布式系统

基础理论：节点与网络、时间与顺序、ACID、CAP/FLP/DLS、一致性理论

设计模式：网关模式，Gateway、边车模式，Sidecar、服务网格，Service Mesh、防腐层，Anti-corruption Layer、分布式锁、配置中心、异步通讯、幂等性

弹性（容错）设计：服务降级、服务限流、熔断设计、补偿事务、隔离设计、异步通讯、幂等性设计、重试设计、bulkhead

工程：监控追踪、服务/资源调度、流量控制、数据调度、开发与运维自动化、负载均衡+健康检查、服务发现+动态路由+健康检查、自动化运维

相关学习资源

- 1、赵军平等译，《数据密集型应用系统设计》
- 2、Marko Luksa，《Kubernetes in Action》
- 3、Distributed systems theory for the distributed systems engineer
- 4、孙宇聪译，《SRE：Google运维解密》
- 5、崔力强等译，《微服务设计》

第三部分

技术人必备工具书籍

- 前端篇
- 后端篇
- 客户端篇
- 服务端篇
- 测试篇
- 多媒体技术篇
- 计算机视觉篇
- 数据挖掘与数据分析篇
- 推荐系统篇
- 3D&AR篇

前端篇

1. 《JavaScript高级程序设计》（ [美] 尼古拉斯 ）
2. 《图解HTTP》（ [日] 上野宣 ）
3. 《React 进阶之路》（ 徐超 ）
4. 《设计模式》（ [美] Erich Gamma ）
5. 《代码整洁之道》（ [美] Robert C. Martin ）
6. 《CSS世界》（ 张鑫旭 ）
7. 《JavaScript设计模式与开发实践》（ 曾探 ）
8. 《JavaScript函数式编程指南》（ [美] 路易斯·阿泰西奥 ）
9. 《深入浅出Node.js》（ 朴灵 ）
10. 《精通CSS：高级Web标准解决方案》（ [英] Andy Budd ）
11. 《CSS权威指南》（ [美] Eric A.Meyer ）

后端篇

1. 《码出高效java开发手册》（ 杨冠宝 ）
2. 《深入理解Java虚拟机》（ 周志明 ）
3. 《数学之美》（ 吴军 ）
4. 《Redis实战》（ [美] Josiah Carlson、黄健宏译 ）
5. 《Java编程思想》（ [美] Bruce Eckel、陈昊鹏译 ）
6. 《重构》（ [美] Martin Fowler ）
7. 《硝烟中的Scrum和XP》（ [瑞典] 克里伯格、李剑译 ）
8. 《Head First 设计模式 》（ [美] 弗里曼 ）
9. 《Effective Java 》（ [美] Joshua Bloch 、俞黎敏译 ）

客户端篇

1. 《app架构师实践指南》（ SkySeraph ）
2. 《移动app性能优化与评测》（ TMQ 专项测试团队 ）
3. 《移动app测试实战》（ 邱鹏 / 陈吉 / 潘晓明 ）
4. 《Effective Objective-C 2.0 编写高质量iOS与OS X代码的52个有效方法》（ [英] Matt Galloway ）
5. 《深入解析Mac OS X & iOS操作系统》（ [美] 莱文 ）
6. 《重构：改善既有代码的设计》（ [美] 福勒 ）
7. 《Kotlin核心编程》（ 水滴技术团队 ）
8. 《JavaScript高级程序设计》（ [美] 尼古拉斯·泽卡斯 ）
9. 《C++ primer plus 》（ [美] Stephen Prata ）

服务端篇

1. 《程序员修炼之道》（ [美] 大卫·托马斯 ）
2. 《领域驱动设计》（ [美] Eric Evans ）
3. 《微服务架构设计》（ [美] 克里斯·理查森 ）
4. 《Kubernetes权威指南》（ 龚正 ）
5. 《阿里云云原生架构实践》（ 阿里集团阿里云智能事业群云原生应用平台 ）
6. 《Effective Java 中文版》（ 俞黎敏 ）
7. 《架构整洁之道》（ [美] Robert C. Martin ）
8. 《Java并发编程实战》（ [美] Brian Goetz ）
9. 《金字塔原理：思考、表达和解决问题的逻辑》（ [美] 芭芭拉·明托 ）

测试篇

1. 《自动化测试最佳实践》（ [美] Dorothy Graham / Mark Fewster ）
2. 《代码整洁之道》（ [美] Robert C. Martin ）
3. 《架构整洁之道》（ [美] Robert C. Martin ）
4. 《Google软件测试之道》（ 黄利/李中杰/薛明 ）
5. 《Effective Java 中文版》（ 俞黎敏 ）
6. 《Google SRE运维解密》（ [美] Betsy Beyer ）
7. 《深入理解java虚拟机》（ 周志明 ）
8. 《编程珠玑》（ [美] Jon Bentley ）
9. 《测试工程师全栈技术进阶与实践》（ 茹炳晟 ）

多媒体技术篇

1. High Efficiency Video Coding (HEVC) Algorithms and Architectures
2. Audio Signal Processing and Coding, Andreas Spanias
3. Fundamentals of Music Processing, Meinard Müller
4. An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics, Alexander Lerch
5. 《OpenGL编程指南》（[美] 施莱尔）
6. 《数字图像处理》（[美] 冈萨雷斯）
7. 《音视频开发进阶指南》（展晓凯 / 魏晓红）
8. 《新一代高效视频编码H.265/HEVC：原理、标准与实现》（万帅，杨付正）
9. 《FFmpeg从入门到精通》（刘歧 / 赵文杰）
10. 《WebRTC Native 开发实战》（许建林）
11. 《数字音频原理及应用 第3版》（卢官明 宗昉）
12. 《数字视频编码技术原理》（高文）

计算机视觉篇

1. pattern recognition and machine learning , Christopher Bishop
2. Machine Learning: A Probabilistic Perspective , Murphy
3. Deep Learning, Bengio 4. GANs in Action, Jakub Langr and Vladimir Bok
5. 《数字图像处理》第三版 ([美] 冈萨雷斯)
6. 《unix环境高级编程》第三版 ([美] 史蒂文斯)
7. 《计算机视觉——算法与应用》 ([美] Szelisk)
8. 《深度学习推荐系统》 (王喆)
9. 《机器学习》 (周志华)
10. 《统计学习方法》 (李航)
11. 《深度学习轻松学》 (冯超)
12. 《动手学深度学习》 (李沐)

数据挖掘与数据分析篇

1. 《数据挖掘导论》（ [美] Pang-Ning Tan ）
2. 《机器学习》（ 周志华 ）
3. 《深入浅出数据分析》（ [美] Michael Milton ）

推荐系统篇

1. Deep Learning, Ian Goodfellow
2. 《深度学习推荐系统》（王喆）
3. 《推荐系统实践》（项亮）
4. 《从零开始构建企业级推荐系统》（张相於）
5. 《机器学习》（周志华）

3D&AR篇

1. 《游戏引擎架构》（[美] Jason Gregory）
2. 《More Effective C++》（[美] Scott Meyers）
3. 《计算几何：算法与应用》（译者：邓俊辉）
4. Physically Based Rendering: From Theory to Implementation, Matt Pharr, Wenzel Jakob, Greg Humphreys
5. Fundamentals Of Computer Graphics , Peter Shirley
6. Real-Time Rendering, Fourth Edition, Tomas Akenine-Möller, Eric Haines, Naty Hoffman
7. Deep Learning with Python, François Chollet

第四部分

GitHub上的实用练手项目

- 推荐21个GitHub上好用又有趣的移动端项目（涵盖初中高阶）
- 推荐26个GitHub上好玩又有挑战的前端项目（涵盖初中高阶）
- 推荐10个GitHub上适合练手的后端项目（涵盖初中高阶）

• 推荐21个GitHub上好用又有趣的移动端项目（涵盖初中高阶）

- 作者：淘系技术
- 出品：淘系技术

初级篇

R.swift

一个工具库，编译时会自动识别项目中的资源，将项目中的资源类型化，消除字面量。目前支持图片、字体、颜色、LocalizedStrings、Storyboards等等。缺点是组件化开发时可能需要修改脚本。

```
imageView.image = UIImage(named: "SettingsIcon")
```

github地址：<https://github.com/mac-cain13/R.swift>

SwiftLint

一个工具库，编译时自动检查项目中不规范的Swift代码，消除不规范代码必备。可以通过编辑.swiftlint.yml来自定义规则。也支持通过brew安装，自己手动检查

```
1 let someForceCast = NSObject() as! Int // Force Cast Violation: Force casts should be avoided. (force_cast)
2 let colonOnWrongSide :Int = 0 // Colon Violation: Colons should be next to the identifier when specifying a type. (colon)
3 // SwiftLint is syntax-aware
4 // NSNumber() as! Int => no error
5 "let colonOnWrongSide :Int = 0" // => no error
```

github地址：<https://github.com/realm/SwiftLint>

LSUnusedResources

一个开源Mac工具App，可以分析检测项目中没用到的图片等资源。想要精简App包大小时，可以使用该工具辅助。

github地址：<https://github.com/tinymind/LSUnusedResources>

淘系工程师个人项目

这是一个模仿网易云特效的demo，在这么demo中，可以学习到酷炫的动画效果，也可以学习到贝塞尔曲线的来龙去脉。

项目地址：https://github.com/LiuLei0571/jingyun_breizer

Pedometer

swagger-ui一个简单轻量级的计步器，使用硬件传感器来计算步数，同时考虑到了电量使用问题，最小化电量消耗。

项目地址：<https://github.com/j4velin/Pedometer>

Lottie

Airbnb 开发的一款能够为原生应用添加动画效果的开源工具。Lottie 目前提供了 iOS, Android, 和 React Native 版本，能够实时渲染 After Effects 动画特效。

Lottie 在不需要对代码进行重写的情况下让工程师更加方便的创建更丰富的动画效果。有了 Lottie 你就不再需要使用 Gif 动画来展现效果。

目前，Lottie支持路径修剪，蒙版、遮盖等操作。此外还有一个可选的缓存机制，对那些频繁使用的东西能够更快加载。这款应用的目标就是帮助开发者和动画师能够更轻松的为应用创建动画，从而在整体上提升互动因素。



项目地址：<https://github.com/j4velin/Pedometer>

初级篇

CollectionKit

一个Swift列表库，继承自UIScrollView，可以自行对布局、数据源、渲染进行定制扩展。能快速自由搭建好看的列表页面，比UICollectionView的使用简单，比UITableView扩展性更强。



github地址: <https://github.com/SoySauceLab/CollectionKit>

swift-style-guide

swift-style-guide 是一个关于 Swift 语法的仓库。该仓库非常详细地介绍大多数常用 Swift 语法的推格式、注释等。对于 Swift 初学者来说是个非常有帮助的仓库。作者也维护了自己的网站，里面有非常多的高质量进阶的文章以及书籍，链接在图片下方。

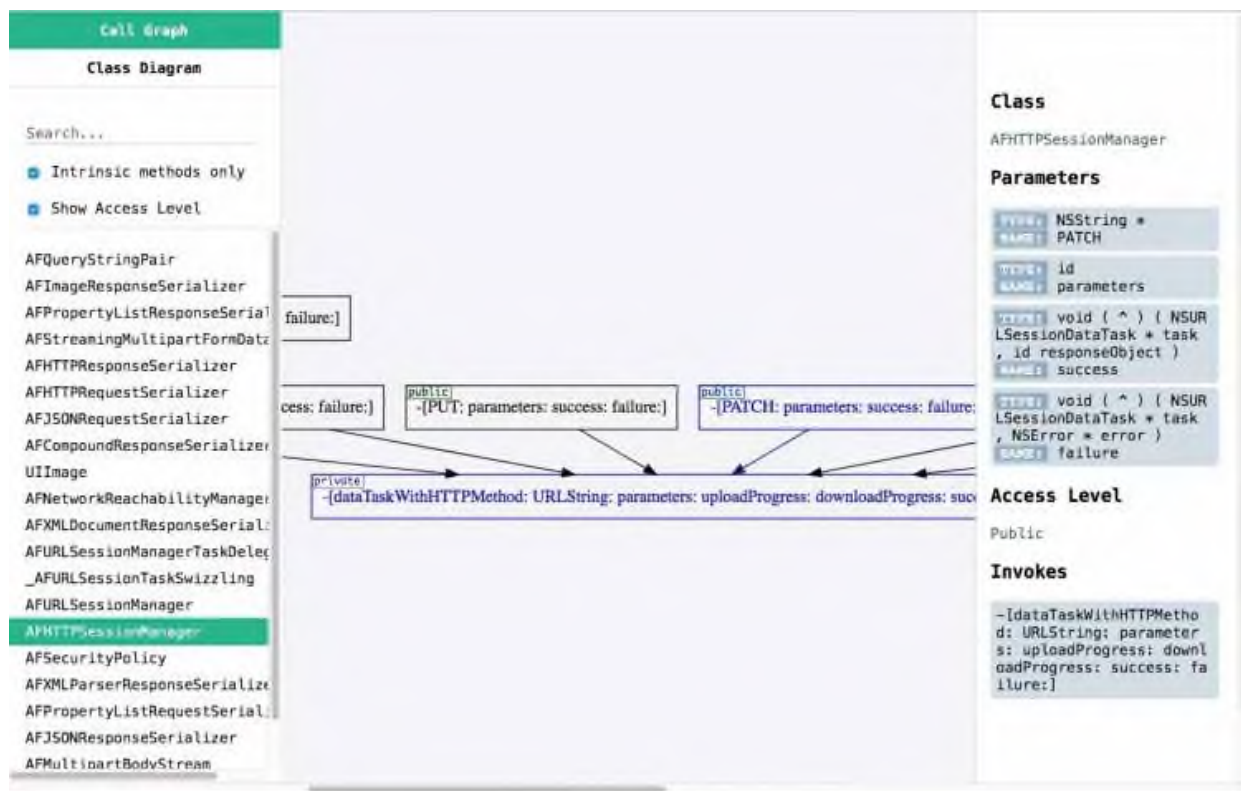


WebPage地址: <https://www.raywenderlich.com/whats-new>

GitHub地址: <https://github.com/raywenderlich/swift-style-guide>

Drafter

Drafter是一个命令行工具，用于分析iOS工程的代码，支持Objective-C和Swift。自动解析代码并生成方法调用关系图。自动解析代码并生成类继承关系图。



github地址: <https://github.com/L-Zephyr/Drafter>

高仿短视频推荐

目前短视频 app 火爆异常，没几年已经成为国民级 app，是时候来学习视频播放的相关知识了，下面两个项目是模仿抖音 app 的开源项目，可以作为开发者学习短视频开发的入门 demo。

iOS: douyin-ios-objectc

项目地址: <https://github.com/sshiqiao/douyin-ios-objectc>

Android: Titok

项目地址: <https://github.com/running-libo/Tiktok>

iOS demo预览地址: <https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpf-cp/efb935983b343e4b71384119d4009df~tplv-k3u1fbpfcp-watermark.awebp>

Android demo预览地址: <https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a54df25c4f104f839c6d702401e54d0b~tplv-k3u1fbpfcp-watermark.awebp>

Google breakpad

一个跨平台的崩溃转储和分析框架和工具集合。Breakpad可以在移除编译器调试信息后，抓取、压缩minidump 信息，将其发送回你的服务器，然后为 C/C++ 生成调用栈。

项目地址: <https://github.com/google/breakpad>

老司机周报

老司机技术周报是一个由移动开发爱好者组成的团体，是目前为数不多还在活跃的 iOS 社区组织。每周一定期发周报，内容不局限于具体的技术相关，也包括对于 iOS 开发者有价值的信息，比如软件工程、程序员话题等。除此之外，老司机周报每年还会组织 iOS 开发者对 Apple WWDC 上的 Session 进行翻译、分析和总结，并发表在小专栏上。



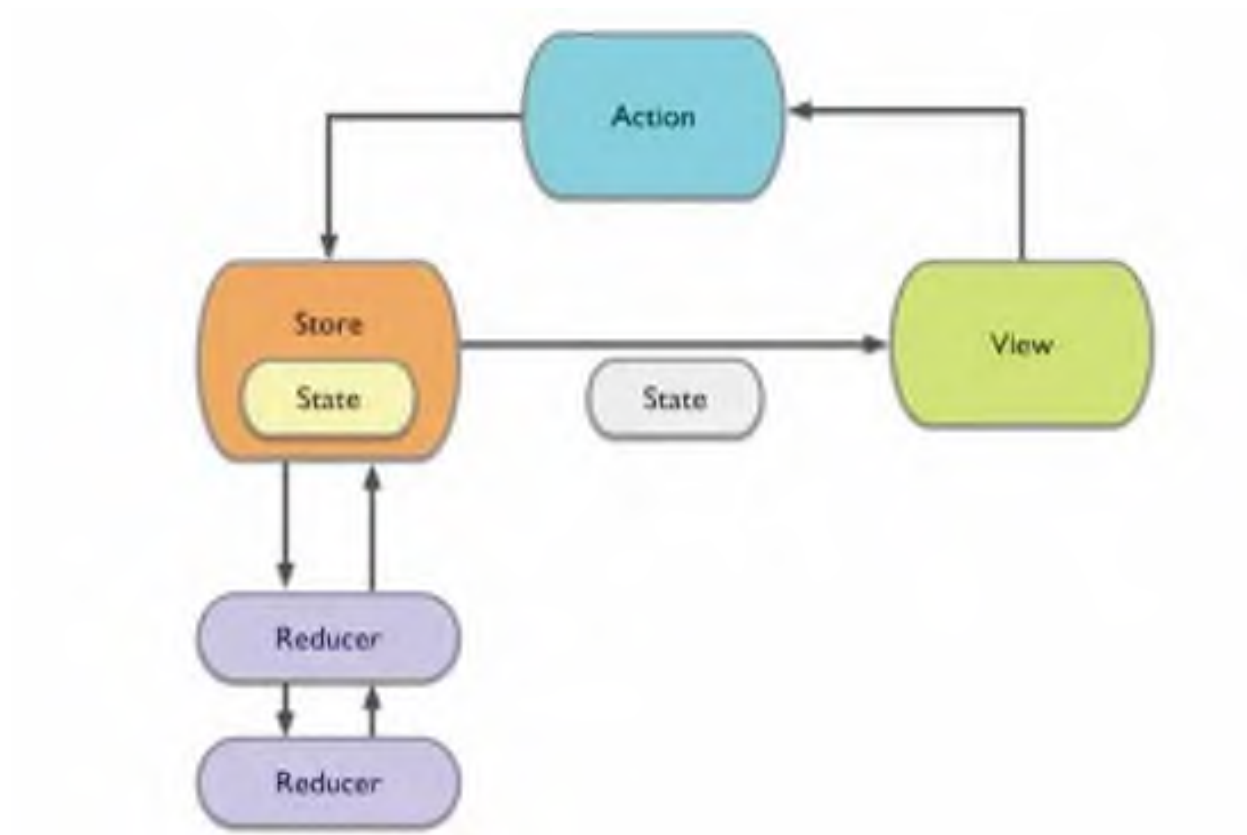
小专栏: <https://xiaozhuanlan.com/u/damonwong>

GitHub地址: <https://github.com/SwiftOldDriver/iOS-Weekly>

高级篇

ReSwift

Swift版本的 单向数据流框架。了解单向数据流思想可以参照Redux。



github地址: <https://github.com/ReSwift/ReSwift>

RxSwift

Swift版本的响应式编程框架，与RxJava作者相同，思想一致。Rx已经形成了完备的生态，基于Rx的网络库、列表、UI等扩展都可以在github找到。



github地址: <https://github.com/ReactiveX/RxSwift>

Texture

异步渲染UI框架，该库的设计思想对很多UI框架产生了影响。虽然入门门槛不高，但是想要理解内部实现机制还是有一定门槛的，而且由于用的人不多，这个库有一些坑比较难跨过去。不建议直接用在生产环境，某些特定场景还是可以用的。对此库建议观摩学习为主，封装动态化框架想要异步渲染，很大程度可以借鉴此库。



github地址: <https://github.com/TextureGroup/Texture>

yoga

一个C++写的跨平台布局引擎，ReactNative和Texture底层都用的该库。也可以基于此库快速实现一个Flex布局框架。



github地址: <https://github.com/facebook/yoga>

VirtualAPK

VirtualAPK是滴滴出行自研的一款优秀的插件化框架，通过将业务模块插件化，可随时更新插件来发布新功能，具备版本随时发布的能力。



项目地址: <https://github.com/didi/VirtualAPK>

wanAndroid

玩Android网站——努力做一个优质的 Android 站点，目前是国内比较火的Android开源社区，并且一直在持续不断的优化当中，作者已经开发了相关的 API 。

项目地址: <https://www.wanandroid.com/index>

项目地址: <https://github.com/hongyangAndroid/wa->

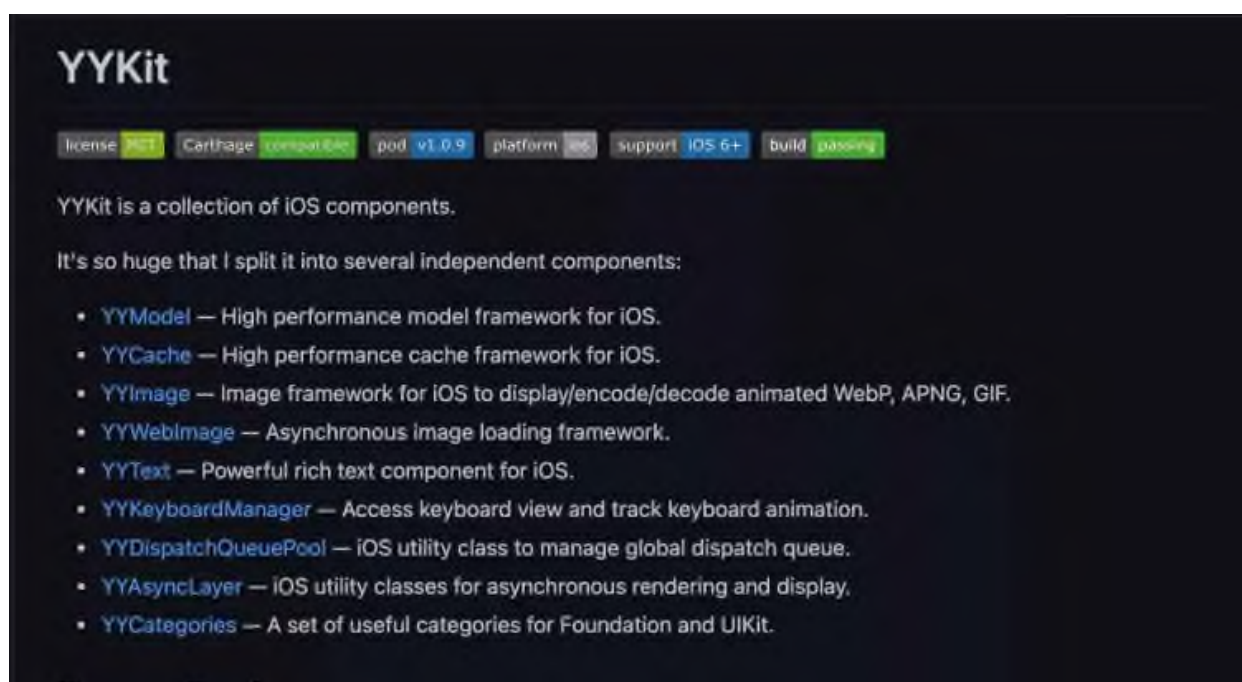
Epic

是一个在虚拟机层面、以 Java Method 为粒度的 运行时 AOP Hook 框架。简单来说，Epic 就是 ART 上的 Dexposed（支持 Android 5.0 ~ 11）。它可以拦截本进程内部几乎任意的 Java 方法调用，可用于实现 AOP 编程、运行时插桩、性能分析、安全审计等。

项目地址: <https://github.com/tiann/epic>

YYKit

YYKit 是 ibireme 开源的一组功能丰富的 iOS 组件，其中包括 序列化框架 YYModel、缓存框架 YYCache、富文本框架 YYText 等。虽然这是一个相对较老的 OC 库，但其代码质量、性能至今仍是非常高的水平。除了了解这些库的设计思路 and 性能分析，也可以阅读源码和对应的博客，深入分析原理和业界方案。



Blog地址: <https://blog.ibireme.com/>

GitHub地址: <https://github.com/ibireme/YYKit>

• 推荐26个GitHub上好玩又有挑战的前端项目（涵盖初中高阶）

- 作者：观浪、理火、车铭
- 出品：淘系技术

初级篇

这里一般适用于刚入门前端的同学，一般入门前端都是从html和css开始的，应该让这些同学提起对前端的兴趣，因此初级篇选用几个比较有趣的css库来让同学入门前端。

animate

animate是一个css动画库，动画内容一般也是我们所熟悉的，例如渐入、渐出、弹跳、旋转、抖动等特效。ppt里面的许多特效在animate都能找到。而且使用简单，只需引入库并添加相关css即可。



演示地址：<https://animate.style/>

GitHub地址：<https://github.com/animate-css/animate.css>

CSS-Inspiration

CSS-Inspiration涵盖了css的许多常见的特效，例如布局方式、border、伪元素、滤镜、背景3d等。这些都是css里面十分重要的知识点，当我们学习完css基础之后，可以利用此项目来制作一些常用的特效，也可以巩固基础知识。



演示地址：<https://chokcoco.github.io/CSS-Inspiration/#/>

GitHub地址：<https://github.com/chokcoco/CSS-Inspiration>

Bootstrap

Bootstrap是Twitter推出的一个用于前端开发的开源工具包。它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap是可以直接应用到我们平常的项目里的，而且非常方便、功能特别多，也是我们常用的前端UI框架。



官网: <https://getbootstrap.com/>

GitHub地址: <https://github.com/twbs/bootstrap>

tesseract.js

一个js ocr识别库, 支持包括中英文等许多语言的图片和视频文字识别, 底层封装了Tesseract OCR引擎来实现。



视频实时ocr:



demo: <http://tesseract.projectnaptha.com>

github: <https://github.com/naptha/tesseract.js>

vanillawebprojects

这里一共包含20+个项目, 包括表单验证、音视频播放器、打字等小应用, 有趣且方便学习。

20+ Web Projects With Vanilla JavaScript

This is the main repository for all of the projects in the course.

- [Course Link](#)
- [Course Info Website](#)

#	Project	Live Demo
01	Form Validator	Live Demo
02	Movie Seat Booking	Live Demo
03	Custom Video Player	Live Demo
04	Exchange Rate Calculator	Live Demo
05	DOM Array Methods Project	Live Demo
06	Menu Slider & Modal	Live Demo
07	Hangman Game	Live Demo
08	Mealfinder App	Live Demo
09	Expense Tracker	Live Demo
10	Music Player	Live Demo
11	Infinite Scrolling	Live Demo
12	Typing Game	Live Demo
13	Speech Text Reader	Live Demo
14	Memory Cards	Live Demo
15	LyricsSearch App	Live Demo
16	Relaxer App	Live Demo
17	Breakout Game	Live Demo
18	New Year Countdown	Live Demo
19	Sortable List	Live Demo
20	Speak Number Guessing Game	Live Demo

github: <https://github.com/bradtraversy/vanillawebprojects>

Ant Design

基于 Ant Design 设计体系的 React UI 组件库, 提供了丰富的组件, 基本满足各种开发使用需求。大部分前端项目都会涉及到React / Vue的组件开发, 如何开发封装一个组件是成为一个合格前端的第一步, 可以通过模仿 Ant Design的写法去学习如何开发一个组件, 同时学到框架(React / Vue) + HTML + CSS。

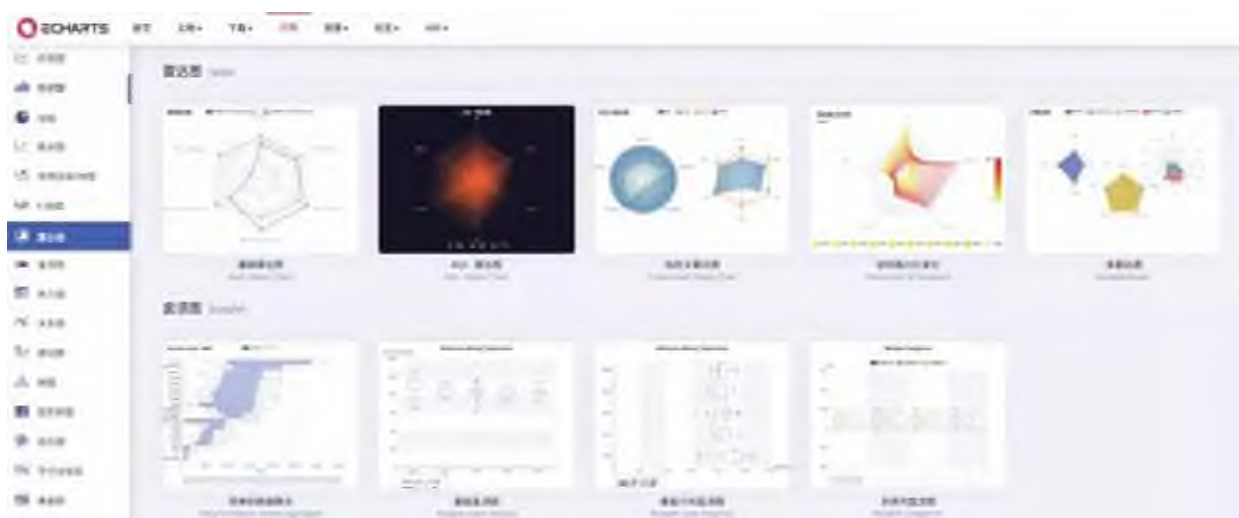


github: <https://github.com/ant-design/ant-design/>（React版本）

<https://github.com/vueComponent/ant-design-vue/>（Vue版本）

Echarts

基于 JavaScript 的开源可视化图表库。Echarts使用数据配置的方式轻松的实现可视化，如果想学习数据可视化，不如从学习Echarts的实现开始。



github: <https://github.com/apache/echarts>

JavaScript的学习

ECMAScript是JavaScript的语言标准，描述了JS的能力，API，基本实现步骤。目前比较通用的标准是ES6，从最初的版本：June 1997，最新的标准已经到了Edition 12 (June 2021)。每个版本都添加 / 修改了新的语法和API，作为一个前端开发者，应该关注JS的变化，可以查看最新的ECMAScript提案，即将发生的变化。

传送门: <https://es6.ruanyifeng.com/> ES6的学习

ECMAScript提案: <https://github.com/tc39/proposals>

中级篇

particles.js

当我们浏览网页的时候，会看见一些网站的背景有粒子效果，看完是不是也很好奇它是怎么实现的呢？在particles.js这个库里面，我们可以制造许多粒子效果，用在网页的背景可是相当好看的。



演示地址: <https://codepen.io/VincentGarreau/pen/pnlso>

GitHub地址: <https://github.com/VincentGarreau/particles.js>

reveal.js

reveal.js是一款网页ppt，界面优美，又支持许多ppt功能，同时也有许多插件满足前端的需求，例如markdown、code等代码高亮插件。



文档地址: <https://revealjs.com>

GitHub地址: <https://github.com/hakimel/reveal.js>

swiper.js

swiper是目前应用广泛的轮播图插件，拥有许多配置，也有许多切换效果，基本可以满足前端开发中轮播图的需求。



文档地址: <https://www.swiper.com.cn/api/index.html>

GitHub地址: <https://github.com/nolimits4web/swiper>

构建 (Build) 工具

大型项目需要构建工具进行打包，使用插件，转义等，所以学习如何配置构建工具，如何基于构建工具的能力自定义实现某些功能是很重要的。Webpack和Gulp是两种常见的打包工具。



Webpack: <https://github.com/webpack/webpack>

Gulp: <https://github.com/gulpjs/gulp>

TypeScript

现在的前端开发中，对于TypeScript的使用越来越多，尤其是大型迭代项目中，JavaScript不能满足需要，学习TypeScript会是一个必备的技能。

github: <https://github.com/microsoft/TypeScript>

github中也有不少翻译的guide book: <https://github.com/jkchao/typescript-book-chinese>

Yeoman

Yeoman是一个能快速生成一个Web APP的脚手架工具，可以通过Yeoman定义常用的模板APP。

github: <https://github.com/yeoman/yeoman>

rrweb

一个可以录制并回放任意 web 界面中的用户操作的纯前端的库，完全借助浏览器api来实现录制回放，实现的想法很有趣，对需要录制回放用户行为的需求很有帮助。

github: <https://github.com/rrweb-io/rrweb>

vite

是一个新型前端构建工具，能够显著提升前端开发体验，和webpack等老牌打包工具不同，vite另辟蹊径，开发环境下使用原生esm，让本地开发更快速。代码相较webpack更轻量，且实现逻辑巧妙，比较适合阅读学习。

github: <https://github.com/vitejs/vite>

UnblockNeteaseMusic

还在因为云音乐因为版权下架了一批你自己喜欢的歌曲，但是又不想单独为了几首歌换平台而烦恼吗，那你可以试试这个库，可以使用QQ / 虾米 / 百度 / 酷狗 / 酷我 / 咪咕 / JOOX 等音源把云音乐灰色歌曲解锁播放。



github: <https://github.com/nondanee/UnblockNeteaseMusic>

高级篇

vue-admin-template

这是一个vue后台管理系统框架，可以让我们轻松地实现一个后台管理系统，界面也比较美观。



文档地址: <https://panjiachen.github.io/vue-element-admin-site/zh/guide/>

GitHub地址: <https://github.com/PanJiaChen/vue-admin-template>

h5-dooring

h5-dooring是一款可视化编辑器，底层是用react写的，使用此工具可以让我们快速生成h5页面。同时通过这个平台也能了解到低代码的相关知识。



文档地址: <http://h5.dooring.cn/doc>

GitHub地址: <https://github.com/MrXujiang/h5-Dooring>

canvas-special

如果大家对图形学感兴趣,可以学一下canvas, canvas-special提供了许多canvas精美的案例供我们学习。



GitHub地址: <https://github.com/bxm0927/canvas-special>

node-question-answering

基于Node.js的用深度学习模型去回答问题

github: <https://github.com/huggingface/node-question-answering>

相关学习资源imgCook

通过设计 生成组件代码的低code工具



github: <https://github.com/imgcook/imgcook>

Midway

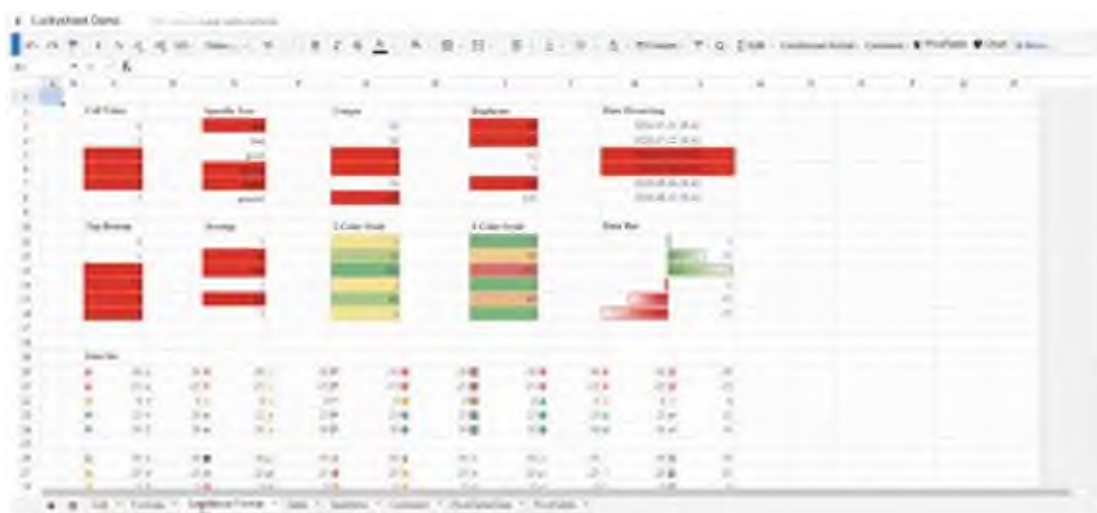
Midway 是一个适用于构建 Serverless 服务，传统应用、微服务，小程序后端的 Node.js 框架



github: <https://github.com/midwayjs/midway>

Luckysheet

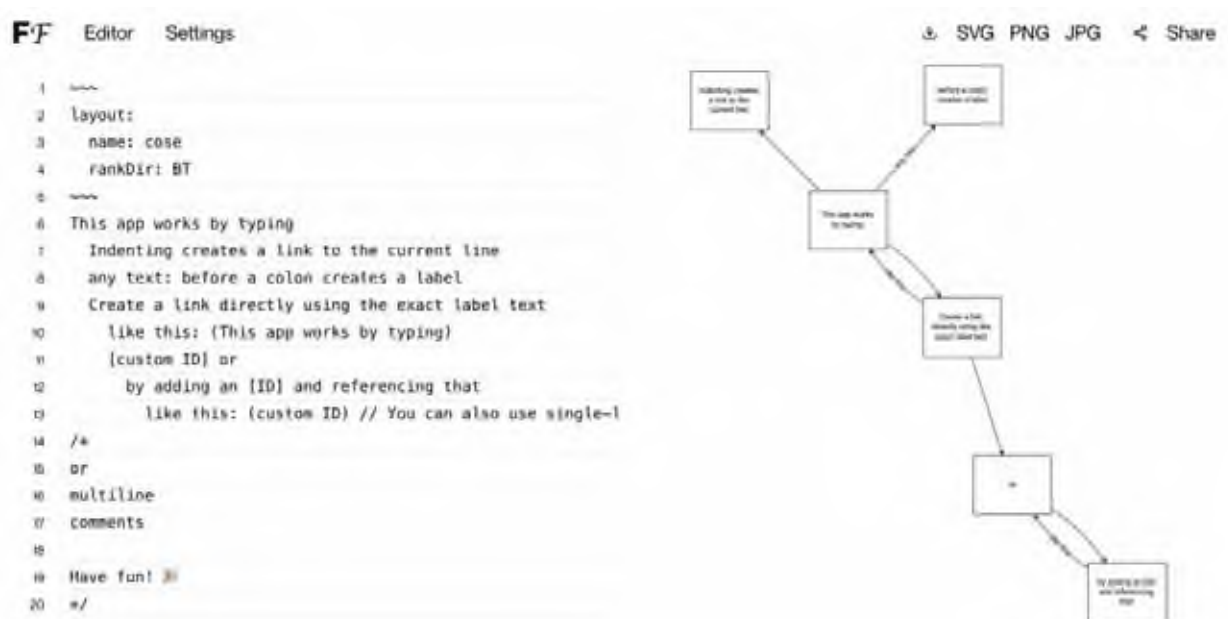
一款纯js实现的在线表格库，功能强大，excel的常见能力都支持，对有表格库集成需求或者要实现类似功能的很有帮助。



github: <https://github.com/mengshukeji/Luckysheet>

flowchart-fun

是一个高效的绘制流程图和思维导图的工具库，输入文字，就能自动生成一个框图，很有意思。不过是基于既定的英文单词串生成框图，可能上手使用会有一定难度。



github: <https://github.com/tone-row/flowchart-fun>

• 推荐10个GitHub上适合练手的后端项目（涵盖初中高阶）

- 作者：京波、空聘
- 出品：淘系技术

初级篇

guava

guava是一个包含各种集合类型及工具类的java库，被誉为java类库中的瑞士军刀，提供多种java标准库之外的复杂集合、map、图等高级数据结构，并包含非常多常用的工具类，例如缓存、IO、哈希、并发、string工具类等。



<https://github.com/google/guava>

guice

guice是一个轻量级的java依赖注入（DI）框架。开发中大型项目时基本上都离不开依赖注入框架的使用，相比spring的庞大复杂，guice是很好的DI入门框架，纯粹的DI能力及少量的代码能够让你快速使用和理解依赖注入的原理。

The logo for Guice, featuring the word "Guice" in a stylized font where each letter is a different color: 'G' is blue, 'u' is green, 'i' is yellow, 'c' is red, and 'e' is blue.

<https://github.com/google/guice>

feign

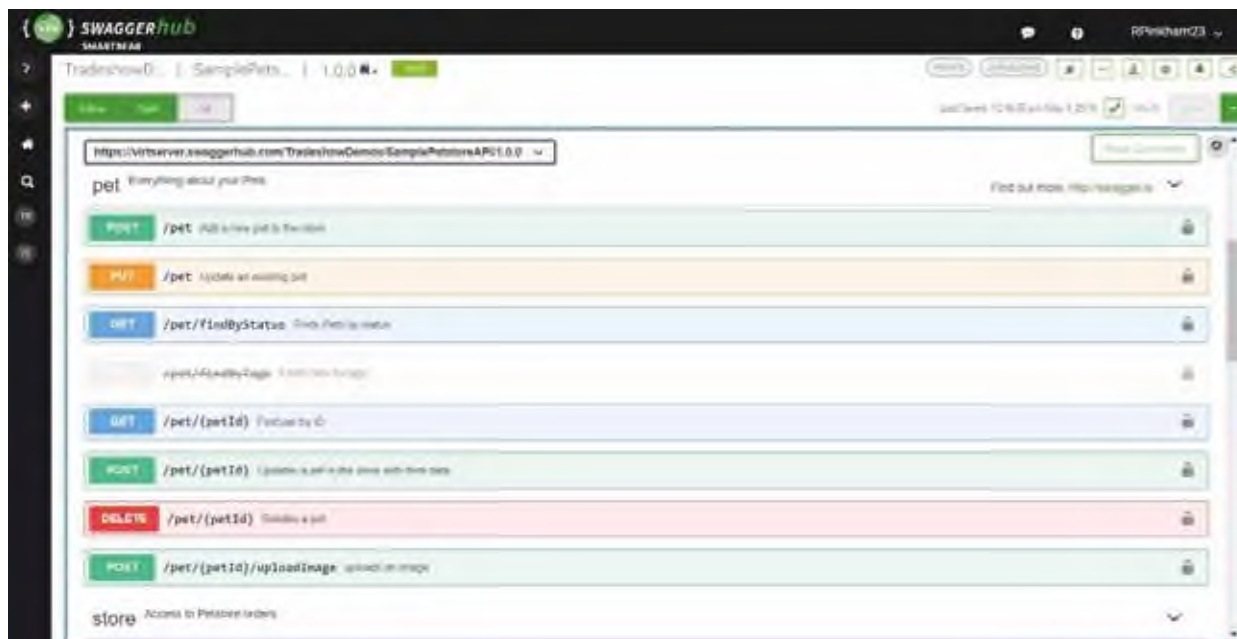
feign是一个java HTTP客户端绑定器，使用它能通过类似spring mvc注解方式快速创建restful API接口，大大简化了调用远程http接口时所需工作量。



<https://github.com/OpenFeign/feign>

swagger-ui

swagger-ui是一个可视化API展现工具，它根据代码里的OpenAPI规范自动生成带有可视化文档的可交互API界面，便于后端实现和客户端使用。

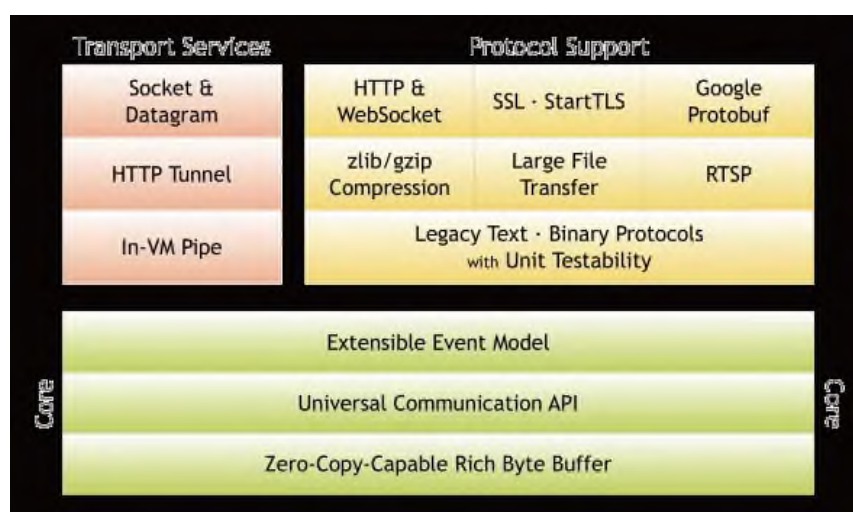


<https://github.com/swagger-api/swagger-ui>

中级篇

netty

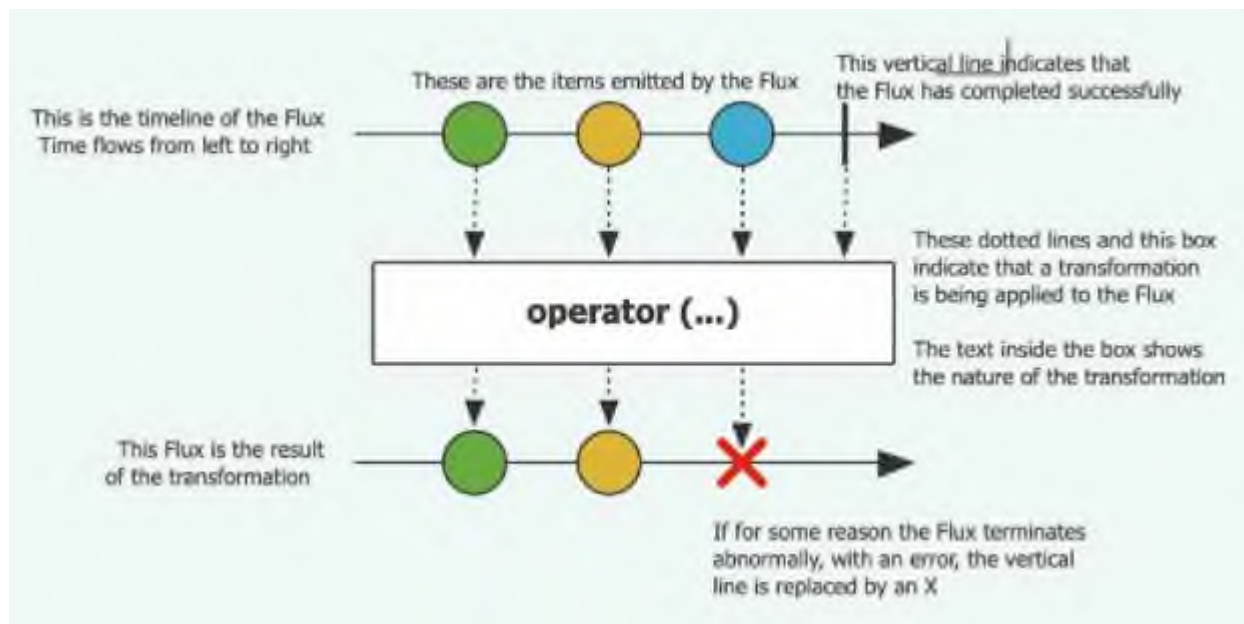
Netty 是一个异步事件驱动的网络应用程序框架，用于快速开发可维护的高性能协议服务器和客户端。



<https://github.com/netty/netty>

reactor

reactor是一个反应式编程（Reactive Streams）的一个具体实现，它使用一种优雅的易于阅读理解的流式编程方式提供java异步编程模型。



<https://github.com/reactor/reactor-core>

arthas

arthas是开源的java诊断工具，当你的线上JAVA服务出现问题而又无法远程debug时，此时可通过arthas查看运行中的java进程状态、监测程序处理过程等而无需修改代码或者重启服务。



<https://github.com/alibaba/arthas>

高级篇

zookeeper

zookeeper是一个java编写的分布式协调框架，它基于Paxos一致性算法，为分布式系统（例如hadoop、hbase、jstorm等）提供配置、消息队列、分布式锁、命名服务等能力，是分布式系统的“大脑”。



<https://github.com/apache/zookeeper>

flink

flink是一个开源的流处理框架，具有强大的流处理和批处理能力。通过它你可使用简单的java代码甚至sql完成一系列复杂的实时计算或离线计算功能。



<https://github.com/apache/flink>

Kubernetes

Kubernetes 是用于自动部署，扩展和管理容器化应用程序的开源系统。源自Google 15 年生产环境的运维经验，同时凝聚了社区的最佳创意和实践，这个项目偏底层，说不上好玩，但是比较硬核。



官网: <https://kubernetes.io/>

github: <https://github.com/kubernetes/kubernetes>

第五部分

2021淘系A类顶会论文精选

- 数据挖掘与机器学习领域
- 机器学习、计算机视觉及多媒体领域
- 交叉/综合/新兴领域
- 数据库、知识管理、信息检索领域

Large-scale Fake Click Detection for E-commerce Recommendation Systems

Jingdong Li^{*1}, Zhao Li^{*2}, Jiaming Huang², Ji Zhang³, Xiaoling Wang¹, Xingjian Lu¹, and Jingren Zhou²

¹Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
Email: jdl@stu.ecnu.edu.cn, xlwang@sei.ecnu.edu.cn, xjlu@cs.ecnu.edu.cn

²Alibaba Group, China
Email: {lizhao.lz, jimmy.hjm, jingren.zhou}@alibaba-inc.com

³Zhejiang Lab, Zhejiang, China
Email: zhangji77@gmail.com

Abstract—With the development of e-commerce platforms, e-commerce recommendation systems are playing an increasingly important role for the purpose of product recommendation. As a new attack model against e-commerce recommendation systems, the “Ride Item’s Coattails” attack creates fake click information to establish the deceptive correlation between popular products and low-quality products in order to mislead the recommendation system of e-commerce platform to boost the sales of low-quality products. This attack is characterized by high concealment and strong destructiveness, which can cause great damage to e-commerce recommendation systems, and adversely affect the usability of the e-commerce platform and users’ shopping experience. It is therefore of great practical significance to study how to quickly and effectively identify the false click information and the corresponding “Ride Item’s Coattails” attack to better safeguard e-commerce recommendation systems. At present, there is no previously reported relevant research work conducted specifically for addressing the detection of the “Ride Item’s Coattails” attack. In this work, we carried out pioneering work in analyzing and summarizing the characteristics of the false click information produced by attackers on the target products in the “Ride Item’s Coattails” attack and designed a set of attack detection techniques suitable for e-commerce recommendation systems. Experimental results on real e-commerce datasets show that our proposed techniques can quickly and effectively detect the large-scale fake click information as well as the associated “Ride Item’s Coattails” attack in e-commerce recommendation systems.

Index Terms—E-commerce Recommendations, Ride Item’s Coattails Attack, Attack Detection

I. INTRODUCTION

Online shopping websites, such as TaoBao and Amazon, have become popular platforms for people to find and purchase products. Compared with traditional shopping, e-commerce provides customers with the opportunity of browsing voluminous product catalogs, comparing prices, and creating their wish lists with great convenience, which at the same time generates a large amount of user behavior data. By collecting and analyzing these user behavior data, e-commerce recommendation systems can provide customers with a better service based on their interests [1]. Specifically, in an item-to-user recommendation scenario, once the user clicks an item A ,

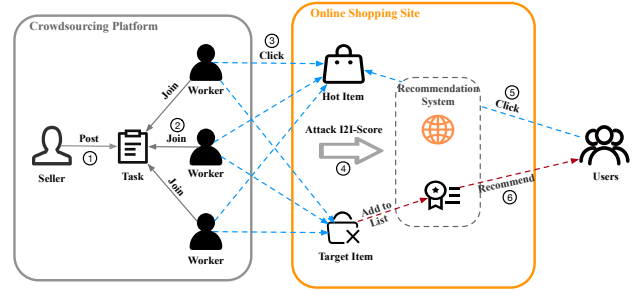


Figure 1. An Example of “Ride Item’s Coattails” Attack

recommendation systems will figure out other items that are “similar” to A , then recommend them to the user. In TaoBao’s recommendation system, item-to-item relevance score (abbreviated as I2I-score) is used to describe and measure the similar relationship between items. It is impacted by not only the similarity of the items’ attributes but also the probability of the two items appearing in users’ click lists at the same time. Of course, recommendation systems will consider other factors for a more comprehensive judgment [2], but the I2I-score turns out to be the most valuable one.

Such I2I recommendation strategy allows users to select highly relevant products together conveniently (e.g., beer and diapers). However, some malicious online sellers exploit it to increase the exposure of their low-quality products by producing a large-scale of fake clicks to affect the evaluation of the I2I-score. More specifically, they maliciously forge common clicks between the low-quality items and hot items to raise the I2I-score between them. So when normal users are browsing through these hot items, the target low-quality items can also have a higher probability of occurrence in the resulted item recommendations list, thereby increasing the exposure of these items. What’s more, malicious online sellers always try to associate multiple hot items with target items to maximize the exposure opportunities. We name this kind of attack as the “**Ride Item’s Coattails**” attack.

The “Ride Item’s Coattails” attack can cause serious negative impacts on the I2I-score leading to higher I2I-score be-

^{*}, These authors contributed to this work equally

tween hot items and target low-quality items, thereby reducing the recommendation accuracy of the online shopping sites' recommendation system and misleading normal users to buy low-quality or even fake items from those malicious online sellers. This not only damages the user's shopping experience, but also affects the reputation of the e-commerce platform. Therefore, in this paper, we aim to study how to effectively detect the large-scale fake click information as well as the newly-emerged "Ride Item's Coattails" attack described above in e-commerce recommendation systems for minimizing its negative impact on normal users and online shopping sites.

To increase their items' exposure, some online sellers manipulate the ranking of the items in the recommendation list with the help of crowd workers (As shown in Fig. 1). First, a malicious online seller from an online shopping site posts a task describing the target items, expectations, and remunerations on the crowdsourcing platform. Then, some experienced workers accept this task and launch a large-scale "Ride Item's Coattails" attack on the online shopping site. After the attack, target items gain higher probabilities of appearing at the top of the item recommendation list.

These attacks conducted by crowd workers instead of sellers bring many challenges: (1) Compared with sellers, crowd workers are less relevant to the target item, so it is hard to detect them directly; (2) Crowd workers with some hot items click records by themselves are very similar to normal ones and difficult to be detected even with manual efforts; (3) Experienced workers will follow some particularly designed guidelines to disguise themselves as normal users [1], such as clicking on other irrelevant items or increasing the number of clicks on hot items; (4) With the increasing popularity of deceptive items, some normal users may also be attracted by them and contribute clicks. In other words, both attackers and target items largely behave like normal records.

To tackle these challenges, we first extract a large-scale user-item bipartite graph from Taobao and investigate the corresponding user and item behavior that are highly likely to be participants in a "Ride Item's Coattails" attack. Then, we analyze the characteristics of the "Ride Item's Coattails" attack from the perspective of users and items. By integrating structural features and correlations (both user-based and item-based), we design a discriminative framework to detect the "Ride Item's Coattails" attack. Through experimental comparisons with competitive baselines, we show that our framework is efficient and effective.

We summarize the major contributions of this paper as follows:

- We discovered and defined a special attack mode against e-commerce recommendation systems and named it the "Ride Item's Coattails" attack. To our best knowledge, ours is the first reported work to investigate this type of attack;
- We provide a comprehensive analysis of the "Ride Item's Coattails" attack from the perspective of users and items;
- We propose a novel detection framework that can effectively detect the large-scale fake click information as

well as the associated "Ride Item's Coattails" attack in e-commerce recommendation systems;

- We have conducted experiments on a large-scale real-world e-commerce dataset. The experimental results show the effectiveness and efficiency of our attack detection framework and algorithms.

II. RELATED WORKS

The detection of the "Ride Items Coattails" attack is related to the research works in the following areas: dense subgraph mining, false information detection, and community detection in bipartite graphs. However, they are not applicable to directly solve the detection problem of the "Ride Items Coattails" attack.

A. Dense Subgraph Mining Algorithms

The "Ride Item's Coattails" attack model determines that the subgraph formed by hot items, target items, and attackers must be a cohesive bipartite graph. The densest structure in a bipartite graph is called the complete bipartite graph (also known as *biclique*, as defined in Definition 1) which has been widely used to capture cohesive bipartite subgraphs in a broad spectrum of bipartite graph applications [3], [4].

Definition 1. Given a bipartite graph $G = (U, V, E)$, a *biclique* $C = (L \cup R)$ is a complete bipartite subgraph of G induced by a pair of two disjoint subsets $L \subseteq U$, $R \subseteq V$, such that $\forall u \in L, \forall v \in R, (u, v) \in E$.

However, enumerating maximal bicliques in a bipartite graph is #P-complete [5], and it is impossible to directly enumerate all the maximal bicliques in a large-scale e-commerce bipartite graph [6]. Moreover, the cohesive subgraph, composed of users, target items, and hot items, may not be a perfect biclique. The definition of quasi-biclique can tackle this problem [7], but as far as we know, the existing quasi-biclique mining algorithms [7], [8] all aim at solving the maximum (vertex) quasi-biclique search problem. This problem is proved to be NP-hard [9] and can only output one near biclique.

B. False Information Detection

The existing false information detection methods can broadly be categorized into three categories: propagation-modeling based, feature-based, and graph-based. Propagation-modeling algorithms use information spread models to identify false information [10], [11]. Feature-based algorithms leverage the unique characteristics for detection by using them as features in a machine learning model or rule-based framework [12], [13]. However, both of them require side information, such as true information or labeled data.

Graph-based algorithms detect attackers by identifying suspicious dense regions of the graph. Such methods are potentially harder to evade because creating fake clicks unavoidably generates edges in the graph. COPYCATCH [4] detects temporally bipartite cores that are ill-gotten "Likes" of Facebook, but without timestamps, it degenerates into enumerating all

the maximal bicliques. CATCHSYNC [14] looks for synchronized behavior in the graph without side information, which forms dense bipartite subgraphs. However, this method is not robust against experienced adversaries and lacks performance guarantees on the detection algorithm. FRAUDAR [15] proposes suspiciousness metrics to catch fraudulent blocks (dense subgraphs), which are robust to camouflage. However, without determining the number of blocks in advance, the algorithm can't find multiple attack groups.

C. Community Detection in Bipartite Graphs

Newman et al. [16] first propose the modularity metric to identify communities in general graphs. Guimera et al. [17] modify the definition given by Newman [16] and introduce a modularity measurement for bipartite graphs.

Raghavan et al. [18] propose a famous label propagation algorithms, which uses the network structure to guide its progress for finding communities in general graphs. In addition, some variants of community detection algorithms have been proposed to discover well-connected communities [19] and dense clusters [20]. Unfortunately, community detection algorithms often require a priori information such as density or number of clusters as input. This preliminary information is hard to acquire ahead from large-scale e-commerce networks, making the above algorithms unable to be directly applied to “Ride Item’s Coattails” attack detection.

III. PROBLEM DEFINITION

We assume that there are a set of m users $U = \{u_1, \dots, u_m\}$ and a set of n items $V = \{v_1, \dots, v_n\}$ connected according to a bipartite graph $G = (U \cup V, E)$, where $(u_i, v_j, p) \in E$ is user u_i has clicked item v_j p times. We now describe our attack model and then our problem definition.

A. Attack Model

We assume that crowd workers are hired to use accounts they control to add a large number of fake clicks/edges between their account and items, inducing a dense subgraph. This general characteristic of the “Ride Item’s Coattails” attack was found to be true in a real Taobao dataset. Many other papers use dense blocks to detect fraud [4], [14], [15], [21]. However, different from the “who-Likes-what” graph on Facebook [4], each user can only “Like” a page once. In the “Ride Item’s Coattails” attack, an account can click an item multiple times, making the size of the attack group smaller and more challenging to be detected. Also, the clicked items include not only target items but also hot items and a small part of random normal items. Therefore, it is unreasonable to use a non-discriminatory metric [4], [15] to distinguish suspicious nodes.

In real e-commerce recommendation systems, there are always adversarial challenges. Experienced crowd workers will add arbitrary “camouflage” to disguise their fraud, i.e., edges pointing from their user accounts to random normal items. This work assumes that attackers have complete knowledge of how the recommendation system works and the attack

detection mechanisms. Under such a strict attack model, the false click information created by the attacker is more purposeful and brings more significant challenges to our detection approach. In Section IV, we conducted a deeper analysis of the strategies that the attacker might adopt.

B. Desired Properties of Detection Approach

Our goal is to detect suspicious nodes typically indicative of groups of users and items participating in “Ride Item’s Coattails” attacks, and the problem is defined as:

Given a bipartite graph $G = (U \cup V, E)$ of clicks between users U and items V , we aim to find a set of suspicious users U_{sus} and a set of suspicious target items V_{sus} that are involved in a set of “Ride Item’s Coattails” attack groups $g = \{g_1, g_2, \dots, g_n\}$ in the graph G , $U_{sus} = \{g_1.u_{sus} \cup g_2.u_{sus} \dots \cup g_n.u_{sus}\}$ and $V_{sus} = \{g_1.v_{sus} \cup g_2.v_{sus} \dots \cup g_n.v_{sus}\}$.

The detection approach should satisfy the following properties:

- (1) *Effectiveness and efficiency*: It accurately detects groups of “Ride Item’s Coattails” attack with suspicious behaviors and is applicable to large e-commerce graphs;
- (2) *Side information oblivious*: Neither the labeled node nor the attribute information of the node is required. It only defines suspicious behavior using a topological structure;
- (3) *Camouflage restriction*: It can restrict the maximum number of false clicks/edges (i.e., an upper bound) that attackers can add without being detected;
- (4) *Easy of use for end-users*: a) It should support scoring for each abnormal node, convenient for business experts to select the top- k nodes for analysis and punishment; b) It can explicitly limit the detected group’s size to avoid the misjudgment of group-buying phenomenon.

IV. “RIDE ITEM’S COATTAILS” ATTACK ANALYSIS

To have a more comprehensive understanding of the “Ride Item’s Coattails” attack, we first extract a large-scale user-item bipartite graph from Taobao’s historical data within the past year, as the basis for analyzing the underlying data characteristics of the attack. Without loss of generality, we conduct stratified sampling on various items to generate a representative bipartite graph. Then, we store it in a table (named TaoBao_UI_Clicks) with three columns: *User_ID*, *Item_ID* and *Click*. Specifically, one record (1, 1, 3) in TaoBao_UI_Clicks means that user with $id = 1$ has clicked the item with $id = 1$ three times.

Table I
DATA SCALE OF TAOBAO_UI_CLICKS

User	Item	Edge	Total_click
20M	4M	90M	200M

Table II
DATA STATISTICS OF TAOBAO_UI_CLICKS

	Avg_clk	Avg_cnt	Stddev
User	11.35	4.32	33.34
Item	54.94	20.49	992.78

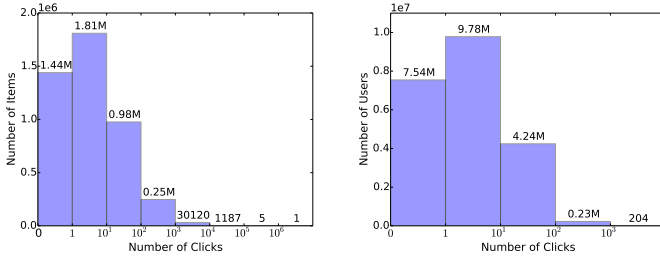


Figure 2. Distribution of Clicks

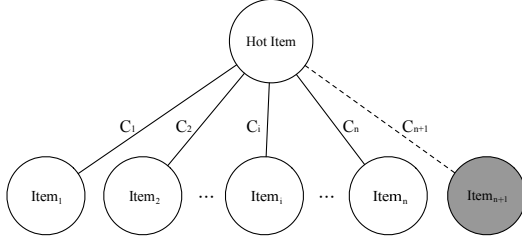


Figure 3. I2I-score Calculation Model

The scale and statistics information of the dataset are shown in Table I and II respectively. The *Edge* represents the number of click records in the dataset, and *Total_click* represents the sum of *Click*. The statistics data include the average number of clicks (*Avg_clk*), the average number of item clicks (*Avg_cnt*), and the standard deviation of the total number of clicks (*Stdev*). Moreover, the distributions of the clicks are shown in Fig. 2a and Fig. 2b. Based on this real-world dataset, we conduct a comparative analysis of the “Ride Item’s Coattails” attack which covers two aspects: user behavior and item behavior. A summary of the behavior of “Ride Item’s Coattails” attack is presented at the end of this section.

A. User Behavior

The reason why the “Ride Item’s Coattails” attack can work is that, by clicking on hot items and target items, crowd workers can increase the I2I-scores between hot items and target items. Then, normal users who have clicked on these hot items will be misled by the e-commerce recommendation system to browse low-quality target items.

To give a clear explanation of this process, we introduce the calculation model of the I2I-score as shown in Fig. 3. Assuming that there are co-click records of a hot item and an ordinary item sets $\{item_1, item_2, \dots, item_n\}$. The C_i between the *hot item* and an ordinary item $item_i$ represents the number of clicks on item $item_i$ under the condition that the *hot item* has been clicked before. So the I2I-score S_i between the hot item and the ordinary item $item_i$ can be calculated as follows:

$$S_i = \frac{C_i}{C_1 + C_2 + \dots + C_n} \quad (1)$$

Next, we give three basic assumptions on the behavior of crowd workers (abnormal users) in a “Ride Item’s Coattails” attack:

Assumption 1. Crowd workers hope to get the remuneration as much as possible without being detected by the e-commerce risk control system;

Assumption 2. Crowd workers have some specifically designed strategies to realize Assumption 1;

Assumption 3. All crowd workers follow similar strategies (as mentioned in Assumption 2).

Based on the above I2I-score model and three assumptions, we focus on analyzing the strategies that crowd workers will adopt to discover some distinctive anomalous patterns.

Suppose that an online seller posts a “Ride Item’s Coattails” attack request and the target item assigned by he/she is $item_{n+1}$. Then, the crowd worker who has accepted this request will click the hot item and the target item $item_{n+1}$ to establish a link at first. At this time, C_{n+1} is initialized to 1.

Because of the purpose to study the click behavior of abnormal users and Assumption 3, we can ignore other user’s influence during this attack and the value of $C_1, C_2, \dots, C_n$ are fixed. For Assumption 1, crowd workers’ goal is to increase the I2I-scores between a target item and hot items as much as possible (because the I2I-score is positively correlated with the target item’s exposure and the compensation they can receive), while keeping their click behavior similar to normal users.

One of the simplest and effective strategies is to set a click budget for each attack because the risk control system can easily detect excessive clicks on a item from a user. Crowd workers need an effective click strategy to maximize the I2I-score with limited number of clicks. Assuming that the click budget assigned to this attack task is C_b , the number of additional clicks to maximize the I2I-score is C , the number of additional clicks on the target item C_{n+1} is C' and the number of additional clicks on other items is $C - C'$, we can get the I2I-score calculation function as follow:

$$S_{n+1} = \frac{C_{n+1} + C'}{C_1 + C_2 + \dots + C_n + (C_{n+1} + C') + C - C'} \quad (2)$$

where $0 \leq C' \leq C \leq C_b - 2$. For establishing a link between a hot item and a target item, two clicks have to be consumed.

Note that given constants m and n , $f(x) = \frac{m+x}{n+x}$ is a strictly monotone increasing function for $n \geq m > 0, x \geq 0$. So, $f(x) \leq f(C_b - 2)$ for $0 \leq x \leq C_b - 2$, and we can derive the following inequality where $C_1, C_2, \dots, C_{n+1}$ are constants:

$$\begin{aligned} S_{n+1} &= \frac{C_{n+1} + C'}{C_1 + C_2 + \dots + C_n + (C_{n+1} + C') + C - C'} \\ &\leq \frac{C_{n+1} + C}{C_1 + C_2 + \dots + C_n + C_{n+1} + C} \\ &\leq \frac{C_{n+1} + C_b - 2}{C_1 + C_2 + \dots + C_n + C_{n+1} + C_b - 2} \end{aligned} \quad (3)$$

S_{n+1} can get the maximum value iff $C' = C = C_b - 2$. Therefore, the best strategy for crowd workers is to click the hot item once and then click the target item as much as possible. In a realistic scenario, to counter the risk control system of the shopping sites, crowd workers will not only

click the hot item and the target item, but they will also click other uncorrelated items to disguise themselves. Next, based on the above behavior analysis results, we will look for users with such click patterns from the dataset.

In the first step, we classify items in the dataset into hot items and ordinary items. As shown in Fig. 2, the distribution of user clicks on the items exhibits a heavy-tailed distribution. It follows Pareto’s principle [22], which means that the clicks on a few hot items (about 20%) account for most of the total number of clicks (about 80%). We rank the items by the number of clicks and get the hot items set by summing clicks up to 80% of the total number of clicks. The threshold T_{hot} equals to the number of clicks of the last hot items which is 1,320 in the dataset. In other words, items with the number of clicks greater than or equal to 1,320 are hot items.

In the second step, based on the above analysis, we screen out the suspicious users who have clicked both hot and ordinary items at the same time and have clicked ordinary items more than a certain number of times from the dataset. In order to determine this threshold T_{click} , we assume that the average total number of clicks (Avg_clk in Table II) is the most reasonable total number of clicks that the crowd users will use to disguise themselves. The distribution of the crowd works’ clicks on the items also exhibits a heavy-tailed distribution, that is, the number of clicks on a few target items (about 20%) account for most of the total number of clicks (about 80%). So we can figure out the T_{click} by Equation 4,

$$T_{click} = \frac{Avg\_clk \times 80\%}{Avg\_cnt \times 20\%} \quad (4)$$

where $Avg\_clk = 11.35$ and $Avg\_cnt = 4.23$ in the table. Then, an ordinary item whose number of clicks greater than or equal to 12 is an abnormal click record.

Table III
PART OF THE CLICK RECORD OF A SUSPECT

ID	Click	Total_click	Hot
1	2	16,853	1
2	1	9,928	1
3	1	6,178	1
4	1	1,912	0
5	2	1,558	0
6	13	1,319	0
7	1	632	0
8	13	368	0

In this way, we roughly screen out more than 1.4 million users ($\geq 7\%$ of all users) with abnormal click records. Here we show a part of the click records of a representative and suspected user in Table III for analysis. We replace the real *Item_ID* with the sequence *ID* for data privacy, and the *Click* column represents how many times this user have clicked this item. The *Total_click* column represents the number of clicks from all users. If the total number of clicks of the item is higher than the threshold T_{hot} , then the value of *Hot* is 1, otherwise it is 0.

As shown in Table III, the suspicious user intentionally clicks hot items ($ID=1,2,3$) as few as possible, and frequently clicks the target items ($ID=6,8$). The suspicious user also clicks

some other items ($ID=4,5,7$) to confuse the risk control system. To increase the I2I-score efficiently, the number of clicks of such non-target items is relatively small. These click behaviors are consistent with the results of attack intention analysis.

Here we show a part of the click records of a normal user in Table IV as a comparison. It can be noticed that the normal user clicks hot items ($ID=1,2$) more frequently. For ordinary items ($ID=3,4,5,6$) the number of clicks is relatively small.

Table IV
PART OF THE CLICK RECORD OF AN ORDINARY USER

ID	Click	Total_click	Hot
1	19	9,206	1
2	4	3,816	1
3	1	1,013	1
4	1	974	0
5	1	894	0
6	1	632	0

In conclusion, we summarize the typical behavioral characteristics of the abnormal users involved in a “Ride Item’s Coattails” attack as follows (in order of significance):

- (1) They click ordinary items more frequently, and the number of clicks of some ordinary items (suspicious target items) is greater than a certain threshold T_{click} ;
- (2) The number of clicks of hot items is far less than that of ordinary items, and the average number of clicks of hot items is extremely small (< 4);
- (3) Because of the need to launch an attack and disguise themselves at the same time, it causes significant differences in their number of clicks on different ordinary items.

B. Item Behavior

A typical scenario of online sellers launching a “Ride Item’s Coattails” attack is that the target items cannot attract users’ clicks due to their low quality or poor design. Sellers hope to increase the exposure of their items in this way artificially. In this scenario, the total number of clicks on the target item and the number of users who have clicked it should be small at the beginning. After the “Ride Item’s Coattails” attack, the total number of clicks of the target items will increase. Because those additional clicks are almost from the crowd workers, the click behavior of the target items will show the characteristics of a high number of clicks from few users.

The suspicious items selected here come from the ordinary item sets that appear in the abnormal click records introduced in the previous section, and we get more than 600,000 suspicious items ($\geq 15\%$ of all items). In order to eliminate the influence of the existing “Ride Item’s Coattails” attack on normal items, we select the items that newly appear in item tables (from the business department) in the past month. Table V shows the statistics of a suspicious item with a total number of clicks of 368 and a normal item with a total number of clicks of 404.

It can be found that when the total number of clicks is not much different ($< 10\%$), the number of users (*User_num*) who have clicked normal items is twice as that of suspicious items. The mean (*Mean*) and standard deviation (*Stdev*) of

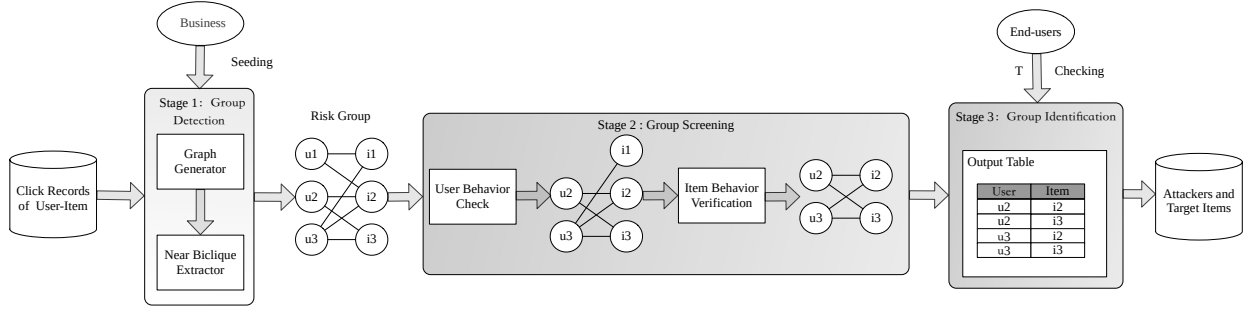


Figure 4. "Ride Item's Coattails" Attack Detection Framework

Table V
STATISTIC DATA OF SUSPICIOUS ITEM AND NORMAL ITEM

Total_click	Mean	Stdev	User_num	Max	Min
368	3.64	7.36	101	40	1
404	1.99	2.52	203	17	1

the number of clicks of normal items are lower than those of suspicious items as well. We also count the frequency that suspicious users appear in the click lists of the normal items and suspicious items, and the results are 0.49% for the normal items and 1.98% for the suspicious items, respectively.

In conclusion, we summarize the typical behavioral characteristics of the target items involved in a "Ride Item's Coattails" attack as follows (in order of significance):

- (1) When the total number of clicks are close, the number of users who have clicked the target item is far less than that of a normal item;
- (2) The proportion of abnormal users in the click list of a target item is much higher than that of a normal item.

C. Summary

From the above analysis results, it can conclude that crowd workers and target items have unique behavioral characteristics that are significantly different from those of normal users and items. These unique behavioral characteristics can be used as an effective additional filtering condition to help us detect crowd workers and target items. However, the screening techniques used in the above analysis are very rough and inaccurate. It leaves more than 7% of all users and 15% of all items in the results. In order to detect the "Ride Item's Coattails" attack in e-commerce recommendation systems quickly and effectively, we need more systematic approaches.

V. "RIDE ITEM'S COATTAILS" ATTACK DETECTION

In this section, based on the analysis results of the behavior of abnormal users and items, we first present a naive algorithm that can judge whether an item or user is abnormal independently. Then, we propose a novel "Ride Item's Coattails" Attack Detection framework (RICD) for systematically detecting the "Ride Item's Coattails" attack groups in e-commerce recommendation systems.

A. Naive Algorithm

The basic idea can be summarized as follows. If most of the users who click an ordinary item have clicked a large number

of hot items, it is very likely that this ordinary item is a target item and the users are suspicious users. The pseudocode of the algorithm is shown in Algorithm 1.

Algorithm 1 Naive Algorithm

Input: User List U , Item List L , Hot Item Threshold T_{hot} , Risk Threshold T_{risk} .

Output: Abnormal Item Set S .

```

1:  $NewItem, HotItem \leftarrow \emptyset$ ;
2: for each item  $l \in L$  do
3:   if  $l.total\_click < T_{hot}$  then
4:      $NewItem.append(l)$ ;
5:   else
6:      $HotItem.append(l)$ ;
7: for each user  $u \in U$  do
8:    $u.Alpha \leftarrow GETALPHA(u, HotItem)$ ;
9: for each item  $l \in L$  do
10:   $l.RiskScore \leftarrow \text{sum } Alpha \text{ of } l's \text{ neighbors}$ ;
11:  if  $l.RiskScore > T_{risk}$  then
12:     $S.append(l)$ ;
13: return  $S$ ;
14: function  $GETALPHA(u, HotItem)$ 
15:   $Alpha \leftarrow 0$ ;
16:  for each item  $l$  adjacent to  $u$  do
17:    if  $l \in HotItem$  then
18:       $Alpha \leftarrow Alpha + l.click$ ;
19:  return  $Alpha$ 

```

The input of the algorithm includes the user list U , item list L , hot item threshold T_{hot} , and risk threshold T_{risk} . Here, the risk threshold is used to balance the trade-off between precision and recall. The algorithm classifies items into hot items and new items in Line 2 to 6. New items are treated as potential target items in this algorithm. The algorithm invokes function $getAlpha$ to calculate the $Alpha$ for every user in Line 8. Then, we can get $riskscore$ of each item by summing their neighbors' $Alpha$ and append items with $riskscore > T_{risk}$ to abnormal item set S in Line 10 to 12. Utilizing the obtained abnormal item set S for classification, we can figure out abnormal users in the same way as Algorithm 1. The advantages of the naive algorithm is that it is very intuitive and can efficiently detect some suspicious users and items, but it has the following disadvantages: (1) The calculation of

riskscore is independent and does not consider the impact between items; (2) Thresholds are hard to set in advance. Therefore, we need a more systematic solution.

B. Detection Framework

The framework of RICD is shown in Fig. 4, which can be divided into the following three sequential modules from left to right:

(1) **Suspicious Group Detection Module.** This module aims to quickly locate suspicious “Ride Item’s Coattails” attack groups in large-scale e-commerce networks. In such networks, abnormal nodes only take up a tiny portion, so it is unreasonable to apply a filtering algorithm directly in original large-scale networks.

As we mentioned in Section IV, creating fake clicks between attackers and target items unavoidably generates edges, which will form dense subgraphs. So the intuitional idea is that we can regard the suspicious “Ride Item’s Coattails” attack group as a near biclique and design a near biclique extraction algorithm (Algorithm 2 Line 3) for the “Ride Item’s Coattails” attack detection. In Section V-C, we will introduce related definitions and algorithm (Line 3) in details.

Algorithm 2 Suspicious Group Detection Algorithm

Input: User-Item Click Table CT , Known Attacker $Seed$, Parameters $Para$.

Output: Suspicious Attack Group Sets B .

```

1:  $BiGraph \leftarrow \emptyset$ ; ▷ Store bipartite graph
2:  $BiGraph \leftarrow \text{GRAPHGENERATOR}(CT, Seed)$ ;
3:  $B \leftarrow \text{NearBicliqueExtract}(BiGraph, Para)$ ;
4: function  $\text{GRAPHGENERATOR}(CT, Seed)$ 
5:   if size of  $Seed$  is not null then
6:     for each node in  $Seed$  do
7:        $BiGraph.append(\text{MaxBiGraph}(node))$ ;
8:        $Seed.remove(node)$ ;
9:   else
10:     $BiGraph \leftarrow \text{TableToBiGraph}(CT)$ ;
11:   return  $BiGraph$ 

```

The input of this module includes the user-item click table, some known abnormal items or users provided by the business department, and the parameters used in the near biclique extraction algorithm. We store the intermediate results in the *Bigraph* in the form of a bipartite graph in Line 2 and use the near biclique extraction algorithm to detect suspicious groups in Line 3. It should be noted that in the suspicious group detection algorithm, the known *Seed* provided by the business department is only utilized as an auxiliary input in Lines 5 to 8. Its main purpose is to help prune the input bipartite graph and locate the suspicious groups quickly. So in an actual attack detection scenario, even if there are no known abnormal items or users, this module can still work properly.

(2) **Suspicious Group Screening Module.** The main function of this module is to further screen the output suspicious groups by the previous module based on the relevant conclusions of the “Ride Item’s Coattails” attack behavior analysis in

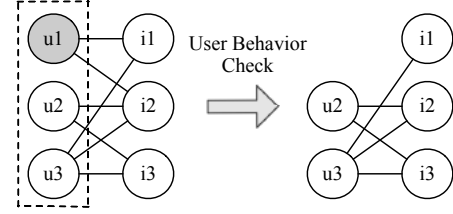


Figure 5. Example of User Behavior Check

Section IV and output more accurate attack groups. We divide the group screen process into the following two steps: user behavior check and item behavior verification.

We first introduce the user behavior check step. For the suspicious group $G = \{u_1, u_2, u_3, i_1, i_2, i_3\}$ produced by the suspicious group detection module and their click records $R = \{C_1^1, C_1^2, C_2^1, C_2^2, C_3^1, C_3^2, C_3^3\}$ (C_i^j representing the number of times user u_i clicked on the item i_j), shown in Fig. 5. We check the corresponding click list of each user in the user list $U = \{u_1, u_2, u_3\}$. For example, for suspicious user u_1 , its click list is $I_1 = \{i_1, i_2\}$. We will check whether they are hot items (clicks greater than T_{hot}). The same method is applied to u_2 and u_3 . Supposed that we find I_1 is a hot item, while I_2 and I_3 are not, which we have $C_2^2, C_3^2 > T_{click} > C_1^2$. Then, items I_2 and I_3 clicked by two users u_2 and u_3 at the same time are more likely to be involved in a “Ride Item’s Coattails” attack, because neither I_1 or I_2 is a hot item and C_1^1 and C_1^2 are so small. Then, we can remove u_1 from G and get a more accurate attack group $G' = \{u_2, u_3, i_1, i_2, i_3\}$. However, i_1 cannot be removed from G at this time, because it is still possible that I_1 and I_2 launch a “Ride Item’s Coattails” attack together.

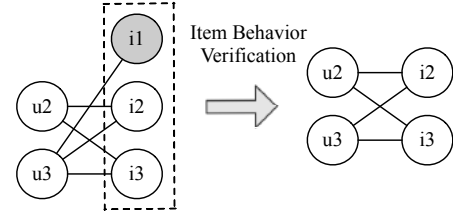


Figure 6. Example of Item Behavior Verification

Next, we will conduct the item behavior verification step, as shown in Fig. 6. We will check the coincidence degree of the clicked user sets of $I = \{i_1, i_2, i_3\}$ in turn. The clicked sets of i_2 and i_3 are exactly the same. If $C_3^2 \gg C_3^1$ is also met at the same time, then the link between u_3 and I_1 is likely to be a disguise, which means that the item I_1 is an ordinary item and can be removed from the attack group G' . We finally get the output suspicious attack group $G'' = \{u_2, u_3, i_2, i_3\}$.

(3) **Suspicious Group Identification Module.** The main function of this module is to convert the output suspicious attack groups into the form of a user-item table ordered by the degree of suspicion and make the framework easy to use for end-users (e.g., business experts and downstream applications). To achieve these goals, we propose two optimization strategies:

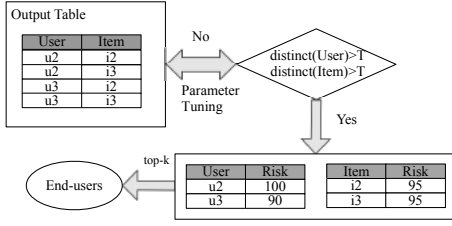


Figure 7. Business-oriented Group Identification Strategy

Ranking strategy based on the risk score. The core idea of this strategy is to efficiently assign a risk score to each user and item in the user-item table returned. The user's risk score refers to the number of suspicious items clicked by the user. The item's risk score can be obtained by calculating the average risk score of the clicked users. In this way, we can sort users and items according to their risk scores. Then, end-users can conveniently select the top- k users and items to punish and amend the manipulated I2I-score.

Parameter adjustment strategy based on feedback. In extreme situations, the scale of suspicious groups output by the first two modules is far less than expected, which means that some parameters in the parameter space may not be properly specified. Because these parameters are highly interpretable, we can quickly adjust the parameters based on our experience (e.g., decrease T_{click}) to increase the recall of the result. This strategy can improve the scalability and robustness of the framework.

These two strategies can be coordinated, as shown in Fig. 7. Each time we will check whether the number of users and items in the output table meets the expectation T given by the end-users. If not, we will adjust the parameters of the suspicious group detection module and suspicious group screen module according to the feedback until the output result meets the expectation T . Then, with the help of the ranking strategy based on risk scores, the end-users can choose what to evaluate.

C. Near Biclique Extraction Algorithm

In Section II-A, we have pinpointed two main shortcomings of utilizing the biclique mining algorithms to detect the "Ride Item's Coattails" attack in large-scale e-commerce bipartite graphs as being (1) the high latency brought by MBE algorithm with a high complexity; (2) the exceedingly strict structural requirements of the biclique (Definition 1). In order to solve these two problems, we define a (k_1, k_2, α) -extension of biclique and design a (k_1, k_2, α) -extension biclique extraction algorithm to detect the "Ride Item's Coattails" attack. Here, we first give the definition of the α -extension of biclique [23].

Definition 2. Given a bipartite graph $G = (U, V, E)$, an α -extension of a biclique $C = (L \cup R)$ is an ordered pair (L', R') where $L' \subseteq (U - L)$ and $R' \subseteq (V - R)$ such that at least $\alpha \times 100\%$ of the nodes in each of the node sets L and R are connected through edges to all nodes in R' and L' , respectively.

By adjusting the parameter α properly, we can extract attack groups in various scales. This definition reflects the click behavior of the crowd workers as well as target items in reality and solves the problem caused by the strict definition of biclique. To improve computational efficiency and find dense subgraphs containing more fake click records, we introduce two parameters k_1 and k_2 to limit the minimum size of the results as given in Definition 3.

Definition 3. Given a bipartite graph $G = (U, V, E)$, an (α, k_1, k_2) -extension biclique is an ordered pair $(L \cup L', R \cup R')$ where (L, R) is a biclique and (L', R') its α -extension such that $|L| \geq k_1$, $|R| \geq k_2$ and at least $\alpha \times 100\%$ of the nodes in each of the node sets L and R are connected through edges to all nodes in R' and L' , respectively.

Algorithm 3 (k_1, k_2, α) -extension Biclique Extraction Algorithm

Input: User-Item Graph $BiGraph$, Parameters $Para$.

Output: Suspicious Attack Group Sets B .

```

1:  $BiGraph \leftarrow COREPRUNING(BiGraph, Para)$ ;
2:  $B \leftarrow SQUAREPRUNING(BiGraph, Para)$ ;
3: function  $COREPRUNING(BiGraph, Para)$ 
4:   for each user  $u \in BiGraph$  do
5:     if  $u.degree < \lceil Para.\alpha \times Para.k_2 \rceil$  then
6:        $BiGraph.remove(u)$ ;
7:   for each item  $i \in BiGraph$  do
8:     if  $i.degree < \lceil Para.\alpha \times Para.k_1 \rceil$  then
9:        $BiGraph.remove(i)$ ;
10:  return  $BiGraph$ 
11: function  $SQUAREPRUNING(BiGraph, Para)$ 
12:  for each user  $u_i \in BiGraph$  do
13:     $num \leftarrow 0$ ;
14:    for each user  $u_j \in BiGraph | u_i$  do
15:      if  $|u_i.adj \cap u_j.adj| \geq \lceil Para.k_2 \times Para.\alpha \rceil$ 
16:        then
17:           $num \leftarrow num + 1$ ;
18:      if  $num < k_1$  then  $\triangleright \alpha > 0$ 
19:         $BiGraph.remove(u_i)$ ;
20:  for each item  $l_i \in BiGraph$  do
21:     $num \leftarrow 0$ ;
22:    for each item  $l_j \in BiGraph | l_i$  do
23:      if  $|l_i.adj \cap l_j.adj| \geq \lceil Para.k_1 \times Para.\alpha \rceil$  then
24:         $num \leftarrow num + 1$ ;
25:    if  $num < k_2$  then
26:       $BiGraph.remove(l_i)$ ;
27:   $B \leftarrow$  the rest of items and users in  $BiGraph$ ;
28:  return  $BiGraph$ 

```

Lemma 1. Given an (α, k_1, k_2) -extension biclique $C = (U, V)$ composes of a biclique $C_b = (L, R)$ and its α -extension $C' = (L', R')$, where $U = L \cup L'$, $V = R \cup R'$ and $L \cap L' = R \cap R' = \emptyset$, we have:

(1) $\forall u \in U(C)$, $u.degree \geq \alpha \times k_2$;

(2) $\forall v \in V(C), v.degree \geq \alpha \times k_1$;

Definition 4. Given an (α, k_1, k_2) -extension biclique $C = (U, V)$ composes of a biclique $C_b = (L, R)$ and its α -extension $C' = (L', R')$, for any $u \in U(C)$ and $u' \in U(C)$, u' is a (α, k) -neighbor of u iff

$$|u.adj \cap u'.adj| \geq \lceil k \times \alpha \rceil$$

For any $u \in U(C)$, the set of (α, k) -neighbor of u can be defined as;

$$S_{\alpha, k}(u, C) = \{u', s.t. |u.adj \cap u'.adj| \geq \lceil k \times \alpha \rceil\}$$

Similarly, we can define $S_{\alpha, k}(v, C)$ for any $v \in V(C)$.

Lemma 2. Given an (α, k_1, k_2) -extension biclique $C = (U, V)$ composes of a biclique $C_b = (L, R)$ and its α -extension $C' = (L', R')$, where $U = L \cup L'$, $V = R \cup R'$, $L \cap L' = R \cap R' = \emptyset$, we have:

(1) $\forall u \in U(C), |S_{\alpha, k_2}(u, C)| \geq k_1$;

(2) $\forall v \in V(C), |S_{\alpha, k_1}(u, C)| \geq k_2$;

The pseudocode of the (α, k_1, k_2) -extension biclique extraction algorithm (Line 3 in Algorithm 1) is shown in Algorithm 3 in details. The input of the algorithm includes the bipartite graph *BiGraph* and parameters k_1, k_2 and α . Our algorithm contains two pruning strategies based on the input parameters in Line 1 to 2. The idea of *Core Pruning* is that we can directly eliminate vertices without sufficient degree (neighbors) in Line 4 to 9 based on Lemma 1. The *Square Pruning* function takes the number of the two-hop neighbors of the vertex with common adjacencies into consideration in Line 15 to 17 based on Lemma 2. We remove a vertex in *BiGraph* and all its adjacent edges from G . Theoretically, after performing these two pruning strategies, the remaining vertices should appear in specific (α, k_1, k_2) -extension bicliques. The selection order of candidate vertices will affect the number of intermediates, thereby affecting the computational efficiency of the algorithm [3]. We select candidates in a non-decreasing order based on the size of common two-hop neighborhood in Line 14 and 21, like *reduce2Hop* in [6].

Note that the reason why our framework can restrict camouflage is that each (α, k_1, k_2) -extension biclique extracted by Algorithm 3 must contains a biclique, if the attacker wants not to be detected by the algorithm, the new edges he adds can't create a new biclique. This problem is known as Zarankiewicz problem [24] and Füredi [25] provides the best general upper bound. In other words, for every attacker who is not detected by RICD, the false clicks he can create have an upper bound.

D. Complexity Analysis

Given a bipartite graph $G = (U, V, E)$, obviously the time complexity of the naive algorithm is $O(|U||V|)$. The time complexity of the (k_1, k_2, α) -extension biclique extraction algorithm is dominated by the *CorePruning* procedure and the *SquarePruning* procedure. Because the *remove* operation (in Line 6 and 9) not only needs to remove a node but all its adjacent edges, the complexity of the *CorePruning*

procedure is $O(|U| + |V| + |E|)$. The time complexity of the *SquarePruning* procedure is dominated by the operations from Line 14 to 16 (from Line 21 to 23) and the complexity of the intersection operations (in Line 15 and 22) is $O(|V|)$ or $O(|U|)$. So the time complexity of the *SquarePruning* procedure is $O(|U|^2|V| + |V|^2|U|)$. The total complexity of the (k_1, k_2, α) -extension biclique extraction algorithm is $O((|U| + |V|)(|V||U| + 1) + |E|)$.

VI. EXPERIMENTS

A. Experimental Setup

Dataset. The real-world e-commerce user-item click data used in our experiments is the large-scale TaoBao_UI_ table (introduced in Section IV), and its statistical information is shown in Table I and Table II. To get the ground truth, we first sample 4,000 nodes from the results of the naive algorithm and ask business experts to label them as suspicious or normal. Then, we intersect these suspicious nodes with attackers already known in the dataset to produce a list of about 2,000 known abnormal nodes.

Platform. All experiments and most of the algorithms (except COPYCATCH and FRAUDAR) are conducted on the Grape [26] and MaxCompute [27]. Grape is a parallel graph engine for graph computations, and MaxCompute is a data processing platform widely used in Alibaba. The Grape has been integrated into MaxCompute so that we can use the graph mining algorithms offered by Grape API by inputting data and setting the number of workers (16 by default). Each worker represents a machine with 48 cores and 512G memory.

Baseline Methods. Since we conduct the first reported research on investigating the "Ride Item's Coattails" attack in online shopping, there is a lack of effective detection methods for this problem that we can directly compare with. Therefore, we compare our proposed detection framework RICD with three widely-used methods for community detection in many fields and two state-of-the-art graph-based anomaly detection algorithms. Meanwhile, to verify the effectiveness of our proposed techniques in the framework, we also add some simplified framework variants as our baseline methods. Details are given below:

- **Label Propagation Algorithm (LPA):** LPA is a fast algorithm for finding communities in a graph, and it does not require prior information about the communities. In the experiment, we use the LPA [18] implemented in Grape for community detection in bipartite graphs with the default input of maximal iteration round (max_round = 20) and every node is initialized with a unique label.
- **Common Neighbors Algorithm (CN):** CN algorithm is widely used to determine the closeness of a pair of nodes [28], which is similar to the idea of RICD. We use the implementation of the CN algorithm for bipartite graphs in Grape and adjust the parameter cn_threshold (minimum number of common neighbors) to 10 to make it consistent with the k_1, k_2 in RICD.
- **Louvain:** It is a heuristic method based on modularity optimization to extract the community structure of large

networks. In the experiment, we use the Louvain algorithm [29] implemented in Grape with the default input of tolerance and minimal progress threshold (tolerance = 2, min_progress = 1,000).

- **COPYCATCH:** This work is the most closely related to ours. Because there is no timestamp in the dataset, the algorithm degenerates to enumerate (near) biclique cores, which is a #P-hard problem. So we refer to the imbea [3] for the implementation and take the result of running the algorithm in a limited time (about 600 seconds) as the final output. ρ, m and n are consistent with the α, k_1 and k_2 in RICD.
- **FRAUDAR:** FRAUDAR is designed to catch fraudulent blocks in graph datasets in a camouflage-resistant way. We refer to the source code provided by the authors (<http://bhooi.github.io/code/camo.zip>), and re-implement it in MaxCompute for detecting multiple blocks.
- **Naive:** The naive algorithm introduced in Section V-A and we implement it in MaxCompute.
- **RICD-UI:** Comparing to RICD, it removes the whole suspicious group screening module. We use this method to illustrate the necessity of the suspicious group screening module.
- **RICD-I:** Comparing to RICD-UI, it only removes the item behavior verification step in the suspicious group screening module. Through this method, we can analyze whether user behavior check step and item behavior verification step are effective.

Evaluation Metrics. We use four metrics, namely precision, recall, F1-score and elapsed time, to evaluate our detection framework RICD and compare RICD with baseline methods. Some of the experimental results (marked with $\star$) shown below come from the experts’ judgments on the sampling results for large output. We define the precision and recall rate as follows:

$$\text{precision} = \frac{\text{Number of detected abnormal nodes}}{\text{Number of output abnormal nodes}} \quad (5)$$

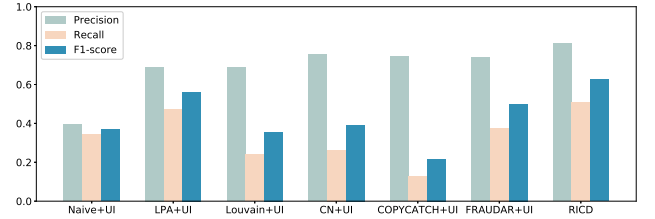
$$\text{recall} = \frac{\text{Number of detected abnormal nodes}}{\text{Number of known abnormal nodes}} \quad (6)$$

We record the end-to-end time of executing these modules as the whole elapsed time, and we need to ensure that the algorithm can output results in an acceptable time. Note that because the number of abnormal nodes in the dataset is more than the known abnormal nodes, the precision rate shown in the results will be lower than the true precision rate, but it is fair for all the algorithms in our experiments.

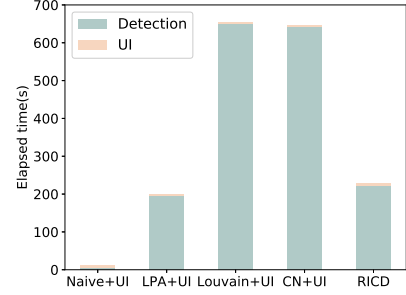
B. Comparing with Baseline Methods

In this section, we compare our framework with the baseline methods to see the effectiveness of suspicious group detection, user behavior check and item behavior verification techniques. We leave the parameters unchanged with $k_1 = 10, k_2 = 10, \alpha = 1.0, T_{hot} = 1,000$ and $T_{click} = 12$.

Effectiveness of Suspicious Group Detection. Because all baselines do not have suspicious group screening module, for the sake of fairness, we add the suspicious group screening



(a) Comparisons of baselines by Precision, Recall and F1-score



(b) Comparisons of baselines by elapsed time

Figure 8. Comparisons of baselines by Precision, Recall, F1-score, and elapsed time

module (User behavior check and Item behavior verification, UI) to all baselines. For example, in the experiments, after the louvain algorithm outputs the partition results (into communities), we filter out communities that do not include enough users and items (less than k_1 and k_2), then perform user behavior check and item behavior verification in every remaining community. The effectiveness of suspicious group screening module can be verified by the performance gap between baselines in Fig. 8a and Fig. 8b.

In Fig. 8a, the best performer is RICD, followed by LPA, FRAUDAR, CN, Naive, Louvain, and COPYCATCH (for brevity, remove “+UI”) in the order of increasingly worse performance. RICD consistently outperforms other baselines with respect to the precision, recall and F1-score. Specifically, compared with the two closest algorithms FRAUDAR and LPA, RICD achieves an improvement over 18% in the precision measure while maintaining the approximately same recall measure with LPA and achieves an improvement over 35% in the recall measure while maintaining a competitive precision measure with FRAUDAR. In general, the dense-graph-based methods can acquire a higher precision, while the recall of community-based methods is relatively high. Only RICD has advantages in both measures. The gap in recall measure between CN and RICD indicates that only considering neighbor information will cause many abnormal users or items to be erroneously undetected. The results also show that the performance of COPYCATCH is poor without timestamps, but attack detection against the dense structure can indeed improve the precision. FRAUDAR has a good performance in the dataset with camouflages, but there is still a gap between FRAUDAR and RICD in F1-score.

Fig. 8b shows the elapsed time of different baselines. Because Grape can’t help accelerate the implements of COPY-

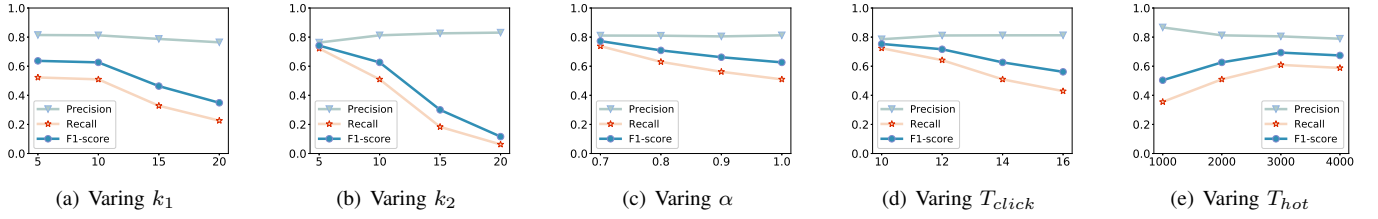


Figure 9. Sensitivity analysis by Precision, Recall and F1-score

CATCH and FRAUDAR, they are not included in the comparison of the elapsed time for the sake of fairness. We can observe that compared with the UI process, the elapsed time of the detection algorithm occupies most of the time. The naive algorithm has achieved the best performance on the elapsed time for its simple logic, but its performance on other measures is not satisfactory (see Naive+UI in Fig. 8a). Compared with the CN+UI and Louvain+UI algorithms, RICD is 35% lower in the elapsed time but achieves a better F1-score. The elapsed time of LPA, a near-linear time algorithm, is slightly (less than 15%) better than that of RICD.

Effectiveness of Suspicious Group Screening. To evaluate the effectiveness of suspicious group screening module in screening suspicious items and users, we compare the precision, recall and F1-score produced by RICD-UI, RICD-I and RICD, which are presented in Table VI. We can see that, with user behavior check and item behavior verification, the precision rate of the framework gradually increases despite a decrease in recall. The results of the F1-score show that RICD has achieved the best balance between precision and recall. By comparing RICD-UI and RICD-I with RICD, we find that removing both steps will decrease the performance to some extent. More specifically, the user behavior check step can help screen out normal users who hardly click unpopular items (a small amount) and distinguish abnormal users. These results will help the item behavior verification step screen out a lot of items to produce a relatively clean click list. So compared to adding the user behavior check step (from RICD-UI to RICD-I), adding the item behavior verification step (from RICD-I to RICD) can further improve the precision.

Table VI
EFFECTIVENESS OF SUSPICIOUS GROUP SCREENING

	Precision	Recall	F1-score
RICD-UI*	0.03	0.82	0.06
RICD-I*	0.14	0.78	0.23
RICD	0.81	0.51	0.63

C. Sensitivity Analysis

We conduct a parameter sensitivity analysis to see how various parameters affect the performance of our “Ride Item’s Coattails” attack detection framework. We vary five types of parameters in our framework:

- k_1 in Definition 2: $k_1 = 5, 10, 15, 20$.
- k_2 in Definition 2: $k_2 = 5, 10, 15, 20$.
- α in Definition 2: $\alpha = 0.7, 0.8, 0.9, 1.0$.
- abnormal click threshold: $T_{click} = 10, 12, 14, 16$.
- hot item threshold: $T_{hot} = 1,000, 2,000, 3,000, 4,000$.

The default parameter settings are $k_1 = 10, k_2 = 10, \alpha = 1.0, T_{click} = 12$ and $T_{hot} = 2,000$.

Fig. 9 shows the results for RICD. In general, the change of parameters has monotonic effect on the precision and recall measure. The only exception is the parameter T_{hot} (Fig. 9e). This is because that as T_{hot} increases from 1,000 to 4,000, a large number of abnormal items in this range are detected (positive) with the number of hot items drops, which will cause some attackers who have only clicked hot items in this range to be missed (negative). The experimental results show that setting T_{hot} to 3,000 can achieve the best recall. Significantly, the increase of k_1 and k_2 brings the opposite change to the precision measure, which indicates that in real scenarios, crowd workers tend to attack frequently (large k_2) on a small scale (small k_1).

VII. CASE STUDY

We take a “Ride Item’s Coattails” attack group detected by RICD in a recent marketing campaign in TaoBao as an example to explain how our framework works in the real-life system. This attack group consists of 13 items (2 hot items and 11 target items) and 28 accounts. In Figure. 10, we show the historical traffic data of these target items. It can be found that before the start date of the marketing campaign (Day 6), the abnormal traffic of these target items had begun to increase, which means that sellers post attack missions before the campaign starts. Because of this, the normal traffic grew rapidly from Day 6 to Day 9, proving that the “Ride Item’s Coattails” attack was launching. On Day 9, such a “Ride Item’s Coattails” attack group was detected by RICD, and the system cleaned the false click information. These target items’ traffic data was restored to the normal level (Day 10), and sellers soon removed these inferior items from their online store in TaoBao (Day 13). Based on the prediction result of the traffic model, our framework protects hundreds of thousands of users from incorrect recommendations in this campaign.

What’s more, using the known affiliation of item and seller and the detection technology for account association, we find that more than 85% of these accounts have associations with each other and 11 target items are only sold by 3 sellers. All of this information provide additional strong evidence that the users and items caught are very suspicious and prove that RICD can effectively catch “Ride Item’s Coattails” attack from large-scale click data.

VIII. CONCLUSIONS

In this paper, we investigate the “Ride Item’s Coattails” attack in online shopping. To have a better understanding of

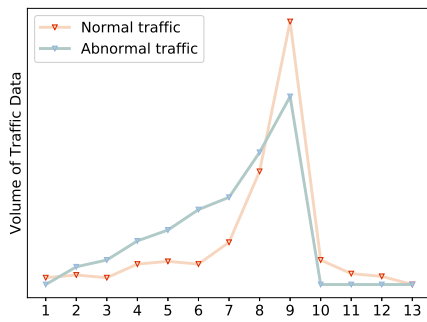


Figure 10. Historical traffic data of target items: the x-axis represents the timeline, and the y-axis represents the volume of traffic data. For security reasons, we don't display the specific event dates and the real volume of traffic data.

this type of newly-emerged malicious attack, we conduct a comprehensive analysis of the behavior of the users and items. Based on the analysis results, we propose a framework, named RICD, for detecting the “Ride Item’s Coattails” attack in e-commerce applications. In RICD, the “Ride Item’s Coattails” attack group is regarded as a dense near-biclique, and two suspicious node filtering strategies are proposed. The proposed dense bipartite graph structure composed of crowd workers and items are well interpretable and generic. The experimental results on the large-scale e-commerce data also show that the proposed methods are superior to the existing attack detection methods in terms of precision, recall and F1-score.

As the future work directions, it is important to study how to add an incremental data processing module to this framework so that it can be applied online to perform the “Ride Item’s Coattails” attack detection in dynamic graphs. In a large-scale online shopping scenario like Taobao’s “Double 11 Shopping Festival”, the earlier these attacks are detected in real time, the more losses can be reduced.

ACKNOWLEDGMENT

The authors would like to thank the support from NSFC grants (No. 61972155), Zhejiang Lab under No.2019KB0AB04, Zhejiang Provincial Natural Science Foundation (LZ21F030001) and the Science and Technology Commission of Shanghai Municipality (20DZ1100300).

REFERENCES

- [1] N. Su, Y. Liu, Z. Li, Y. Liu, M. Zhang, and S. Ma, “Detecting crowd-turfing “add to favorites” activities in online shopping,” in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1673–1682.
- [2] C. Xu and J. Zhang, “Towards collusive fraud detection in online reviews,” in *2015 IEEE international conference on data mining*. IEEE, 2015, pp. 1051–1056.
- [3] Y. Zhang, C. A. Phillips, G. L. Rogers, E. J. Baker, E. J. Chesler, and M. A. Langston, “On finding bicliques in bipartite graphs: a novel algorithm and its application to the integration of diverse biological data types,” *BMC bioinformatics*, vol. 15, no. 1, p. 110, 2014.
- [4] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, “Copycatch: stopping group attacks by spotting lockstep behavior in social networks,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 119–130.
- [5] S. O. Kuznetsov, “On computing the size of a lattice and related decision problems,” *Order*, vol. 18, no. 4, pp. 313–321, 2001.
- [6] B. Lyu, L. Qin, X. Lin, Y. Zhang, Z. Qian, and J. Zhou, “Maximum biclique search at billion scale,” *Proc. VLDB Endow.*, vol. 13, no. 9, pp. 1359–1372, 2020.
- [7] L. Wang, “Near optimal solutions for maximum quasi-bicliques,” *Journal of Combinatorial Optimization*, vol. 25, no. 3, pp. 481–497, 2013.
- [8] D. I. Ignatov, P. Ivanova, and A. Zamaletdinova, “Mixed integer programming for searching maximum quasi-bicliques,” in *International Conference on Network Analysis*. Springer, 2018, pp. 19–35.
- [9] X. Liu, J. Li, and L. Wang, “Quasi-bicliques: Complexity and binding pairs,” in *International Computing and Combinatorics Conference*. Springer, 2008, pp. 255–264.
- [10] K. Wu, S. Yang, and K. Q. Zhu, “False rumors detection on sina weibo by propagation structures,” in *2015 IEEE 31st international conference on data engineering*. IEEE, 2015, pp. 651–662.
- [11] F. Jin, E. Dougherty, P. Saraf, Y. Cao, and N. Ramakrishnan, “Epidemiological modeling of news and rumors on twitter,” in *Proceedings of the 7th workshop on social network mining and analysis*, 2013, pp. 1–9.
- [12] S. Kumar, R. West, and J. Leskovec, “Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes,” in *Proceedings of the 25th international conference on World Wide Web*, 2016, pp. 591–602.
- [13] B. Horne and S. Adali, “This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, 2017.
- [14] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, “Catchsync: catching synchronized behavior in large directed graphs,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 941–950.
- [15] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, “Fraudar: Bounding graph fraud in the face of camouflage,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 895–904.
- [16] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
- [17] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, “Module identification in bipartite and directed networks,” *Physical Review E*, vol. 76, no. 3, p. 036102, 2007.
- [18] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical review E*, vol. 76, no. 3, p. 036106, 2007.
- [19] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos, “Eigenspokes: Surprising patterns and scalable community chipping in large graphs,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2010, pp. 435–448.
- [20] S. Fortunato, “Community detection in graphs,” *Physics reports*, vol. 486, no. 3–5, pp. 75–174, 2010.
- [21] L. Akoglu, M. McGlohon, and C. Faloutsos, “Oddball: Spotting anomalies in weighted graphs,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2010, pp. 410–421.
- [22] Wikipedia contributors, “Pareto principle — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/w/index.php?title=Pareto_principle&oldid=967358099, 2020, [Online; accessed 26-August-2020].
- [23] C. Yan, J. G. Burleigh, and O. Eulenstein, “Identifying optimal incomplete phylogenetic data sets from sequence databases,” *Molecular phylogenetics and evolution*, vol. 35, no. 3, pp. 528–535, 2005.
- [24] P. Kővári, V. T. Sós, and P. Turán, “On a problem of zarankiewicz,” in *Colloquium Mathematicum*, vol. 3. Polska Akademia Nauk, 1954, pp. 50–57.
- [25] Z. Füredi, “An upper bound on zarankiewicz’ problem,” *Combinatorics, Probability and Computing*, vol. 5, no. 1, pp. 29–33, 1996.
- [26] W. Fan, W. Yu, J. Xu, J. Zhou, X. Luo, Q. Yin, P. Lu, Y. Cao, and R. Xu, “Parallelizing sequential graph computations,” *ACM Transactions on Database Systems (TODS)*, vol. 43, no. 4, pp. 1–39, 2018.
- [27] Alibaba Group, “Maxcompute,” <https://www.alibabacloud.com/product/maxcompute>, 2020, [Online; accessed 4-September-2020].
- [28] S. Daminelli, J. M. Thomas, C. Durán, and C. V. Cannistraci, “Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks,” *New Journal of Physics*, vol. 17, no. 11, p. 113037, 2015.
- [29] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

ATNN: Adversarial Two-Tower Neural Network for New Item's Popularity Prediction in E-commerce

Shen Xin*, Zhao Li[†], Pengcheng Zou[†], Cheng Long*, Jie Zhang*, Jiajun Bu[‡] and Jingren Zhou[†]

*Nanyang Technological University, Singapore

Email: {xin.shen, c.long, zhangj}@ntu.edu.sg

[†]Alibaba Group, China

Email: {lizhao.lz, xuanwei.zpc, jingren.zhou}@alibaba-inc.com

[‡]Zhejiang University, China

Email: bjj@zju.edu.cn

Abstract—The e-commerce era is witnessing a rapid increase in online shopping activities and rising new arrivals of items on e-commerce platforms every day. Identifying potential popular items accurately is of great importance in creating commercial value. Click-Through Rate (CTR) is a general indicator to evaluate item popularity. However, existing CTR prediction methods fail in new arrivals prediction because they are confronted with two main challenges: (1) sparse item features and missing item statistics; (2) high time complexity of computing for all pairs of users and items. To tackle the first challenge, we propose a novel Adversarial Two-tower Neural Network (ATNN) model for new arrivals CTR predictions by introducing an adversarial network to a two-tower network. We design a generator and a discriminator to better learn an item vector based on item profiles without item statistics. Moreover, we introduce a Deep & Cross Network (DCN) and a strategy of shared embeddings among multiple tasks to accurately predict CTR for new arrivals. To address the second challenge, we develop a strategy with an $\mathcal{O}(1)$ time complexity for a new item's popularity prediction. We first construct a user group and learn its mean user vector in a time-efficient manner. Then, an item's popularity score is calculated from its item vector along with the learned mean user vector, which leads to a huge decrease in computational overhead. This is the first work to introduce a generative adversarial network in an optimized two-tower neural network to efficiently learn new arrivals vectors and effectively make new arrivals predictions. We implement ATNN and examine it on a large-scale real-world dataset from one of the world's largest e-commerce platforms, "Tmall.com". Empirical results show that ATNN is strongly capable of learning item vectors from item profiles for CTR predictions. We also deploy ATNN on "Tmall.com". The online A/B test illustrates that ATNN is superior over human expert decisions in real-world applications. Furthermore, by introducing multi-task learning technology, we extend ATNN to another scenario, food delivery service, which is quite essential during the particular period of the COVID-19 pandemic. Deployment on one of the most popular food delivery platforms, "Ele.me", showcases that ATNN can be widely applied in various e-commerce services. Both offline and online experimental results demonstrate that ATNN can recognize attractive and welcoming new restaurants that have higher Value per Page View (VpPV) and generate more Gross Merchandise Volume (GMV).

Index Terms—E-commerce, New arrivals prediction, Click-through rate prediction, Neural network, Generative adversarial network, Cold-start, Food delivery.

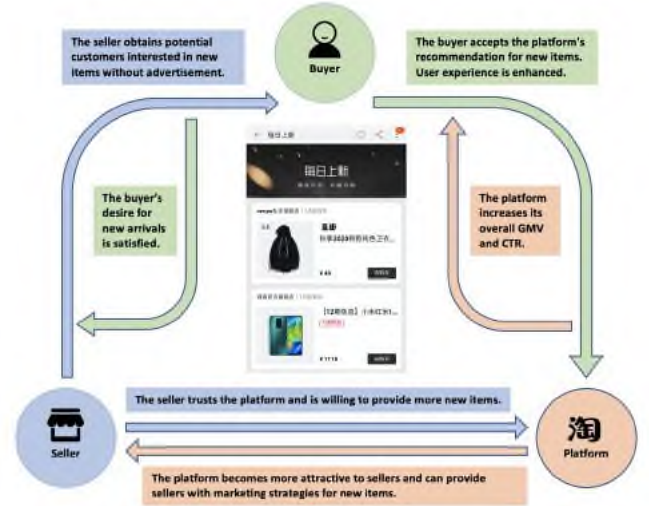


Fig. 1. A tripartite win-win mechanism based on new arrivals prediction

I. INTRODUCTION

With the rapid development of e-commerce, online shopping activities are surging. On a world-leading e-commerce platform, there are tens of millions of items released on the market every day. New arrivals prediction becomes a crucial task in the e-commerce industry. All online shopping participants, including sellers, buyers and platforms, benefit from accurate predictions of new arrivals. Personalized recommendations of new arrivals based on popularity estimates lead buyers to their preferred items, which satisfies buyers' desire for new arrivals and enhances user experience. Meanwhile, sellers are willing to provide more new items on this platform, because they benefit from increasing transactions of new arrivals. The e-commerce platform can also utilize the predicted popularity of new arrivals to implement various marketing strategies. Figure 1 shows a tripartite win-win mechanism based on new arrivals prediction.

Because there exists no general indicator to evaluate new arrivals' popularity, we adopt Click-Through Rate (CTR) as a key indicator to reflect item popularity. CTR Prediction

is a regular task in e-commerce. Accurate CTR prediction leads buyers to their preferred items to improve user shopping experience, by reducing time for retrieving merchandise and discovering items that they are potentially interested in but unaware of. Items would be exposed to a group of potential buyers more effectively, and thus more transactions are generated. Under such circumstances, both sellers' sales volume and platform's Gross Merchandise Volume (GMV) would increase.

CTR prediction for new arrivals is a more challenging task than that for existing items since it suffers from a cold-start problem. We have no information about any user behaviors on new items, and thus the item statistics information is missing. Traditional methods focus on better utilization of item profiles: finding "nearest neighbors" based on contents for new arrivals, then making collaborative filtering predictions and recommendations. However, these methods do not exploit historical user-item interaction data when calculating item similarities. They suffer from very sparse features and missing interactive features.

To address the challenge of missing item statistics data, inspired by Generative Adversarial Networks (GANs) [1], we introduce a generator and a discriminator as an adversarial component to a Two-tower Neural Network (TNN) structure for better feature extraction. TNN consists of two neural networks. One tower maps user features to user embeddings. The other tower maps item features and historical user-item interactions to item embeddings. The generator extracts information from item profiles into vectors such that they are similar to the item vectors from the item tower's encoder. Besides, the discriminator aims to distinguish from two types of item vectors, i.e., maximizing the difference between item vectors obtained by the generator and the item encoder. The proposed Adversarial Two-tower Neural Network (ATNN) framework can be trained by two steps iteratively. The first step is to obtain both user and item vectors from full features for CTR predictions. The second step is to ensure that generated item vectors from item profiles are similar to normal item vectors. Inspired by transfer learning and multi-task learning, we let two item embedding layers share their embeddings. This strategy improves the feature extraction ability of the generator for CTR prediction and accelerates training progress.

To better obtain high-level features, we introduce Deep & Cross Network (DCN) [2] to the ATNN framework to enhance the ability to extract user and item features by learning explicit cross features of bounded-degree and traditional deep representations jointly.

Directly obtaining all new arrivals' popularity for a ranking task is confronted with a challenge of high time complexity. Typically, we need to construct samples of a Cartesian product of all new arrivals and all users to perform overall CTR prediction. However, it is not necessary to calculate the scores of all user-item pairs for determining the item popularity. To reduce the running time, we propose to learn and store a mean user vector of a user group before the prediction stage. We obtain one item's popularity score from its item vector and the stored mean user vector with only $\mathcal{O}(1)$ time complexity.

To examine ATNN, we design both offline and online experiments on a large-scale dataset from "Tmall.com". Offline experiments are divided into two parts. The first mainly aims to demonstrate the high generation ability of the adversarial component in ATNN. We test the performance of personalized CTR prediction compared with several popular methods being used in industry. The second evaluates the new arrivals prediction of ATNN by multiple business indicators, including Item Page View (IPV), adding to favorite, sales volume, etc. Furthermore, We deploy ATNN on one of the world-leading e-commerce platforms, "Tmall.com", to show its superiority of new arrivals prediction over human expert decisions.

Moreover, in the view of the COVID-19 pandemic, food delivery service becomes essential for daily life. The number of applications from new offline restaurants for joining food delivery platforms has increased unprecedentedly. We apply ATNN to this novel online-to-offline (O2O) scenario to choose attractive and welcoming restaurants by their popularity prediction. Based on current business needs, we introduce multi-task learning technology into ATNN to balance two evaluation metrics, Value per Page View (VpPV) and GMV. We also conduct both offline and online experiments. The experimental results show that ATNN successfully selects good restaurants that have higher VpPV and generate more GMV. In the meantime, we take corporate social responsibility to battle COVID-19.

We summarize the major contributions of this paper as follows:

- 1) We propose a novel Adversarial Two-tower Neural Network (ATNN) framework for new arrivals Click-Through Rate (CTR) predictions by enhancing the feature extraction capability. An adversarial network with a generator and a discriminator is developed to better learn item vectors from only item profiles without item statistics.
- 2) We design a Deep & Cross Network (DCN) and a strategy of shared embeddings among multiple tasks in the ATNN framework. The optimized ATNN model is utilized towards better CTR prediction for new arrivals.
- 3) We develop a high-efficiency method with $\mathcal{O}(1)$ time complexity for each new item's popularity prediction. This is achieved by computing one item's popularity score from its item vector and a pre-learned mean user vector of a user group.
- 4) We show both offline and online results of our experimental evaluation on a large-scale real-world dataset from "Tmall.com", which demonstrates the scalability of our method and its superiority in both extracting item vectors from sparse features and solving the task of new arrivals prediction.
- 5) We extend the proposed ATNN model to another popular scenario, food delivery service, by introducing multi-task technology. Empirical results on "Ele.me" show a kind solution to a cold-start problem in our food delivery platform and indicate extensive commercial values of the ATNN model.

The rest of the paper is organized as follows. In Section II, we give a brief review of related work. Section III presents our ATNN model. The experimental results are presented in Section IV. Further application studies are provided in Section V. Finally, we summarize the current work and suggest directions for future work in Section VI.

II. RELATED WORKS

In this section, we briefly introduce some related work about this paper in three aspects. First, we summarize the cold-start challenge in e-commerce and its current solutions. Second, we review popular methods for CTR prediction. Third, we revisit the development of GAN.

A. Cold-start problems in e-commerce

Cold-start problem is a traditional but very challenging task in recommender systems. There are two subproblems: recommendation for new users and new items.

There are some studies to solve the new user cold-start problem. An intuitive way is to let a new user get non-personalized recommendations built on top-seller rankings. Some models focus on exploiting side information, including user attributes and user social networks, to mitigate the cold-start problem. [3] defines a novel social collaborative filtering framework that generalizes standard item-based collaborative filtering to the cold-start recommendation setting. [4] proposes a principled approach based on user multi-relational factorization techniques. Feature mapping models are also used for the cold-start problem, such as [5], which normally learns a feature mapping between the side-information and one of the latent feature representations.

The new arrivals cold-start problem is more challenging. Unlike top-seller rankings recommendations, we cannot recommend “top-buyers” to new items. Meanwhile, we cannot construct social networks for new items without user-item interactive data. Classical content-based collaborative recommendation [6] and its multiple extensions, such as [7] and [8], are utilized to address the item cold-start problem. Recently, a variety of active learning strategies are proposed to solve this problem in the literature, such as [9] and [10]. However, active learning still relies on a few user-item interactions to train models. Under the real-world circumstances in industry, e-commerce markets require to predict overall CTR for new arrivals before they are released on the market. Active learning is incompetent in this case.

B. CTR predictions

CTR prediction is a crucial task in e-commerce, which guides users to click their favored items to increase transactions on the platform and commercial values. Traditional methods for this task is based on logistic regression, such as traditional Logistic Regression (LR) [11], Follow-The-Regularized-Leader (FTRL) [12], Large Scale Piece-wise Linear Model (LS-PLM) [13], etc. Because linear models are weak for extracting sophisticated feature interactions, various

Factorization Machines (FMs) are proposed to improve CTR prediction by better learning interaction features [14].

Due to its powerful nonlinear modeling capabilities, DNNs are utilized for CTR prediction to automatically learn feature representations and high-order feature interactions recently. Two-tower structures are popular for CTR predictions. One item tower encodes item vectors from item features; the other user tower encodes user vectors from user features. Plenty of research works are related to this two-tower structure, such as [15], [16], etc. Deep & Cross Network (DCN) [2] introduces a novel cross network that is more efficient in learning certain bounded-degree feature interactions. Wide & Deep network [17] jointly trains wide linear models and deep neural networks to combine the benefits of memorization and generalization for recommender systems. Neural Factorization Machine (NFM) [18] is proposed for prediction under sparse settings, which seamlessly combines the linearity of FM in modeling second-order feature interactions and the non-linearity of neural network in modeling higher-order feature interactions. DeepFM [19] combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network architecture.

DNNs have been widely deployed in industrial applications. YouTube uses a deep candidate generation model with a separate deep ranking model for personalized recommendations [20]. Google introduces a large scale multi-objective ranking system for recommending videos to watch next [21]. Alibaba utilizes Deep Interest Network (DIN) to adaptively learn the representations of user interests from historical behaviors with respect to a certain advertisement by designing a local activation unit [22] [23]. However, these models need complete profiles of users and items, as well as user-item interactions, and they are not designed for solving cold-start problems.

C. Generative adversarial network

GANs are a class of machine learning systems invented by Goodfellow et al. in 2014 [1]. In a typical setting, GAN is composed of two components: one is a generative network G ; the other is a discriminative network D . The generator G generates samples from its distribution. The discriminator D determines whether a sample is real data or generated data. The training objective of a generative network is to increase the error rate of the discriminative network. The discriminative network learns to distinguish candidates produced by the generator from the true data distribution. Both D and G are trained by a minimax game.

GANs are widely used in image recognition tasks. Conditional GANs are an extension of the GAN framework. For example, Stacked Generative Adversarial Networks (StackGANs) focus on improving the quality of the image conditioned on text descriptions [24]. The Deep Convolutional Generative Adversarial Networks (DCGAN) is used to improve GAN’s performance by adding new constraints and modifying the CNN architecture [25]. GANs succeed in image super resolution [26], image to image translation [27], face aging [28] and video generation [29].

There are also some studies on GANs for information retrieval. For example, IRGAN [30] uses a generative model to learn to fit the relevance distribution over documents via the signals from a discriminative model and the discriminative model to exploit the unlabelled data selected by the generative model to achieve a better estimation for document ranking. Adversarial Personalized Ranking (APR) [31] enhances the pairwise ranking method Bayesian Personalized Ranking (BPR) by performing adversarial training. A general Minimax Game based method for selective Transfer Learning (MGTL) [32] is proposed for item recommendations by selecting high-quality source data to improve the transferring ability among different domains.

To predict popularity of new arrivals, we introduce a generative adversarial network in a two-tower network and utilize item vectors explicitly obtained from the two-tower network to enhance the generator's ability of feature extraction. We use item vectors obtained by the well-trained generator to accurately predict CTR and popularity for new arrivals.

III. ADVERSARIAL TWO-TOWER NEURAL NETWORK

In this section, we first describe the problem of new arrivals prediction. CTR is regarded as a crucial indicator of this problem. Then we present our Adversarial Two-tower Neural Network (ATNN) for ranking all new arrivals according to their popularity in detail.

A. Problem description

We aim to solve the cold-start problem for predicting popularity of new arrivals on e-commerce platforms. Since there is no common evaluation for new arrivals popularity, we use CTR prediction as a crucial task to solve this cold-start problem. Before releasing new items on the market, we utilize the model to make personalized recommendations according to CTR predictions. Accurate CTR predictions lead buyers to their preferred items, which satisfies buyers' consumption desire for new arrivals and enhances user experience. Meanwhile, sellers are willing to provide more new items on this platform, because they benefit from the increasing number of transactions for new arrivals.

Moreover, our main goal is to discover the potential popular items among new arrivals. Nevertheless, it is difficult for a model to evaluate the item popularity. We adopt that if one item has high CTR for a majority of buyers, it should be attractive with a high probability. Therefore, we extend our model to obtain the popularity of new arrivals among all users based on large-scale industrial data. This model enables to obtain the popularity of new items before releasing them on the market.

Specifically, we use item information of the first 30 days since they are released on the market as training data. We collect the statistics data of these items, including the number of Page Views (PV), the number of Unique Visitors (UV), series of user behaviors, such as clicking, adding to cart, adding to favorite and purchasing. We also obtain item profiles and user profiles as features of the training data. Item profiles

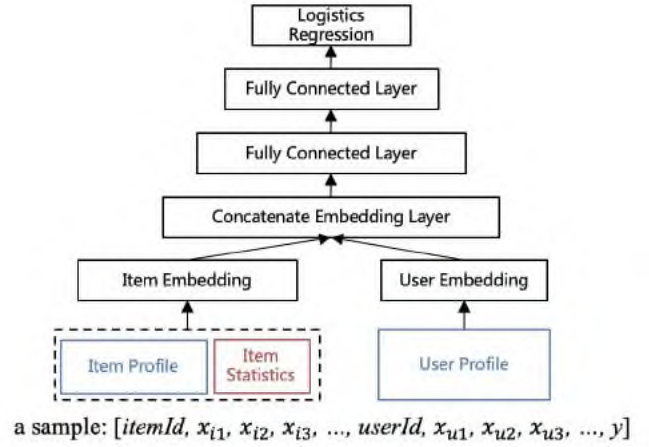


Fig. 2. A standard DNN model for pairwise user-item CTR prediction

contain seller information, product names, product images and categories. User profiles consist of personal data, such as user name, gender, location information, purchase preference, purchase power ratings, etc. New arrivals only have item profiles without item statistics information. Our goal is to rank all new arrivals according to their popularity among all users in the platform.

B. Two-tower neural network for ordinary pairwise user-item CTR prediction

DNNs are widely utilized for ordinary CTR prediction to automatically learn feature representations and high-order feature interactions. Item vectors obtained by DNNs can be used in various tasks, including popularity prediction for new arrivals.

Figure 2 shows a standard DNN model for pairwise user-item CTR prediction. It is a classical method that first concatenates an item embedding and a user embedding. We cannot obtain item vector nor user vector by this model.

To explicitly capture item vectors for new item's popularity prediction, we construct a two-tower neural network shown in Figure 3. The left tower extracts information from item profiles and item statistics to achieve item vectors; the right tower utilizes user profiles to get user vectors. We can explicitly capture item vector and user vector by this two-tower DNN. Obtained item vector and user vector can be utilized to train other models and solve other tasks related to pairwise CTR predictions. We use the obtained item vectors in our proposed adversarial network to train a generator, which will be described later.

We train the models by feeding each pair of one item and one user to the network, including the detailed interactions between every user-item pair. A sample of input is as follows:

$$[itemId, x_{i1}, x_{i2}, x_{i3}, \dots, userId, x_{u1}, x_{u2}, x_{u3}, \dots, y]$$

where $itemId$ and $userId$ are unique identifiers for each item and user, x_i and x_u indicate features of an item and a user, respectively, and y is the label for CTR prediction.

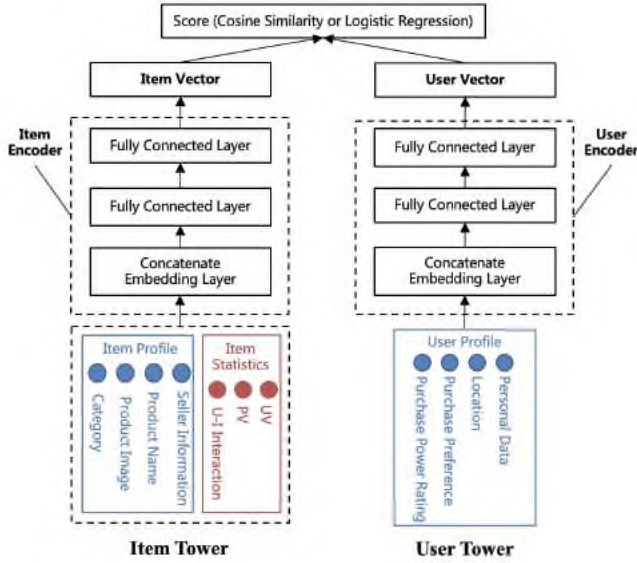


Fig. 3. A two-tower neural network for explicitly capturing item vector and user vector in pairwise user-item CTR prediction

C. Adversarial two-tower neural network framework for new arrivals CTR prediction

New arrivals CTR prediction differs from ordinary CTR prediction due to the lack of user-item interactions. For newly released items on the market, we usually have only a few user behaviors on them compared with the ordinary scenario. The data of clicking, adding to cart, adding to favorite and purchasing are too sparse to train a neural network. In addition, for new arrivals that have not been released on the market, there is no item statistics data. All classical methods face the challenge of missing item statistics information including PV, UV and series of user behaviors.

Motivated by the idea of GANs, we design an item generator and a discriminator for better learning item vectors from only item profiles. As we mentioned above, an original two-tower DNN model is able to achieve item vectors and user vectors, because there is an explicit layer behind item encoder and user encoder. We utilize the item vectors generated by the two-tower network to enhance the generator's ability of feature extraction. The quality of the generated item vector and user vector influences the precision of CTR prediction.

We propose to introduce an adversarial network in the two-tower structure for CTR predictions and call the resulting two-tower structure as *Adversarial Two-tower Neural Network* (ATNN) framework. The ATNN structure is shown in Figure 4. The left part is the adversarial component to learn to better extract item vectors using item profiles without any item statistics.

The generator is designed to generate item vectors from item profiles such that the generated item vectors are similar to item vectors generated by the item encoder, which is learned from both item profiles and item statistics. The discriminator is designed to distinguish generated item vectors generated

by the item generator from item vectors generated by the item encoder. The generator and discriminator play a minimax game to improve their performance. The loss function of the discriminator is based on the similarity between two types of item vectors and is defined as $\mathcal{L}_s$.

In addition, we use all item vectors generated by both the generator and the encoder to train the CTR prediction model in the framework. The loss function of the original two-tower model for the CTR prediction task is defined as $\mathcal{L}_i$. The loss function of CTR prediction between the generated item vectors and user vectors is defined as $\mathcal{L}_g$.

$$\mathcal{L}_i = -\frac{1}{N} \sum_{i=1}^N (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$$

$$\mathcal{L}_g = -\frac{1}{N} \sum_{i=1}^N (y_i \log \hat{y}_i^{(g)} + (1 - y_i) \log(1 - \hat{y}_i^{(g)}))$$

where $y_i \in \{0, 1\}$ is the label indicating whether the user clicks the item according to the records of user behaviors, $\hat{y} \in (0, 1)$ is the label predicted based on the item vector and the user vector, $\hat{y}^{(g)} \in (0, 1)$ is the label predicted based on the generated item vector and the user vector and N is the number of training samples.

We train this model by feeding each pair of one item and one user to the network, including the detailed interactions between every user-item pair. We iteratively optimize ATNN by reducing these three losses mentioned above. Both generation ability and CTR prediction quality can be enhanced.

Deep & Cross Network (DCN) [2] is utilized in all generators and encoders in the framework. DCN introduces a novel cross network that is more efficient in learning certain bounded-degree feature interactions without manual feature engineering. DCN explicitly applies feature crossing at each layer and adds negligible extra complexity to the DNN model. In e-commerce, there are plenty of high level features, e.g., item PV, seller PV and category PV. Manual feature engineering usually considers 2-level or 3-level features. Utilizing DCN can automatically obtain these bounded-degree features. In Section IV, we design a controlled experiment to test performance on both ordinary fully connected network and DCN in our model.

Furthermore, inspired by transfer learning and multi-task learning, we let two item embedding layers share their embeddings. The embedding layers map large-scale sparse features to low-rank vectors, requiring a large number of input samples to train a better model. Sharing features between embedding layers improves the generator component's ability for mapping item profiles to vectors.

We summarize the training steps for ATNN in Algorithm 1.

In each iteration, we first optimize ATNN by the following loss function:

$$\mathcal{L}_i(H(f_i(\mathbf{X}_i), f_u(\mathbf{X}_u)), y)$$

where $\mathbf{X}_i$ and $\mathbf{X}_u$ indicate features of one item and one user, respectively, $f_i(\mathbf{X}_i)$ and $f_u(\mathbf{X}_u)$ indicate item vector and

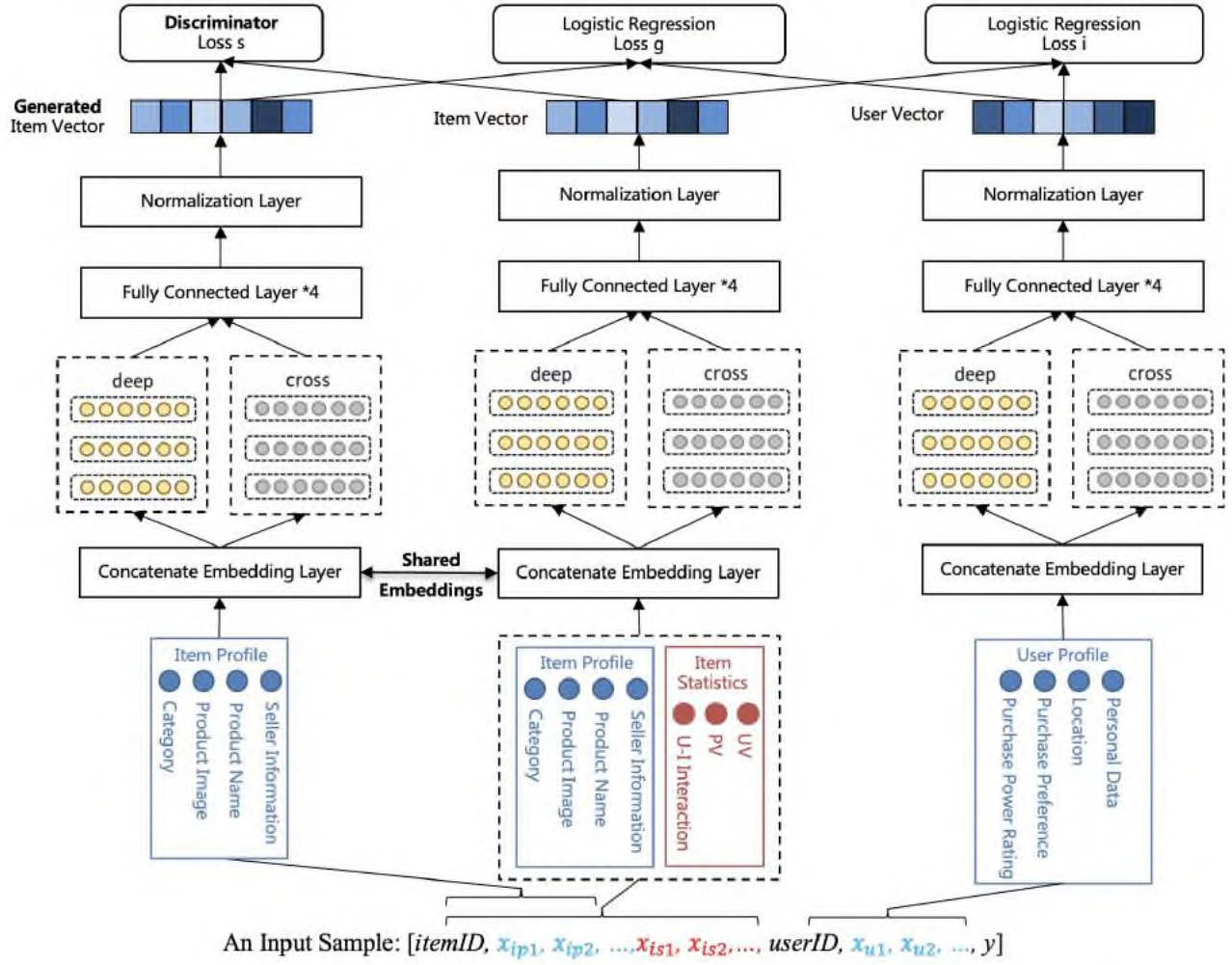


Fig. 4. Adversarial two-tower DNN model for new arrivals CTR prediction in E-commerce

user vector obtained by the item encoder and user encoder, respectively. $H(\cdot)$ function indicates CTR prediction score between one item and one user.

$\mathcal{L}_i$ uses logistic regression to evaluate the CTR predictions from both item profiles and item statistics information, according to given labels. In this step, we optimize CTR prediction by using item tower and user tower.

Then, we optimize ATNN by the following loss function:

$$\mathcal{L}_g(H(g(\mathbf{X}_{ip}), f_u(\mathbf{X}_u)), y) + \lambda \mathcal{L}_s(S(g(\mathbf{X}_{ip}), f_i(\mathbf{X}_i)))$$

where $\mathbf{X}_{ip}$ is the features of one item's profiles, $g(\mathbf{X}_{ip})$ is the generated item vector, λ is a weighting parameter to balance two losses and $S(\cdot)$ function indicates the similarity between a generated item vector and a normal item vector.

$\mathcal{L}_g$ uses logistic regression to evaluate the CTR predictions from only item profiles information, according to given labels. $\mathcal{L}_s$ uses the mean squared error as follows:

$$\mathcal{L}_s(X) = \text{mean}((1 - x_i)^2)$$

where $\mathcal{L}_s$ evaluates average similarity between generated item vectors and normal item vectors. In this step, we minimize the difference between generated item vector from the generator and item vector from the item encoder.

D. Large-scale new arrivals popularity prediction based on ATNN

We aim to discover potential popular new arrivals by ranking all items by their popularity. However, there is no general evaluation for rating items popularity. Based on a reasonable assumption that if one item has a high CTR for a majority of buyers, this product is very likely to be more attractive, we can utilize ATNN to estimate new arrivals popularity.

However, using a pairwise user-item CTR prediction model to achieve new arrivals popularity faces a critical challenge of high time complexity. In practice, for ranking new arrivals, we require to obtain all new arrivals popularity. In the prediction step, we need to construct a Cartesian product of all new arrivals and all users. Therefore, the time complexity of the prediction stage is $\mathcal{O}(N_U * N_{NA})$, where N_{NA} indicates the

Input: user features X_u , item profile features X_{ip} and item statistics features X_{is}

Output: generated item vector X_g

```

1  $X_i \leftarrow \text{CONCAT}(X_{is}, X_{ip})$ 
2 for number of training iterations do
3   Update D by minimizing loss:
4      $\mathcal{L}_i(H(f_i(X_i), f_u(X_u)), y)$ ;
5   Update G by minimizing loss:
6      $\mathcal{L}_g(H(G(X_{ip}), f_u(X_u)), y) + \lambda \mathcal{L}_s(S(G(X_{ip}), f_i(X_i)))$ ;
7    $X_g \leftarrow G(X_{ip})$ ;
8 end
9 Note:
10  $\mathcal{L}_i$ : ctr prediction loss between item vector  $G(X_i)$  and user vector  $f_u(X_u)$ 
11  $\mathcal{L}_g$ : ctr prediction loss between generated item vector  $G(X_{ip})$  and user vector  $f_u(X_u)$ 
12  $\mathcal{L}_s$ : similarity loss between generated item vector  $G(X_{ip})$  and item vector  $f_i(X_i)$ 

```

Algorithm 1: ATNN Algorithm

number of new arrivals. In one of the largest e-commerce platforms, there are hundreds of millions of existing users and millions of items released on the market every day. An $\mathcal{O}(N_U * N_{NA})$ algorithm does not work in a real-world system, especially in a world-leading e-commerce platform.

To rank new arrivals, it is unnecessary to obtain all pairs of user-item CTR predictions. Instead, we select the top 20 million active users who prefer new arrivals and regard them as a user group. We learn and store their mean user vector at the training stage. When predicting one item's popularity, we only need to utilize the stored mean user vector to make the prediction, which reduces the running time complexity for one item from $\mathcal{O}(N_U)$ to $\mathcal{O}(1)$. Figure 5 shows the efficient ATNN model for new arrivals popularity prediction.

IV. EXPERIMENTS AND RESULTS

We implement and evaluate our algorithm on a large-scale dataset of a world-leading e-commerce company. In this section, we design two offline experiments and one online experiment. The first offline experiment aims to demonstrate the high generation ability of the adversarial component in ATNN. The second offline experiment evaluates the new arrivals popularity prediction of ATNN by multiple business indicators. The online experiment shows results of A/B test in a real-world e-commerce.

A. Offline experiments on item generation ability of ATNN

In this experiment, we aim to demonstrate the high generation ability of the adversarial component in ATNN compared with other popular methods.

1) *Data description:* We conduct the experiment on a real-world dataset. There are 23,107,452 items, 4 million users and 40 million user-item interactions in this large-scale dataset. User profiles include personal data, location, purchase

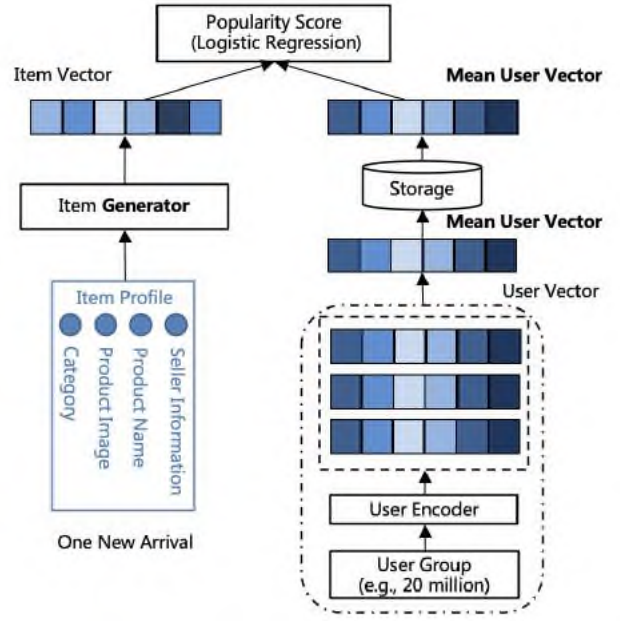


Fig. 5. ATNN for new arrivals popularity prediction

preference, purchase power rating, etc. Item profiles contain seller information, product name, product image, category, etc. Item statistics include UV, PV, user behaviors among user-item interactions, etc. The numbers of raw features of user profiles, item profiles and item statistics are 19, 38 and 46, respectively. We apportion the data into training and test sets, with an 80-20 split.

2) *Compared algorithms:* To the best of our knowledge, no existing CTR prediction model utilizes an adversarial component to generate item vectors in the case of missing item statistics. Our baseline algorithms are as follows:

- Gradient boosting decision tree (GBDT) [33]
- Two-tower neural network with fully connected layers (TNN-FC)
- Two-tower neural network with deep & cross network layers (TNN-DCN)
- Adversarial two-tower neural network (ATNN)

GBDT is a popular machine learning method for regression and classification problems in industrial applications. It produces a prediction model in the form of an ensemble of weak prediction models. TNN-FC is our baseline method. TNN-DCN is tested to show whether DCN can improve the prediction.

3) *Experimental settings:* Categorical features cannot be fed into networks directly. We map these categorical features to fixed-length vectors according to their numbers of categories. For instance, user identifier, user occupation, user category preference, item category and item sub-category are pre-processed into 16, 8, 16, 6 and 16 dimensional vectors, respectively. We use the same network structure for obtaining generated item vector, item vector and user vector. The numbers of dimensions in DCN are 512, 256 and 128. The numbers

TABLE I
RESULTS OF OFFLINE EXPERIMENTS ON ITEM GENERATION ABILITY OF ATNN

Model	AUC for only item profiles (cold start scenario)	AUC for complete item features (ideal baseline)	Performance degradation due to missing item statistics
GBDT	0.6149	0.6590	-6.69%
TNN-FC	0.5934	0.6048	-1.88%
TNN-DCN	0.6860	0.7169	-4.31%
ATNN	0.7121	0.7124	-0.04%

of dimensions for fully connected layers after DCN are 256, 256, 256 and 128. Therefore, the numbers of dimensions of generated item vectors, item vectors and user vectors are all 128. Empirically, we set the weighting parameter λ as 0.1 in our experiments.

4) *Evaluation metrics*: We use Area Under the Curve (AUC) to evaluate CTR predictions. Higher AUC means better predictions. However, in this experiment, we aim to evaluate the feature extraction ability of the proposed ATNN. Better predictions indicate a more powerful item generation ability. Performance degradation due to missing item statistics is defined as:

$$\text{Performance degradation} = \frac{\text{AUC}_{\text{profile only}} - \text{AUC}_{\text{complete}}}{\text{AUC}_{\text{complete}}}$$

5) *Experimental results and analysis*: Table I summarizes the experimental results. The following conclusions can be drawn from the experimental results:

- 1) ATNN and TNN-DCN outperform GBDT and TNN-FC in the traditional CTR prediction from complete item features including both item profiles and item statistics. Introducing DCN as a part of encoders in a two-tower network can better extract bounded-degree features to improve CTR prediction.
- 2) Performance decreases significantly for all three compared algorithms in new arrivals CTR predictions, i.e., in the case of missing item statistics. Specifically, performance degradations are -6.69% , -1.88% and -4.31% for GBDT, TNN-FC and TNN-DCN, respectively.
- 3) The generator in ATNN has a strong ability to extract item vectors from only item profiles. There is nearly no performance degradation from using item vectors obtained by the original item encoder and item vectors obtained by the generator, for CTR predictions. This experimental result demonstrates the high generation ability of the adversarial component in ATNN.

B. Offline commercial value validations on new arrivals popularity prediction of ATNN

In this offline experiment, we introduce several key indicators of e-commerce to evaluate the new arrivals popularity prediction of ATNN and to validate its commercial value.

1) *Data description and Experimental settings*: Both the dataset and the experimental settings, including network settings and parameter settings, are the same as the previous experiment. However, differing from pairwise user-item CTR predictions, this experiment is designed to show the quality of new arrivals popularity prediction. We first obtain popularity scores for all new arrivals. Then we divide these new arrivals into five groups to observe their performance in the first 30 days starting from the release day. Since there is no general evaluation metrics for judging the item popularity, we introduce some important business indicators in e-commerce for evaluations.

2) *Evaluation metrics*: We utilize several business evaluation metrics to present the quality of new arrivals popularity predictions. Three indicators are listed as follows:

- 1) Item Page View (IPV): the number of requests for viewing an item webpage. An item with a larger IPV means that its item webpage has been watched for more times.
- 2) Add to Favorite (AtF): the number of user behaviors of adding the item to their favorite list. AtF explicitly shows how many users like this item.
- 3) Gross Merchandise Volume (GMV): the prevailing evaluation metric in online retail to indicate a total sales dollar value for merchandise.

The observation terms are set as 7 days, 14 days and 30 days. We calculate the average IPV, average AtF and average GMV for each item group. Ideally, the higher-ranking group should have larger IPV, AtF and GMV.

3) *Experimental results and analysis*: Table II summarizes the experimental results in terms of IPV, AtF and GMV for 7-day, 14-day and 30-day observation terms in five item groups, which are divided by their predicted popularity.

We group new arrivals by their popularity scores. The first group containing the top 20% predicted popular new arrivals achieves the best performance among all three evaluation metrics, i.e., IPV, AtF and GMV, for the first 7 days and 14-days on the market. All these business indicators present a positive correlation with the popular scores predicted by ATNN. This experiment result shows that the proposed ATNN succeeds in the popularity prediction of new arrivals in e-commerce.

C. Online experiments on new arrivals popularity prediction of ATNN

We deploy ATNN in an e-commerce platform of a world-leading Internet company. We design an online A/B test by selecting 300 thousand potential popular new arrivals among tens of millions of items to compare ATNN with manual selections by experts. The average time for the first five successful transactions is used to evaluate the popularity of new arrivals. A new item is more attractive if it has five successful transactions in a shorter time. The average time for the first five successful transactions reflects the overall quality of new arrival selections. Shorter average time means a better prediction for new arrivals.

TABLE II
RESULTS OF OFFLINE COMMERCIAL VALUE VALIDATIONS ON NEW ARRIVALS POPULARITY PREDICTION OF ATNN

Popularity Ranking (Top %)	7-day IPV	14-day IPV	30-day IPV	7-day AtF	14-day AtF	30-day AtF	7-day GMV	14-day GMV	30-day GMV
0-20	63.94	132.24	199.30	1.06	2.19	3.46	51.40	110.50	226.32
20-40	42.95	92.43	148.06	0.70	1.50	2.49	28.84	63.63	147.35
40-60	39.09	81.65	135.23	0.60	1.26	2.23	41.63	91.74	240.46
60-80	21.65	45.10	77.44	0.37	0.72	1.28	16.68	35.58	91.55
80-100	16.20	32.43	53.81	0.27	0.55	0.92	13.34	27.85	73.14
Average	36.77	76.77	122.77	0.60	1.24	2.08	30.38	65.86	155.76

TABLE III
RESULTS OF ONLINE A/B TEST

Expert selection	ATNN selection	Improvement
10.47 days	9.72 days	7.16%

The experimental results of the online A/B test are shown in Table III. The average time of the first five transactions of new arrivals selected by experts is 10.47 days. Outperforming the experts by 7.16%, our ATNN method not only improves the popularity prediction of new arrivals but also reduces human resources costs.

D. Deployments

ATNN has been deployed on Alibaba’s online real-time data engine since August 2019. It is implemented in TensorFlow. The real-time data engine can obtain user behaviors, including clicking, adding to favorite, purchasing, etc. ATNN is capable of calculating new arrivals’ popularity. Nowadays, two downstream applications benefit from ATNN’s predictions. One is personalized search and recommendation, the other is smart selection of items for promotions.

V. FURTHER APPLICATIONS

We introduce multi-task learning technology and extend the ATNN model to a popular and essential scenario, food delivery service, to solve the cold-start problem of new restaurants. First, we describe the background and problem definition. Then, we propose the extended ATNN model for the food delivery scenario. Offline and online experimental settings and results are included at the end of this section.

A. Background and problem description

The rapid spread of the novel coronavirus, COVID-19, has raised alarm bells all over the world. To lower the risk of COVID-19 exposure and spread, a large number of restaurants and bars have suspended dine-in services and a lot of residents choose to stay home and keep social distancing. The demand for food delivery service has increased unprecedentedly.

Ele.me is an online-to-offline (O2O) catering and food delivery platform. As part of Alibaba Group’s “New Retail”, Ele.me had become the largest food delivery service in China with a 53.4% market share by 2019.¹

¹<https://daxueconsulting.com/o2o-food-delivery-market-in-china/>

During the battle period of COVID-19, tens of millions of restaurants have applied to Ele.me for providing delivery service on the platform. In order to try our best to ensure consumer safety and food quality, we must manually review the qualifications of each store, sign the contract and provide support services. It is impossible to review all applicants carefully in a short time. To create more commercial values, a model is required to assist us with selecting the most attractive and welcoming restaurants automatically, so that we can provide services to these merchants as soon as possible.

The problem turns out to be very similar to the new item’s popularity prediction in e-commerce because two main challenges still exist: (1) sparse item features and missing item statistics; (2) high time complexity of computing for all pairs of users and items. Since all new applicants have not joined our platform, we only have the basic features of these restaurants, for instance, brands, locations, themes, cuisines, etc., without any retail data. In addition, calculation on each pair of one restaurant and one user is time-consuming and unnecessary for predicting the popularity of a new merchant, because we have tens of millions of new restaurants and hundreds of millions of users. Motivated by this, we attempt to modify our ATNN model to apply in the food delivery platform to solve the problem of new restaurant’s popularity prediction.

Differing from choosing CTR prediction as a crucial task for new arrivals popularity prediction in e-commerce, in our real-world business scenario, Value per Page View (VpPV) and Gross Merchandise Volume (GMV) are keys to increasing business value. Unlike traditional online shopping, users tend to make quick decisions when they select food. The number of overall PVs in the food delivery platform should be much less than typical e-commerce. Current traffic allocation methods are difficult in creating more PVs. Based on the limited overall PVs, VpPV is crucial for evaluating a new restaurant. Besides, GMV is a general metric for measuring business value, which directly indicates the market size. Therefore, we change the training task from CTR prediction to VpPV prediction and GMV prediction.

B. Extended ATNN model for the food delivery scenario

As an O2O service, food delivery is location sensitive. Users often choose merchants close to them to keep food warm and fresh. We divide users into different groups based on their location information. Similar to Section III, we use mean

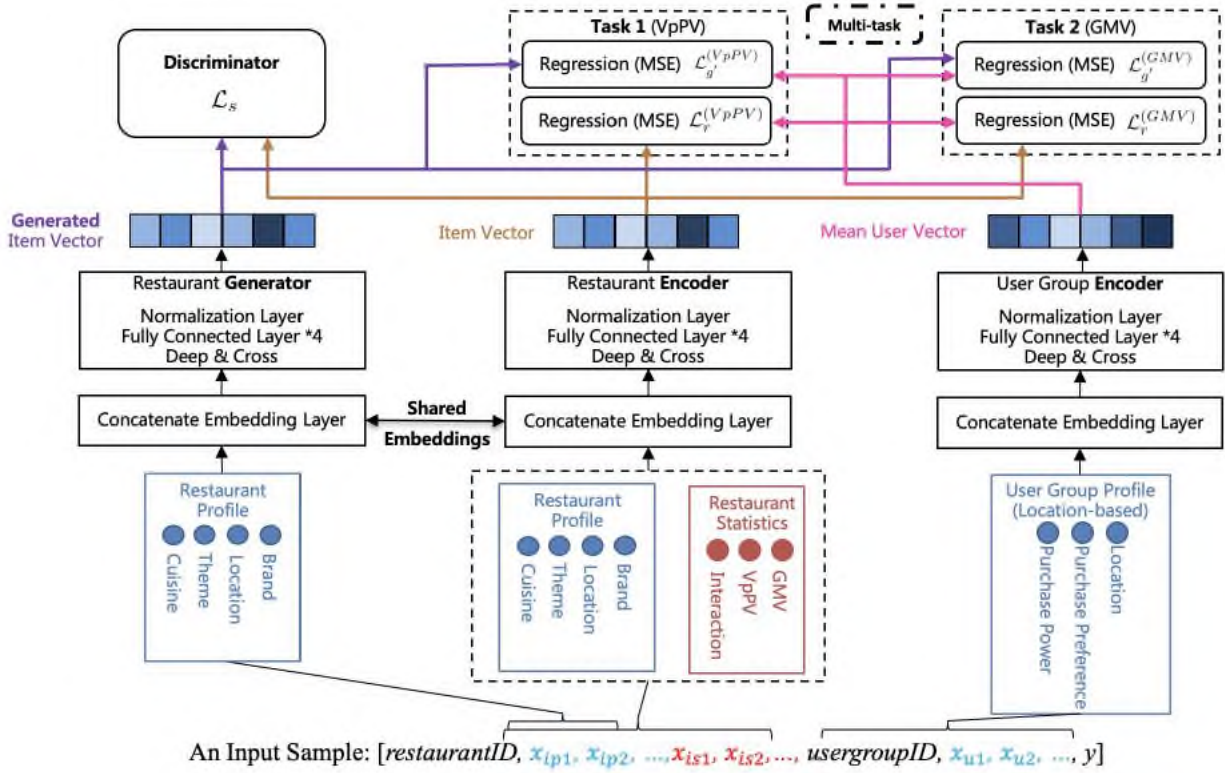


Fig. 6. Extended adversarial two-tower DNN model for new restaurant's popularity prediction in food delivery service

user features to replace single-user features to reduce time complexity.

To precisely predict new restaurant's popularity based on both VpPV and GMV aspects, we introduce multi-task learning technology in ATNN. Figure 6 shows the extended ATNN model to solve the cold-start challenge for new restaurants in a food delivery platform. By sharing networks, especially the restaurant representation among multiple tasks, our model can achieve better performance.

The loss functions for VpPV prediction task and GMV prediction task are as follows:

$$\begin{aligned}\mathcal{L}_r^{(GMV)} &= \frac{1}{N} \sum_{i=1}^N \left(y_i^{(GMV)} - \hat{y}_i^{(GMV)} \right)^2 \\ \mathcal{L}_{g'}^{(GMV)} &= \frac{1}{N} \sum_{i=1}^N \left(y_i^{(GMV)} - \hat{y}_i^{(g') (GMV)} \right)^2 \\ \mathcal{L}_r^{(VpPV)} &= \frac{1}{N} \sum_{i=1}^N \left(y_i^{(VpPV)} - \hat{y}_i^{(VpPV)} \right)^2 \\ \mathcal{L}_{g'}^{(VpPV)} &= \frac{1}{N} \sum_{i=1}^N \left(y_i^{(VpPV)} - \hat{y}_i^{(g') (VpPV)} \right)^2\end{aligned}$$

where $y_i^{(GMV)}$ is the GMV of $restaurant_i$, $\hat{y}_i^{(GMV)}$ is the GMV predicted based on the item vector and the mean user vector, $\hat{y}_i^{(g') (GMV)}$ is the GMV predicted based on the

generated item vector and the mean user vector and N is the number of training samples.

In each iteration, we first optimize ATNN by the following loss function:

$$\begin{aligned}\mathcal{L}_r^{(GMV)}(H(f_i(\mathbf{X}_i), f_u(\mathbf{X}_u)), y^{(GMV)}) \\ + \lambda_1 \cdot \mathcal{L}_r^{(VpPV)}(H(f_i(\mathbf{X}_i), f_u(\mathbf{X}_u)), y^{(VpPV)})\end{aligned}$$

Then, we optimize ATNN by the following loss function:

$$\begin{aligned}\mathcal{L}_{g'}^{(GMV)}(H(g(\mathbf{X}_{ip}), f_u(\mathbf{X}_u)), y^{(GMV)}) \\ + \lambda_1 \cdot \mathcal{L}_{g'}^{(VpPV)}(H(f_i(\mathbf{X}_{ip}), f_u(\mathbf{X}_u)), y^{(VpPV)}) \\ + \lambda_2 \cdot \mathcal{L}_s(S(g(\mathbf{X}_{ip}), f_i(\mathbf{X}_i)))\end{aligned}$$

where $\mathbf{X}_{ip}$ is the features of one restaurant's profiles, $g(\mathbf{X}_{ip})$ is the generated item vector, λ_1, λ_2 are weighting parameters to balance two losses and $S(\cdot)$ function indicates the similarity between a generated item vector and a normal item vector.

We summarize the training steps for extended ATNN with multi-task learning in Algorithm 2.

C. Experimental settings and results

We implement both offline and online experiments to test extended ATNN in a food delivery scenario. Offline experiments are designed to verify the effectiveness of our model compared with other popular DNN models. Online experiments are used for resolving real-world business to show that our model outperforms human expert decisions in creating business value.

Input: user group features $\mathbf{X}_u$, restaurant profile features $\mathbf{X}_{ip}$ and restaurant statistics features $\mathbf{X}_{is}$

Output: generated vector $\mathbf{X}_g$

```

1  $\mathbf{X}_i \leftarrow \text{CONCAT}(\mathbf{X}_{is}, \mathbf{X}_{ip})$ 
2 for number of training iterations do
3   Update D by minimizing loss:
    $\mathcal{L}_r^{(GMV)}(H(f_i(\mathbf{X}_i), f_u(\mathbf{X}_u)), y^{(GMV)}) + \lambda_1 \cdot$ 
    $\mathcal{L}_r^{(VpPV)}(H(f_i(\mathbf{X}_i), f_u(\mathbf{X}_u)), y^{(VpPV)});$ 
4   Update G by minimizing loss:
    $\mathcal{L}_{g'}^{(GMV)}(H(g(\mathbf{X}_{ip}), f_u(\mathbf{X}_u)), y^{(GMV)}) + \lambda_1 \cdot$ 
    $\mathcal{L}_{g'}^{(VpPV)}(H(f_i(\mathbf{X}_i), f_u(\mathbf{X}_u)), y^{(VpPV)}) +$ 
    $\lambda_2 \mathcal{L}_s(S(G(\mathbf{X}_{ip}), f_i(\mathbf{X}_i)));$ 
5    $\mathbf{X}_g \leftarrow G(\mathbf{X}_{ip});$ 
6 end
```

Algorithm 2: Extended ATNN Algorithm for New Restaurant’s Popularity Prediction in Food Delivery

TABLE IV
RESULTS OF OFFLINE EXPERIMENTS FOR FOOD DELIVERY

Model	VpPV (MAE)	GMV (MAE)
TNN-DCN	0.077	1.445
ATNN	0.069	1.206
Improvement	10.4%	16.5%

1) *Offline experiments:* We utilize a dataset from “Ele.me” containing 1.2 million new sign-up restaurants. VpPV and GMV of the first 30-day upon sign-up are regarded as labels. We apportion the data into training and test sets with an 80-20 split. The features of restaurants include brands, locations, themes, cuisines, the number of similar restaurants nearby, overall VpPV, overall GMV, overall CTR, etc. The number of dimensions of the restaurants after data preprocessing is 211. λ_1 and λ_2 are set as 100 and 10. The network structure is the same as the previous ATNN model mentioned in Section IV. In the O2O scenario, there are more categorical features reducing the performance of GBDT. Previous experiments also show that GBDT and TNN-FC are significantly weaker than TNN-DCN. Therefore, we regard TNN-DCN as the compared algorithm and Mean Absolute Error (MAE) of VpPV and GMV as the evaluation metrics.

Table IV shows the results of offline experiments for food delivery. Compared with TNN-DCN, our ATNN model reduces the MAE of VpPV from 0.077 to 0.069 by 10.4% and the MAE of GMV from 1.445 to 1.206 by 16.5%. Introducing the adversarial network improves the accuracy of VpPV and GMV predictions for new restaurants significantly.

2) *Online experiments:* In traditional food delivery platform, we hire human experts to select competitive merchants for sign-up. Benefiting from the accurate forecasts by ATNN, we can filter tens of millions of new applications automatically. During the online experiment, we compare ATNN to human experts. Both ATNN and human experts select attractive and

TABLE V
RESULTS OF ONLINE EXPERIMENTS FOR FOOD DELIVERY

Source	VpPV	GMV
Human Experts	0.2656	191.23
ATNN	0.2872	219.33
Improvement	8.1%	14.7%

welcoming restaurants for sign-up in our business. Reviewing the first 30-day performance of newly recruited restaurants in this July, we regard actual VpPV and GMV as evaluation metrics.

Table V shows the results of online experiments for food delivery. Compared with human experts, our ATNN model increases VpPV from 0.2656 to 0.2872 by 8.1% and GMV from 191.23 to 219.33 by 14.7%. Reviewing this A/B test, our ATNN model outperforms human experts on recruiting new restaurants in creating more business value.

VI. CONCLUSIONS

In this paper, we study a problem in e-commerce about new arrivals prediction, which is widely existed. Because there is no general indicator to evaluate new item popularity, we introduce Click Through Rate (CTR) to indicate item popularity. Existing CTR prediction models fail in new arrivals CTR prediction because of the sparse interactive features and missing item statistics. To tackle this challenge, we propose an Adversarial Two-tower Neural Network (ATNN) framework, which utilizes a novel adversarial component consisting of a generator and a discriminator. The adversarial component learns to better generate item vectors from only item profiles for CTR prediction. Besides, we introduce Deep & Cross Network (DCN) and shared embeddings in the ATNN framework. Experimental results show that ATNN has a powerful ability to generate item vectors from item profiles in CTR predictions. Another challenge in new arrivals prediction lies in huge overhead since pairwise computing between new arrivals and hundreds of million users is computationally costly. To address this problem, we construct a user group and store its mean user vector before the prediction stage. To predict a new item, we can obtain its popularity score from the item vector generated by the item generator using item profiles and the mean user vector stored in the system. This prediction step has only $\mathcal{O}(1)$ time complexity. Extensive offline and online experimental results on a large-scale e-commerce platform, “Tmall.com”, validate the effectiveness and efficiency of our ATNN. In addition, by introducing multi-task learning, we extend ATNN to a food delivery platform, “Ele.me”, to accurately predict VpPV and GMV of new restaurants.

There are several interesting problems to be investigated in our future work:

- It will be very interesting and valuable to extend our ATNN model to multiple online applications. Cold-start challenge exists widely in various online services using personalized recommendations or personalized information retrievals. ATNN utilizes an adversarial network to

generate item vectors from only item profiles. Besides in e-commerce, this strategy can be applied to other scenarios, for example, movie recommendation.

- We can further group users by their preferences before making new arrivals predictions. Different groups have diverse preferences for different types of items. Introducing this information may further improve the accuracy of new arrivals predictions.

ACKNOWLEDGMENT

This work is partly conducted within the Delta-NTU Corporate Lab for Cyber-Physical Systems with funding support from Delta Electronics Inc. and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

We would like to thank Alibaba Group for valuable datasets and platforms and Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies (AZFT) for technical support.

REFERENCES

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [2] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, 2017, pp. 1–7.
- [3] S. Sedhain, S. Sanner, D. Brazhinas, L. Xie, and J. Christensen, "Social collaborative filtering for cold-start recommendations," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 345–348.
- [4] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 173–182.
- [5] S. Sedhain, A. K. Menon, S. Sanner, L. Xie, and D. Brazhinas, "Low-rank linear cold-start recommendation from social data," in *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [6] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.
- [7] M. Nasery, M. Braunhofer, and F. Ricci, "Recommendations with optimal combination of feature-based and item-based preferences," in *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, 2016, pp. 269–273.
- [8] M. Saveski and A. Mantrach, "Item cold-start recommendations: learning local collective embeddings," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 89–96.
- [9] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai, "Addressing the item cold-start problem by attribute-driven active learning," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [10] M. Aharon, O. Anava, N. Avigdor-Elgrabli, D. Drachler-Cohen, S. Golan, and O. Somekh, "Excuseme: Asking users to help in item cold-start recommendations," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 83–90.
- [11] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: estimating the click-through rate for new ads," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 521–530.
- [12] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin *et al.*, "Ad click prediction: a view from the trenches," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1222–1230.
- [13] K. Gai, X. Zhu, H. Li, K. Liu, and Z. Wang, "Learning piece-wise linear models from large scale data for ad click prediction," *arXiv preprint arXiv:1704.05194*, 2017.
- [14] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.
- [15] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Z. Zhao, L. Wei, and E. Chi, "Sampling-bias-corrected neural modeling for large corpus item recommendations," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 269–277.
- [16] M. Naumov, D. Mudigere, H.-J. M. Shi, J. Huang, N. Sundaraman, J. Park, X. Wang, U. Gupta, C.-J. Wu, A. G. Azzolini *et al.*, "Deep learning recommendation model for personalization and recommendation systems," *arXiv preprint arXiv:1906.00091*, 2019.
- [17] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.
- [18] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.
- [19] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 1725–1731.
- [20] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.
- [21] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, and E. Chi, "Recommending what video to watch next: a multitask ranking system," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 43–51.
- [22] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1059–1068.
- [23] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5941–5948.
- [24] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.
- [25] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [26] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.
- [27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [28] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5810–5818.
- [29] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535.
- [30] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "Irgan: A minimax game for unifying generative and discriminative information retrieval models," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 515–524.
- [31] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 355–364.
- [32] B. Wang, M. Qiu, X. Wang, Y. Li, Y. Gong, X. Zeng, J. Huang, B. Zheng, D. Cai, and J. Zhou, "A minimax game for instance based selective transfer learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 34–43.
- [33] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

Billion-scale Pre-trained E-commerce Product Knowledge Graph Model

Wen Zhang*
Zhejiang University
Hangzhou, China
wenzhang2015@zju.edu.cn

Chi-Man Wong*
Alibaba Group,
University of Macao
chiman.wcm@alibaba-inc.com

Ganqiang Ye
Zhejiang University
Hangzhou, China
yeganqiang@zju.edu.cn

Bo Wen
Zhejiang University
Hangzhou, China
wenbo1@zju.edu.cn

Wei Zhang
Alibaba Group
Hangzhou, China
lantu.zw@alibaba-inc.com

Huajun Chen[§]
College of Computer Science
Hangzhou Innovation Center
Zhejiang University
AZFT Joint Lab for Knowledge Engine
huajunsir@zju.edu.cn

Abstract—In recent years, knowledge graphs have been widely applied to organize data in a uniform way and enhance many tasks that require knowledge, for example, online shopping which has greatly facilitated people's life. As a backbone for online shopping platforms, we built a billion-scale e-commerce product knowledge graph for various item knowledge services such as item recommendation. However, such knowledge services usually include tedious data selection and model design for knowledge infusion, which might bring inappropriate results. Thus, to avoid this problem, we propose a Pre-trained Knowledge Graph Model (PKGM) for our billion-scale e-commerce product knowledge graph, providing item knowledge services in a uniform way for embedding-based models without accessing triple data in the knowledge graph. Notably, PKGM could also complete knowledge graphs during servicing, thereby overcoming the common incompleteness issue in knowledge graphs. We test PKGM in three knowledge-related tasks including item classification, same item identification, and recommendation. Experimental results show PKGM successfully improves the performance of each task.

Index Terms—knowledge graph, pre-training, e-commerce

I. INTRODUCTION

Online shopping has greatly contributed to the convenience of people's lives. More and more retailers open online shops to attract increasing people favoring online-shopping, thus the e-commerce era witnessed rapid development in the past few years. The growing transaction on e-commerce platforms is based on billions of items of all kinds, which should be well organized to support the daily business.

Knowledge Graphs (KGs) represent facts as triples, such as (*iPhone*, *brands*, *Apple*). Due to the convenience of fusing data from various sources and building semantic connections between entities, KGs have been widely applied in industry and many huge KGs have been built (e.g., Google's Knowledge Graph, Facebook's Social Graph, and Alibaba/Amazon's e-commerce Knowledge Graph). We built an e-commerce product knowledge graph and make it a uniform way to

integrate massive information about items on our platform, which helps us build an integrated view of data. Notably, the e-commerce product knowledge graph contributes to a variety of tasks ranging from searching, question answering, recommendation, and business intelligence, etc. Currently, our product KG contains 70+ billion triples and 3+ million rules.

Our product KG has been widely used for knowledge-enhanced e-commerce applications, such as item attributes prediction, same item identification, item category prediction, and item recommendation. These are key tasks for managing more complete and clean data and providing better item services for customers. In supervised paradigms, these knowledge-enhanced tasks rely heavily on data that are vital for them, for example, item title for same item identification and user-item interaction bipartite graph for recommendation. Apart from these data, information contained in the product knowledge graph about items' attributes such as brands, series, and colors, are valuable side information to enhance these tasks.

To serve these knowledge-enhanced tasks, we used to provide item information formed as triples in product KG to facilitate knowledge enhancement. While it not only results in tedious data selection working for us, mainly refers to neighbor sampling, but also leaves tedious knowledge enhancement model design for downstream use. More importantly, similar to other KGs, our product KG is still far from complete and the key missing information may bias those downstream tasks.

Recently, a variety of knowledge graph embedding methods [1] are proposed for completion, which learn representation for entities and relations in continuous vector spaces, called embeddings, and infer the truth value of a triple via embedding calculation. Embeddings usually encode global information of entities and relations and are widely used as general vector space representations for different tasks. Thus for services of knowledge-enhanced tasks, instead of providing triple data, we are wondering that **is it possible to make embeddings from product KG as a uniform knowledge provider for knowledge-enhanced tasks, which not only avoid tedious**

*Equal contribution.

[§]Corresponding author.

data selection and model design, but also overcome the incompleteness of product KG.

Since the concept of ‘pre-training and fine-tuning’ has proven to be very useful in the Natural Language Processing community [2], referring to pre-training a language model with a huge amount of language data and then fine-tune it on downstream tasks such as sentiment analysis [3], relation extraction [4], text classification [5] and so on. We also consider to pre-train our product KG and make it possible to conveniently and effectively serve downstream tasks requiring background knowledge about items. The convenience refers that downstream tasks do not need to revise their model much to adapt to background knowledge. The effectiveness refers that downstream tasks enhanced with knowledge could achieve better performance, especially with a small amount of data.

The purpose of **Pre-trained Knowledge Graph Model (PKGM)** is to provide knowledge services in continuous vector space, through which downstream tasks could get necessary fact knowledge via embedding calculation without accessing triple data. The key information of product KG concerned by downstream tasks mainly includes two parts: (1) whether one item has a given relation(attribute), (2) what is the tail entity(property value) of given an item and relation(attribute). Thus making the incompleteness issue of knowledge graph also into consideration, a pre-trained knowledge graph model should be capable of: (1) showing whether an entity has a certain relation, (2) showing what is the tail entity for a given entity and relation, and (3) completing the missing tail entity for a given entity and relation if it should exist.

In this paper, we introduce a pre-trained e-commerce product knowledge graph model with two modules. One is Triple Query Module to encode the truth value of an input triple, which could serve for tail entity queries after training. Since a lot of knowledge graph embedding methods are proposed for triple encoding and link prediction, we apply the simple and effective TransE [1] in the triple query module. It represents entities and relations as vectors and restricts the addition of head entity embedding and relation embedding should approach tail entity embedding if the input triple is true. The other one is Relation Query Module to encode the existence of a relation for an entity, which could serve for queries of relation existence after training. In this module, we take entity and relation embeddings from the triple query module and create an entity transformation matrix $\mathbf{M}_r$ for each relation r . With $\mathbf{M}_r$, we make the transformed head entity embedding $\mathbf{h}'$ approaches to relation embedding $\mathbf{r}$ if h owns relation r .

After pre-training, the triple query module and the relation query module could provide knowledge service vectors for any target entity. Triple query module provides service vectors containing tail entity information of target entity and relation. For example, suppose the target entity as a smartphone, the triple query module will provide predicted tail entity embedding with target relations such as *brands*, *series*, *memory*, etc., no matter if these triples exist in the knowledge graph or not. Thus advantages of the triple query module are that (1) it could provide knowledge service based on computation in vector

space without accessing triples data, (2) it could complete the tail entity of target entity and relation during service. Relation query module provides service vectors containing existence information of different relations for target entities, in which the corresponding service vector will approach zero vector if the target entity (should) have the relation. Combining service vectors from triple query module and relation query module, information could be provided without executing symbolic queries by the pre-trained knowledge graph model are more than triples contained in the product knowledge graph.

With service vectors, we propose general ways to incorporate them in embedding-based methods, called base models, for downstream tasks. We classify embedding-based methods into two types according to the number of input embeddings of target entities. The first type refers to methods with a *sequence of embeddings* for target entities, and the second one refers to a *single embedding* for target entities. We propose a general way to apply service vectors into each of the two types of methods. For sequence-embedding inputs, we extend the input sequence by appending service vectors after original inputs of target entities. And for single-embedding inputs, we condense service vectors into one and concatenate it with the original input for target entities.

We pre-train the billion-scale product knowledge graph with PKGM model and apply service vectors to enhance three item related tasks, item classification, same product identification and item recommendation, following our proposed servicing and application methods. The experimental results show that PKGM successfully improves performance of these three tasks with a small amount of training data.

In summary, contributions of this work are as follows:

- We propose a way to pre-train a knowledge graph, which could provide knowledge services in vector space to enhance knowledge-enhance downstream tasks in a general way.
- Our proposed PKGM has two significant advantages, one is completion capability, and the other one is triple data independency.
- We practice PKGM on our billion-scale product knowledge graph and test it on three tasks, showing that PKGM successfully enhances them with item knowledge and improves their performance.

II. PRE-TRAINED KNOWLEDGE GRAPH MODEL (PKGM)

Product knowledge graph $\mathcal{K} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$ stores information of items and relations between items as triples, in which $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{F}$ are entity set, relation set and triple set respectively. $\mathcal{E} = \{\mathcal{I}, \mathcal{V}\}$ contains a set of items $\mathcal{I}$ and a set of values $\mathcal{V}$. $\mathcal{R} = \{\mathcal{P}, \mathcal{R}'\}$ contains a set of item properties $\mathcal{P}$ and a set of relations between items $\mathcal{R}'$. Our product knowledge graph contains more than 70 billion triples.

We propose a pre-trained knowledge graph model to encode $\mathcal{K}$ into continuous vector space, making it possible to provide knowledge services for knowledge-enhanced downstream tasks based on calculation in vector space. Knowledge services for other tasks usually refer to returning data in knowledge

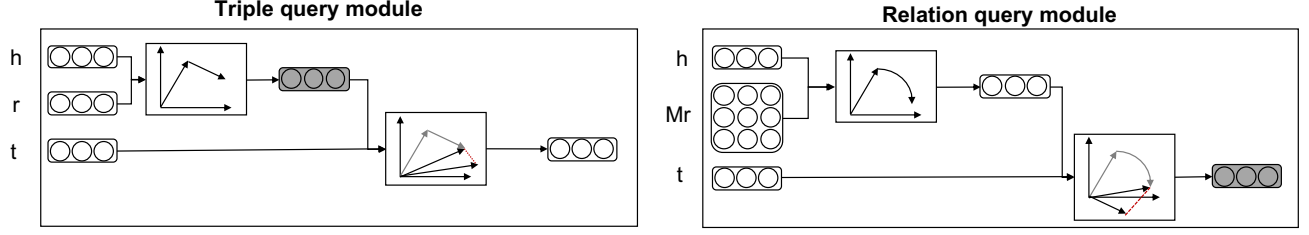


Fig. 1: Pre-trained Knowledge Graph Model.

graph matching knowledge queries. Triple query and relation query are two types of basic queries commonly executed in knowledge graphs. Triple queries are in the following form

```
SELECT ?t
WHERE {h r ?t}
```

where the target is to get the tail entity t given an head entity h and relation r . Relation queries are

```
SELECT ?r
WHERE {h ?r ?t}
```

where the target is to get relations of a given entity h . With entity and relation sets known, combining these two types of queries, we could recover all triples in a knowledge graph.

In PKGM, we build two query modules to simulate information queries of knowledge graphs in continuous vector space. One is Triple Query Module for triple queries, which maintains an assumption to transfer the head entity embedding and relation embedding to tail entity embedding in vector space, expressed as $f_{triple}(\mathbf{h}, \mathbf{r}) \approx \mathbf{t}$. The other one is Relation Query Module for relation queries. It maintains an assumption to show whether an entity should have a relation or not. In vector space, we make zero vector $\mathbf{0}$ represent EXISTING, thus the function in Relation Query Module should be $f_{rel}(\mathbf{h}, \mathbf{r}) \approx \mathbf{0}$ if exists.

A. Triple query module

The purpose of the triple query module is to transfer the head entity and relation to the tail entity and encode the truth value of a triple. Since a lot of knowledge graph embedding methods have been proposed for this task, we apply TransE [1] for its simplicity and effectiveness.

Each $e \in \mathcal{E}$ and $r \in \mathcal{R}$ is embedded as a vector. Following the translation assumption, for each positive triple (h, r, t) , we make $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{h} \in \mathbb{R}^d$, $\mathbf{r} \in \mathbb{R}^d$ and $\mathbf{t} \in \mathbb{R}^d$ are embedding of h, r and t respectively. Score function for (h, r, t) is

$$f_{triple}(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \quad (1)$$

where $\|\mathbf{x}\|$ is the L1 norm of $\mathbf{x}$. It should make $\mathbf{h} + \mathbf{r}$ approach $\mathbf{t}$ if (h, r, t) is positive and far away otherwise.

B. Relation query module

The purpose of relation query module is to encode whether an entity should have one relation or not. We represent the concept of EXIST as $\mathbf{0}$ in vector space. To encode the existence of relation r for h , we design a function $f_{rel}(\mathbf{h}, \mathbf{t})$ and make $f_{rel}(\mathbf{h}, \mathbf{t}) \approx \mathbf{0}$ if h (should) have r and keep $f_{rel}(\mathbf{h}, \mathbf{t})$ far away from $\mathbf{0}$ otherwise.

We devise a transferring matrix for each relation $r \in \mathcal{R}$, denoted as $\mathbf{M}_r$. With $\mathbf{M}_r$ we could transfer $\mathbf{h}$ to $\mathbf{r}$. More specifically, the discrepancy between the transferred head entity representation and relation embedding should be close to $\mathbf{0}$ if h has relation r or should have. And the discrepancy should be far away from $\mathbf{0}$ if h does not have or should not have relation r . Thus f_{rel} is designed as follows

$$f_{rel}(h, r) = \|\mathbf{M}_r \mathbf{h} - \mathbf{r}\| \quad (2)$$

For a positive pair (h, r) , $f_{rel}(h, r)$ should be as small as possible, while be as large as possible for negative ones.

C. Loss function

The final score of the given triple (h, r, t) is

$$f(h, r, t) = f_{triple}(h, r, t) + f_{rel}(h, r) \quad (3)$$

in which positive pairs should have a small score and negative pairs should have a large one. A margin-based loss is set as a training objective for PKGM

$$L = \sum_{(h, r, t) \in \mathcal{K}} [f(h, r, t) + \gamma - f(h', r', t')]_+ \quad (4)$$

where (h', r', t') is a negative triple generated for (h, r, t) by randomly sample an entity $e \in \mathcal{E}$ to replace h or t , or randomly sample a relation $r' \in \mathcal{R}$ to replace r . γ is a hyperparameter representing the margin which should be achieved between positive sample scores and negative sample scores. And

$$[x, y]_+ = \begin{cases} x & \text{if } x \geq y \\ y & \text{if } x < y \end{cases} \quad (5)$$

D. Service

After training, PKGM has (1) good model parameters including entity embeddings, relation embeddings and transferring matrices, (2) functions in triple query module and relation query module. Then it could provide service for triple queries and relation queries in vector space.

TABLE I: Functions for pre-training and servicing stage of PKGM. The subscript letter T and R denotes *triple* and *rel*.

Module	Pre-training	Servicing
Triple	$f_T(h, r, t) = \ \mathbf{h} + \mathbf{r} - \mathbf{t}\ $	$S_T(h, r) = \mathbf{h} + \mathbf{r}$
Relation	$f_R(h, r) = \ \mathbf{M}_r \mathbf{h} - \mathbf{r}\ $	$S_R(h, r) = \mathbf{M}_r \mathbf{h} - \mathbf{r}$

1) *Service for triple queries*: PKGM provides the representation of candidate tail entity, when given a head entity h and a relation r as follows:

$$S_{triple}(h, r) = \mathbf{h} + \mathbf{r} \quad (6)$$

with which $S_{triple}(h, r)$ will approximate to $\mathbf{t}$ if $(h, r, t) \in \mathcal{K}$ as a result of training objective (Equation(4)), and the output of $S_{triple}(h, r)$ will be an entity representation that is most likely to be the right tail entity if there is no triple existing with h as head entity and r as relation in $\mathcal{K}$. This is the inherent completion capability of knowledge graph embedding methods, which has been widely proved and applied for knowledge graph completion [6]. We summarize the function for pre-training and servicing in Table I.

2) *Service for relation queries*: PKGM provides a vector representing whether an entity h has or should have triples about relation r as follows:

$$S_{rel}(h, r) = \mathbf{M}_r \mathbf{h} - \mathbf{r} \quad (7)$$

There are three situations: (1) if h links to entities according to r , then $S_{rel}(h, r)$ will approximate to EXIST embedding $\mathbf{0}$, (2) if h does not link to any entity according to r but it should, then $S_{rel}(h, r)$ will also approximate to EXIST embedding $\mathbf{0}$, (3) if h does not link to any entity according to r and it should not, $S_{rel}(h, r)$ will be away from $\mathbf{0}$. This is the relation completion capability of PKGM gaining from Equation (2).

There are two significant advantages of getting knowledge via PKGM's query services given h and r :

- We could access the tail entity in an implicit way via calculation in vector space without truly querying triples existing in the knowledge graph. This makes query service independent with data and ensures data protection.
- Results for each input pair are uniformed as two vectors from two query modules instead of triple data, making it easier to apply them in downstream use models since designing models to encode triple data is avoided.
- We could get the inferred tail entity t even triple $(h, r, t) \notin \mathcal{K}$, which greatly overcomes the incompleteness disadvantages of KG.

E. Applying service vectors in downstream use models

With service vectors from PKGM, we introduce two general ways to apply them in embedding-based models for downstream use. We categorize embedding-based models into two classes according to the number of input embeddings for an entity in the model, one is inputting a sequence of embeddings and the other one is inputting a single embedding. Given an item as target entity in the context of e-commerce,

service vectors from triple query modules are denoted by $[S_1, S_2, \dots, S_k]$ and those from relation query module denoted by $[S_{k+1}, S_{k+2}, \dots, S_{2k}]$.

The idea of integrating service vectors from PKGM into sequence models is shown in Fig.2. In these models, the

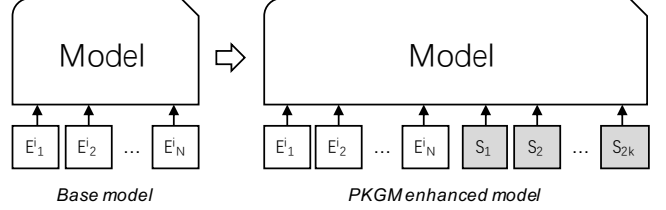


Fig. 2: Application of PKGM for models with sequence embedding inputs.

sequential embedding for one $item_i$ are usually generated according to item side information, like word embedding of description text or labeled features. And these models are capable of automatically capturing interactions between embeddings in the sequence, thus we propose to append all service vectors for $item_i$ from PKGM at the end of the input sequence. The model will make service vectors interact with other input embeddings automatically. Supposing that the original input is $[E_1^i, E_2^i, \dots, E_N^i]$, after appending service vectors, the input will be $[E_1^i, E_2^i, \dots, E_N^i, S_1, S_2, \dots, S_{2k}]$, where we first append service vectors from triple query module and then append those from relation query module.

The idea of integrating service vectors from PKGM into models with single embedding input for an entity is shown in Fig.3. In these models, the single embedding usually refers to the embedding of $item_i$ in current latent vector space and is learnt during training. To keep the informational balance for an entity between embeddings from the original model and our PKGM, we propose to firstly integrate all service vectors for $item_i$ from PKGM into one. During the integration, service vectors corresponding to the same relation from two modules should be considered together, thus we firstly make a concatenation with service vectors from triple query module and relation query module as follows:

$$S'_j = [S_j; S_{j+k}], \text{ where } j \in [1, k] \quad (8)$$

where $[x; y]$ means concatenating vector x and y . Then we integrate them into one as follows:

$$S = \frac{1}{k} \sum_{j \in [1, k]} S_j \quad (9)$$

Then we concatenate S with original item embedding E_i as the single input.

III. EXPERIMENTS

A. Pre-training

1) *Dataset*: We pre-train PKGM on a sub dataset of our billion-scale E-commerce product knowledge graph (PKG) called PKG-sub. PKG has a total number of 1.2B attribute

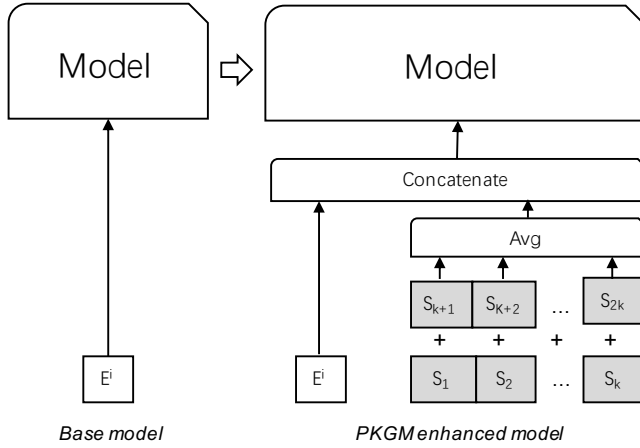


Fig. 3: Application of PKGM for models with single embedding input.

records for 0.2B items, where all the attributes are filled by sellers in Taobao. We process the data into triplet format using Alibaba Maxcompute, a max-reduce framework. We remove the attributes with occurrences less than 5000 in PKG, since these attributes are likely noisy that not only increase largely the model size but also deteriorate the model performance. The statistic details of PKG-sub are shown in Table II.

TABLE II: Statistics of PKG-sub for pre-training.

	# items	# entity	# relation	# Triples
PKG-sub	142,634,045	142,641,094	426	1,366,109,966

For each item $item_i$ in the dataset, we select 10 key relations for it according to its category. More specifically, suppose $item_i$ belongs to category C , we gather all items belonging to C and account for the frequency of properties in those items, then select top 10 most frequent properties as key relations. After pre-training, our PKGM will provide service vectors for target items with regard to those key relations.

2) *Training details*: Our PKGM is implemented with Tensorflow [7] and Graph-learn[§]. Graph-learn [8] is a large-scale distributed framework for node and edge sampling in graph neural networks. We use Graph-learn to perform edge sampling, and 1 negative sample is generated for each edge. For model setting, Adam [9] optimizer is employed with initial learning rate as 0.0001, training batch size as 1000, and embedding dimension as 64. Finally the model size is 88GB, and we train it with 50 parameter servers and 200 workers for 2 epochs. The whole training process consumed 15 hours.

B. Item Classification

1) *Task definition*: The target of item classification is to assign an item to a class in the given class list. Item classification is a common task in our e-commerce platform since

new tags and features are added frequently to items, such as lifestyle and crowd that items are suitable for. For different classification tasks, the candidate class varies. For example, candidate classes for lifestyle classification include Vacation, Yoga, Hiking, and so on, and those for crowd classification include Programmer, Parents, Girlfriend, and so on.

Usually, item titles are used for classification since most items in our platform have a title filled by managers of online shops. Thus we could frame it as a text classification task. Under supervised training, the target is to train a mapping function $f: \mathcal{T} \mapsto \mathcal{C}$ where $\mathcal{T}$ is a set of item titles and $\mathcal{C}$ is a set of classes. For each title $t \in \mathcal{T}$, $t = [w_1, w_2, w_3, \dots, w_n]$ is an ordered sequence composed of words.

2) *Model*: **Base model** Text classification is an important task in Natural Language Processing (NLP) and text mining, and many methods have been proposed for it. In recent years, the deep learning model has been shown to outperform traditional classification methods [2], [10]. Given the input document, the mapping function f firstly learns a dense representation of it via representation learning and then uses this representation to perform final classification. Recently, large-scale pre-trained language models, such as ELMo [11], GPT [12] and BERT [2], have become the *de facto* first representation learning step for many natural language processing tasks. Thus we apply BERT as the base model for item classification.

BERT is a pre-trained language model with multi-layers of bidirectional encoder representations from Transformers, trained with unlabeled text by jointly conditioning on both left and right context in all layers. BERT is trained on a huge amount of texts with masked language model objective. Since the use of BERT has become common and we conduct item classification experiments with released pre-trained BERT by Google. We will omit an exhaustive background description of the model architecture and refer the readers to [2] as well as the excellent guides and codes[§] of fine-tuning BERT on downstream tasks.

We show details of applying BERT on item classification task in Figure 4. We input the text sequence of title into BERT and take out the representation corresponds to [CLS] symbol C . To predict the class of items, we input the [CLS] representation into a fully connected layer as follows:

$$p = \sigma(\mathbf{W}\mathbf{C} + \mathbf{b}) \quad (10)$$

where $\mathbf{W} \in \mathbb{R}^{d \times n_c}$ is a weighed matrix in the shape of $d \times n_c$. d is the word embedding dimension, also called hidden size, of BERT and n_c is the number of classes in current task. $\mathbf{b} \in \mathbb{R}^{n_c}$ is a bias vector and $\sigma()$ is an activate function.

With predicted p and input label $l \in \mathbb{R}^{n_c}$, we fine-tune the model with a cross-entropy loss. We assign an unique identity to each class beginning with 0. For an item belonging to the i th class, $l[i] = 1$ and others are 0.

Base+PKGM model To enhance item classification with our PKGM, for each item, we provide service with k pre-

[§]<https://github.com/alibaba/graph-learn>

[§]<https://github.com/google-research/bert>

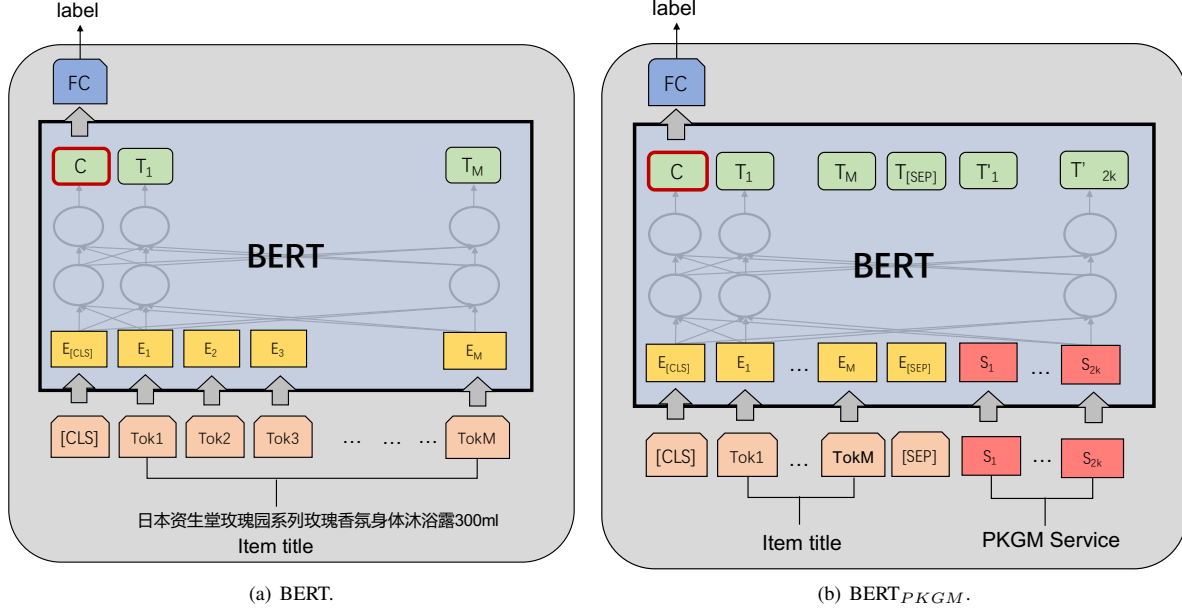


Fig. 4: Models for item classification.

sentations from both triple query module and relation query module, $2 \times k$ in total. We first add a [SEP] symbol at the end of the title text and then input the $2 \times k$ service vectors from PKGM into BERT, and we named it as *Base_{PKGM-all}*. Embedding look up is unnecessary for service vectors and they are directly appended after the last token embedding of the input sentence. Details are shown in Figure 4.

We also fine-tune the Base+PKGM model with only k triple query module representations to explore how much of a role the *triple query service* plays in the model. The final k title texts are replaced with service vectors from triple query module, and we abbreviate this model as *Base_{PKGM-T}*. In the same way, *Base_{PKGM-R}* refers to replacing final k title texts with k service vectors from relation query module.

3) *Dataset*: In this experiment, we take item categories as target classes. In our platform, each item is assigned to at least one category, and we maintain an item category tree to better organize billions of items and help recall items for a lot of business, like selecting items for the online market in a special event like Double Eleven. Thus we always try to assign at least one category for each item.

Experiment dataset contains 1293 categories. Details are shown in Table III. The number of positive samples and negative samples are 1 : 1. To show the power of pre-trained models, both language model and knowledge graph model, with which we could get good performance on downstream tasks with a few training samples, we constrain the instance of each category less than 100 during data preparation.

TABLE III: Data statistic for item classification task

	# category	# Train	# Test	# Dev
dataset	1293	169039	36225	36223

4) *Experiment details*: We take the pre-trained BERT_{BASE} model trained on Chinese simplified and traditional corpus[§], whose number of layers is 12, hidden size is 768 and the number of attention heads is 12.

Following BERT, we add a special classification token [CLS] ahead of the input sequence, which is used as the aggregate sequence representation in the final hidden state. In this experiment, we set the length of the sequence of titles to 128. For titles shorter than 127 (with [CLS] symbol excepted), we padding it with several zero embeddings to the required length as did in BERT, while for titles longer than 127, we reserve the first 127 words.

For BERT_{PKGM-all} model, we replace the last $2 \times k$ embeddings with service vectors from PKGM, and for BERT_{PKGM-T} model and BERT_{PKGM-R}, we replace the last k embeddings with service vectors from triple query module and relation query module respectively.

We fine-tune BERT with 3 epochs with batch size as 32 and learning rate as $2e^{-5}$. All parameters in BERT are unfix and representations from PKGM fixed during fine-tune.

To evaluate performances, we report the accuracy of item classification in TABLE IV. We abbreviate the accuracy as AC and it is reported as the percentage from the validation dataset. Because the amount of categories is too large compared with

[§]Released at https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

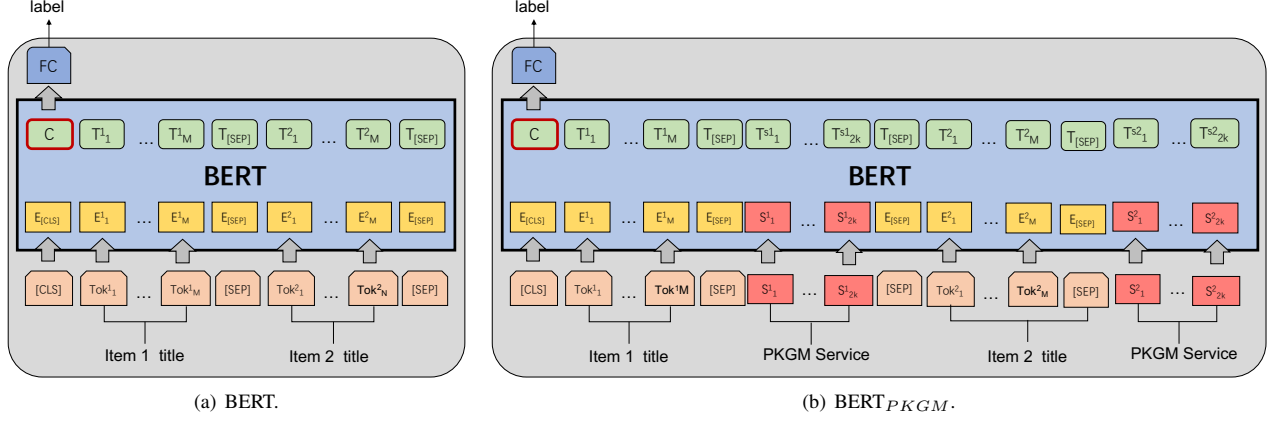


Fig. 5: Models for item alignment.

other traditional classification tasks, we also report metrics $Hit@k$ ($k = 1, 3, 10$) to describe the validity of the experimental results more accurately. We calculate $Hit@k$ by getting the rank of the correct label as its predicting category rank and $Hit@k$ is the percentage of test samples with prediction rank within k .

TABLE IV: Results for item classification task

Method	Hit@1	Hit@3	Hit@10	AC
BERT	71.03	84.91	92.47	71.52
BERT _{PKGM-T}	71.26	85.76	93.07	72.14
BERT _{PKGM-R}	71.55	85.43	92.86	72.26
BERT _{PKGM-all}	71.64	85.90	93.17	72.19

5) *Results*: The results are shown in Table IV, showing that BERT_{PKGM} outperforms BERT on all of these three datasets, on both $Hit@k$ and prediction accuracy metrics. Specifically, BERT_{PKGM-all} has the best performance of $Hit@1$, $Hit@3$ and $Hit@10$ and BERT_{PKGM-R} achieves best accuracy. These results demonstrate the effectiveness of information from PKGM, among which relation query service vectors contributes more useful information than triple query service vectors.

Besides, BERT_{PKGM-R} sometimes outperforms BERT_{PKGM-all}, which violates the idea that more features matter. We think it's probably because words in the original title play a more important role in the item classification task than service vectors provided from PKGM since less words are replaced in BERT_{PKGM-R} than BERT_{PKGM-all}.

C. Product Alignment

1) *Task definition*: The target of the item alignment task is to find two items referring to the same product. For example, there are many online shops selling iPhoneXI with Green color and 256 GB capacity. They are different items on the platform while from the perspective of the product, they refer to the same one. Detecting items belonging to the same product contributes to daily business greatly, for example,

recommending aligned items of items that a user is reviewing, which help users deeply compare the prices, services after selling, and so on. More importantly, the number of items are much larger than the number of products, organizing items from the perspective of the product helps reduce the load of data management and mining.

Similar to item classification, item titles are used for alignment. And item alignment task could be framed as a paraphrase identification task, to distinguish whether two input sentences express the same meaning [13]. In the context of the item alignment, the target is to distinguish whether two input item titles describe the same product. The prevailing approach of training and evaluating paraphrase identification models are constructed as a binary classification problem: the model is given a pair of sentences and is judged by how accurately it classifies pairs as either paraphrases or non-paraphrases. Thus a mapping function $f: \mathcal{P}_T \mapsto \mathcal{C}$ will be learned, in which $\mathcal{P}_T$ is a set of item title pairs and $\mathcal{T}$ is a set of item title. $\mathcal{C}$ contains binary classes *True* and *False*. Similar to item classification, for each item title $t \in \mathcal{T}$, $t = [w_1, w_2, w_3, \dots, w_n]$ is an ordered sequence composed of words.

2) *Model*: **Base model** Paraphrase identification is a well-studied sentence pair modeling task and is very useful for many NLP applications such as machine translation(MT) [14], question answering(QA) [15] and information retrieval(IR) [16]. Many methods have been proposed for it in recent years including pairwise word interaction modeling with deep neural network system [17], character level neural network model [18], and pre-trained language model [2]. Since the pre-trained language models have reached state-of-the-art performance on paraphrase identification tasks, we apply BERT as the base model for the same alignment.

We show the details of applying BERT on the item alignment in Figure 5. The title of two items are input into BERT as a sentence pair and we take the representation corresponding to [CLS] symbol as the representation of the sentence pair which will be used for binary classification with a fully connected layer, as done in item classification task.

Base+PKGM For each item, we provide service with $2 \times k$ vectors from the triple query module and relation query module of PKGM to enhance the item alignment task with item knowledge. Similar to the operation in item classification, we add a [SEP] symbol at the end of each title text and $4 \times k$ service vectors are added to BERT finally. After that, we concatenate two-sentence input together. Details are shown in Figure 5. Similar to item classification, with all service vectors applied, we call the model as $Base_{PKGM-all}$. $Base_{PKGM-T}$ and $Base_{PKGM-R}$ are defined in the same way which refer to input the k service vectors from triple query module and relation query module in PKGM respectively. In this task, *Base* refers to BERT.

3) *Dataset*: We experiment on 3 datasets containing different types of items. In our platform, item alignment is only necessary to distinguish items with the same type since items with different types refer to different products for sure. The first dataset is generated from skirts for girls, the second one from hair decorations and the last one from children's socks. In each dataset, one data sample includes two item titles and a label, in which samples belonging to the same products are labeled with 1 and 0 otherwise. There are less than 10 thousand training samples in each dataset and we keep the number of item pairs in train/test/dev dataset as 7 : 1.5 : 1.5. Details of these datasets are shown in Table V.

TABLE V: Data statistic for item alignment task.

	# Train	# Test-C	# Dev-C	# Test-R	# Dev-R
category-1	4731	1014	1013	513	497
category-2	2424	520	519	268	278
category-3	3968	852	850	417	440

4) *Experiment details*: Consistent with the previous item classification task, we take the same pre-trained BERT_{BASE} model and the same input format, except for some difference in the input data. The input embedding sequence consists of two different items. A [CLS] symbol is added in the beginning of the title sequence and one [SEP] symbol is appended at the last position of each title sequence to differential these two items. Since the whole sequence length is 128, the length of each item sequence is restricted within 63 and we adopt the same method to formalize the item sequence with item classification task.

We report prediction accuracy and metrics $Hit@k$ ($k = 1, 3, 10$) to evaluate the performance on item alignment. For metric $Hit@k$, we generate n negative triples by replacing one of the aligned items randomly and together with the original aligned pairs, we will get $n + 1$ prediction results. Afterward, we rank $n + 1$ prediction probabilities in ascend order and get the rank of the aligned pairs as its prediction rank. $Hit@k$ is the percentage of test samples with prediction rank within k . In this case, the number of negative triples for each aligned pair is 99 and in other words, we get prediction rank from 100 candidate samples.

5) *Results*: Table VI shows the rank results for item alignment. BERT_{PKGM-all} model outperforms BERT model

on $Hit@10$ metrics on all three datasets and it has the best performance at both category-2 and category-3 with all $Hit@k$ metrics. These results show the effectiveness of PKGM and it promotes the accuracy of the prediction better. In the meantime, BERT model has a weak advantage over BERT_{PKGM-all} model at the category-1 on $Hit@1$ metrics and this could probably be due to the larger amount of data in this category. To a certain extent, the title text with enough training examples could make the model learn item information better, while the PKGM could play a greater role in the case of a small amount of data relatively.

Table VII shows the accuracy results. Obviously, BERT_{PKGM-all} has the best performance on all datasets and it convincingly demonstrates the PKGM could promote the effect on item alignment task.

TABLE VI: Hit@k results for item alignment.

Method	dataset	Hit@1	Hit@3	Hit@10
BERT	category-1	65.06	76.06	86.68
BERT _{PKGM-all}		64.75	77.50	87.43
BERT	category-2	65.86	78.07	87.59
BERT _{PKGM-all}		66.13	78.19	87.96
BERT	category-3	49.64	66.18	82.37
BERT _{PKGM-all}		50.60	67.14	83.45

D. Item Recommendation

1) *Task definition*: Item recommendation aims to properly recommend items to users that users will interact with a high probability. As implicit feedback (click, buy, etc) is more commonly observed in modern e-commerce platforms, we consider only implicit feedback for recommending items to corresponding users with the help of service vectors provided by PKGM. To learn user preference and make proper recommendations from observed implicit user feedback, we treat it as a ranking problem like described in [19].

2) *Model*: **Base model** We make use of NCF [19] as a general framework of our base model. In NCF, GMF layer and MLP layer are used for modeling item-user interactions, in which GMF uses a linear kernel to model the latent feature interactions, and MLP uses a non-linear kernel to learn the interaction function from data. There are 4 layers in NCF including input layer, embedding layer, neural CF layer, and output layer.

The input layer contains user and item sparse feature vectors $\mathbf{v}_u^U$ and $\mathbf{v}_i^I$ that describe user u and item i respectively. Since we conduct the downstream recommendation experiment based on a pure collaborative setting, we make $\mathbf{v}_u^U$ and

TABLE VII: Accuracy results for item alignment.

	category-1	category-2	category-3
BERT	88.94	89.31	86.94
BERT _{PKGM-T}	88.65	89.89	87.88
BERT _{PKGM-R}	89.09	89.60	87.88
BERT _{PKGM-all}	89.15	90.08	88.13

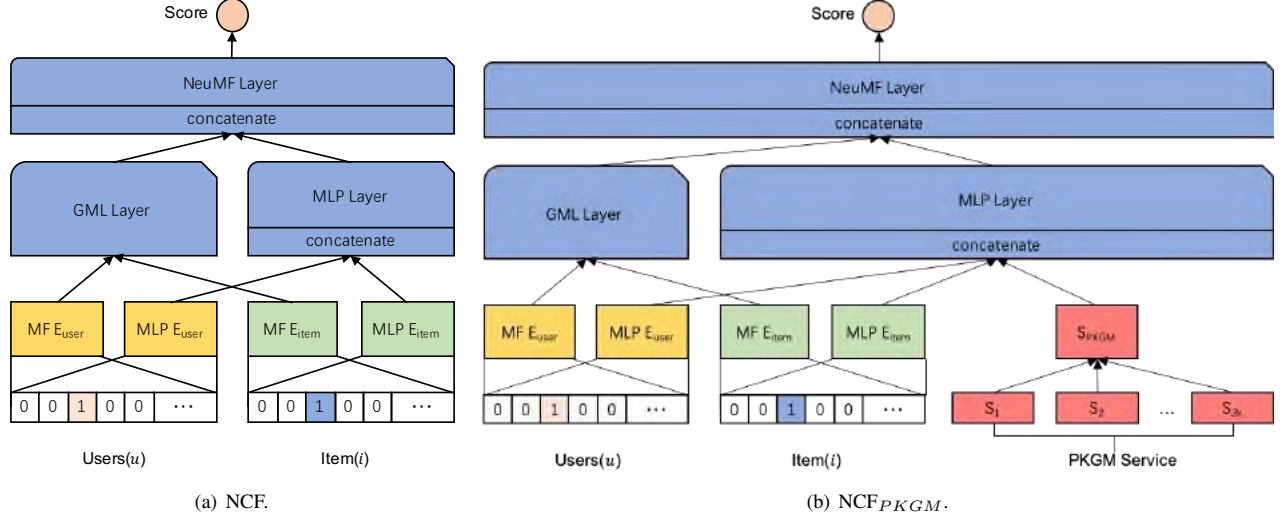


Fig. 6: Models for item recommendation.

$\mathbf{v}_i^I$ one-hot sparse vector generated according to the identity of u and i in the dataset.

Above the input layer, there is an embedding layer which is a fully connected layer to project the sparse input representation to a dense vector, called embedding. User embedding and item embedding could be regarded as the latent vector for users and items in the context of latent factor models.

$$\mathbf{p}_u = \mathbf{P}^T \mathbf{v}_u^U, \mathbf{q}_i = \mathbf{Q}^T \mathbf{v}_i^I \quad (11)$$

Then user embedding $\mathbf{p}_u$ and item embedding $\mathbf{q}_i$ are fed into a neural collaborative filtering layer, to map the latent vectors to prediction scores. Thus the NCF predictive model could be formulated as

$$\hat{y}_{ui} = f_{NCF}(\mathbf{p}_u, \mathbf{q}_i | \mathbf{P}, \mathbf{Q}, \Theta_f), \quad (12)$$

where $\mathbf{P} \in \mathbb{R}^{M \times K}$ and $\mathbf{Q} \in \mathbb{R}^{N \times K}$ are embedding matrix for users and items respectively. And Θ_f is a set of parameters in function f_{NCF} . f_{NCF} is a user-item interaction function that makes $\mathbf{p}_u$ and $\mathbf{q}_i$ interact and influence with each other and finally gives a score for the input item-user pair. And there are two ways to make such interactions, one is matrix factorization (MF) and the other one is multi-layer perceptron (MLP).

MF is the most popular model for recommendation and has been investigated extensively in literature [20]. The mapping function in neural generalized matrix factorization (GMF) layer is

$$\phi^{GMF}(\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u \circ \mathbf{q}_i, \quad (13)$$

where $\circ$ denotes the element-wise product of vectors.

Using two pathways to model users and items has been widely adopted in multimodal deep learning models [21], [22]. Thus in the MLP interaction layer, we first make a vector concatenation of $\mathbf{p}_u$ and $\mathbf{q}_i$

$$\mathbf{z}_1 = \phi_1^{MLP}(\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix} \quad (14)$$

then we add hidden layers on the concatenated vector, using a standard MLP to learn the interaction between user and item latent features. This endows the model a large level of flexibility and non-linearity to learn the interactions between $\mathbf{p}_u$ and $\mathbf{q}_i$

$$\phi_2^{MLP}(\mathbf{z}_1) = a_2(\mathbf{W}_2^T \mathbf{z}_1 + \mathbf{b}_2) \quad (15)$$

$$\dots \quad (16)$$

$$\phi_L^{MLP}(\mathbf{z}_{L-1}) = a_L(\mathbf{W}_L^T \mathbf{z}_{L-1} + \mathbf{b}_L) \quad (17)$$

where $\mathbf{W}_x$, $\mathbf{b}_x$ and a_x denote the weight matrix, bias vector, and activation function for the x -th layer's perceptron, respectively.

So far we have two interaction vectors for the input user and item embeddings, then we fuse these two vectors as follows:

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix}) \quad (18)$$

The objective function to minimize in NCF method is as follows

$$L = - \sum_{(u,i) \in \mathcal{Y}} y_{ui} \log(\hat{y}_{ui}) + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \quad (19)$$

where y_{ui} is the label for input user u and item i , and if they have interaction in the dataset $y_{ui} = 1$ and $y_{ui} = 0$ otherwise. $\mathcal{Y}$ is the training set containing positive pairs and negative pairs. Since the original dataset only contains positive pairs, we take the same strategy used in [19] to generate negative pairs. Negative pairs are uniformly sampled from unobserved interactions for each positive pair, and replace the positive item with negative items that are unobserved. We refer negative sampling ratio as the number of negative pairs for each positive pair.

Base+PKGM Since NCF is built based on item embeddings, we integrate vector services from PKGM with item embeddings in the second way as introduced before. Specifically,

TABLE VIII: Results for item recommendation.

	HR@1	HR@3	HR@5	HR@10	HR@30	NDCG@1	NDCG@3	NDCG@5	NDCG@10	NDCG@30
NCF	27.94	44.26	52.16	62.88	81.26	0.2794	0.3744	0.4069	0.4415	0.4853
NCF _{PKGM-T}	27.96	44.83	52.43	63.51	81.62	0.2796	0.3778	0.4091	0.4449	0.4880
NCF _{PKGM-R}	31.01	47.99	56.10	66.98	84.73	0.3101	0.4091	0.4424	0.4777	0.5200
NCF _{PKGM-all}	30.76	47.92	55.60	66.84	84.71	0.3076	0.4079	0.4395	0.4758	0.5185

for each user-item pair, PKGM with provide $2 \times k$ vectors for the item, denoted as $[S_1, S_2, \dots, S_{2k}]$. We first combine these vectors into one as follows

$$S_{PKGM} = \frac{1}{k} \sum_{i \in [1, 2, \dots, k]} [S_i; S_{i+k}] \quad (20)$$

then we integrate S_{PKGM} into MLP layer by making $\mathbf{z}_1$ as a concatenation results of three vectors

$$\mathbf{z}_1 = \phi_1^{MLP}(\mathbf{p}_u, \mathbf{q}_i, S_{PKGM}) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \\ S_{PKGM} \end{bmatrix}. \quad (21)$$

Other parts of NCF stay the same. Fig. 6 shows the general framework of NCF and NCF_{PKGM}.

3) *Dataset*: We conduct experiments on real data sampled from records on Taobao platform. The statistics of our sample data are shown in TABLE IX. There are 29015 users and 37847 items with 443425 interactions between them. The number of interactions for each user is assured of being more than 10. Each interaction record is a user-item pair representing that the user has an interaction with the item.

TABLE IX: Data statistics for item recommendation.

	# Items	# Users	# Interactions
TAOBAO-Recommendation	37847	29015	443425

4) *Experiment details*: The performance of item recommendation is evaluated by the *leave-one-out* strategy which is used in [19]. For each user, we hold-out the latest interaction as the test set and others are used in the train set. As for the testing procedure, we uniformly sample 100 unobserved negative items, ranking the positive test item with negative ones together. In this way, *Hit Ratio*(HR)@k and *Normalized Discounted Cumulative Gain*(NDCG)@k are used as the final metrics of ranking where $k \in \{1, 3, 5, 10, 30\}$. Both metrics are computed for each test user, and average scores are regarded as final results.

We randomly sample one interaction for each user and it is treated as a validation sample to determine the best hyper-parameters of the model. For the GMF layer part, the dimension of user embedding and item embedding are both set to 8. As for MLP layers, the dimension of user embedding and item embedding are both set to 32. Three hidden layers of size [32, 16, 8] after embedding layers are used for both NCF and NCF_{PKGM}. For NCF_{PKGM}, there are external PKGM features fed to concatenate with MLP

user embedding and MLP item embedding. The model is optimized with loss defined in Equation (19) with an external $L2$ regularization on user and item embedding layer in MLP and GMF. Regularization factor λ is chosen as 0.001. Mini-batch Adam [9] is used for model training with a batch size of 256 and the total number of epochs is 100. Learning rate is set as 0.0001. And the dimension of the prediction layer is 16, which consists of two partial outputs of 8 and 8 from GMF layer and MLP layer respectively. In this work, we use a negative sampling ratio of 4. For simplicity and effectiveness, we use the non-pre-train version of NCF model for base model and NCF_{PKGM} as PKGM featured model.

5) *Results*: Experiment results are shown in TABLE VIII. From the result we can see that:

Firstly, all of the PKGM feature augmented models outperform the NCF baseline in all metrics. For NCF_{PKGM-T} model, it outperforms the NCF baseline with an average promotion of 0.37% on Hit Ratio metrics and 0.0023 on NDCG metrics. For NCF_{PKGM-R} model, it outperforms the NCF baseline with an average promotion of 3.66% on Hit Ratio metrics and 0.0343 on NDCG metrics. For NCF_{PKGM-all} model, it outperforms the NCF baseline with an average promotion of 3.47% on Hit Ratio metrics and 0.0324 on NDCG metrics. This promotion illustrates that features learned by our PKGM model provide external information that are not included in original user-item interaction data and it is useful for downstream tasks like e-commerce recommendation.

Secondly, the key information of different service vectors provided by PKGM model are different. Performances on NCF_{PKGM-R} are better than those on NCF_{PKGM-T}. Thus the information from relation query module is relatively more useful than the information from PKGM-triple query module in item recommendation scenario, which is largely due to the fact that properties are more effective than entities and values when modeling user-item interaction.

IV. RELATED WORK

A. Knowledge Graph Embedding

With the introduction of embedding entities and relations of the knowledge graph into continuous vector space by TransE [1], the method of Knowledge Graph Embedding (KGE) has made great progress in knowledge graph completion and application. The key idea that transforming entities and relations in triple (h, r, t) into continuous vector space aims to simplify the operation in KG. TransH [23] proposes that an entity has different representations under different relationships to overcome the limitations within multi-relation problems. TransR [24] introduces relation-specific spaces other

than entity embedding spaces. TransD [25] decomposes the projection matrix into a product of two vectors. TransSparse [26] simplifies TransR by enforcing sparseness on the projection matrix. MuRP [27] and ATTH [28] lead hyperbolic geometry into embedding learning of knowledge graph to capture logical hierarchies and patterns. All of these models try to achieve better effects on multi-relation problems.

Different from the translational distance models mentioned above, the semantic matching model uses a similarity based scoring function. They measure the credibility of facts by matching the latent semantics of entities and the relationships, methods include RESCAL [29], DistMult [30], ComplEx [31] and ConvE [32], etc.

The knowledge graph embedding method learns representations for entities and relations that preserve the information contained in the graph. The method has been applied to various tasks and applications and achieved good results in the field of the knowledge graph, including link prediction [33], rule learning [34], relation extraction [35], and entity alignment [36]. We adopt the representation learning method in the billion-scale-commerce product knowledge graph to calculate and to evaluate the relationship among items.

B. Pre-train Language Models.

The pre-trained language models have achieved excellent results in many NLP tasks and bring convenience in completing various specific downstream tasks, such as BERT [2], XLNet [37], and GPT-3 [38]. Recently, more and more researchers pay attention to the combination of knowledge graph and pre-trained language model, to enable PLMs to reach better performance with the awareness of knowledge information.

K-BERT [39] injects knowledge triples into a sentence to generate unified knowledge-rich language representation. ERNIE [40] integrates entity representations from the knowledge module into the semantic module to represent heterogeneous information of tokens and entities into a united feature space. KEPLER [41] encodes textual descriptions for entities as text embeddings and treats the description embeddings as entity embeddings. KnowBert [42] uses an integrated entity linker to generate knowledge enhanced entity-span representations via a form of word-to-entity attention. K-Adapter [43] takes RoBERTa as the pre-trained model and injects factual knowledge and linguistic knowledge with a neural adapter for each kind of infused knowledge. DKPLM [44] could dynamically select and embed knowledge context according to the textual context for PLMs, with the awareness of both global and local KG information. JAKET [45] proposes a joint pre-training framework, including the knowledge module to produce embeddings for entities in text and language modules to generate context-aware embeddings in the graph. What's more, KALM [46], ProQA [47], LIBERT [48] and other researchers explore the fusion experiment with knowledge graph and pre-trained language module in different application tasks.

Based on BERT or other pre-trained language models, these methods aim at adjusting the language model by injecting

knowledge bases or completing knowledge graphs from textual representation. Whereas, we pre-train our model on the structural and topological information of a knowledge graph without language model and apply the model into several downstream tasks in e-commerce knowledge.

V. CONCLUSION AND FUTURE WORK

In this work, we present our experience of pre-training knowledge graph to provide knowledge services for other tasks on the billion-scale e-commerce product knowledge graph. We propose Pre-trained Knowledge Graph Mode (PKGM) with two modules to handle triple queries and relation queries in vector space, which could provide item knowledge services in a uniform way for embedding-based models without accessing triple data in KG. We also propose general ways of applying service vectors in different downstream task models. With three types of downstream tasks, we prove that PKGM successfully improves their performance. In future work, we would like to apply PKGM to more downstream tasks and explore other candidate ways to apply service vectors.

VI. ACKNOWLEDGEMENTS

This work is funded by NS-FCU19B2027/91846204/61473260, national key research program 2018YFB1402800 and supported by Alibaba Group through Alibaba Innovative Research Program.

REFERENCES

- [1] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013, pp. 2787–2795.
- [2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*. Association for Computational Linguistics, 2019, pp. 4171–4186.
- [3] H. Tian, C. Gao, X. Xiao, H. Liu, B. He, H. Wu, H. Wang, and F. Wu, "SKEP: sentiment knowledge enhanced pre-training for sentiment analysis," in *ACL*. Association for Computational Linguistics, 2020, pp. 4067–4076.
- [4] C. Alt, M. Hübner, and L. Hennig, "Fine-tuning pre-trained transformer language models to distantly supervised relation extraction," in *ACL (1)*. Association for Computational Linguistics, 2019, pp. 1388–1398.
- [5] O. Melamud, M. A. Bornea, and K. Barker, "Combining unsupervised pre-training and annotator rationales to improve low-shot text classification," in *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 2019, pp. 3882–3891.
- [6] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications," *CoRR*, vol. abs/2002.00388, 2020.
- [7] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *OSDI*. USENIX Association, 2016, pp. 265–283.
- [8] R. Zhu, K. Zhao, H. Yang, W. Lin, C. Zhou, B. Ai, Y. Li, and J. Zhou, "Aligraph: a comprehensive graph neural network platform," *Proceedings of the VLDB Endowment*, vol. 12, no. 12, pp. 2094–2105, 2019.
- [9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.
- [10] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *ACL (1)*. Association for Computational Linguistics, 2018, pp. 440–450.

- [11] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*. Association for Computational Linguistics, 2018, pp. 2227–2237.
- [12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.
- [13] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *COLING*, 2004.
- [14] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, L. Kaiser, N. Kalchbrenner, N. Parmar *et al.*, "Tensor2tensor for neural machine translation," *arXiv preprint arXiv:1803.07416*, 2018.
- [15] W. Yih, M. Chang, X. He, and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *ACL (1)*. The Association for Computer Linguistics, 2015, pp. 1321–1331.
- [16] A. Berger and J. Lafferty, "Information retrieval as statistical translation," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 219–226.
- [17] H. He and J. J. Lin, "Pairwise word interaction modeling with deep neural networks for semantic similarity measurement," in *HLT-NAACL*. The Association for Computational Linguistics, 2016, pp. 937–948.
- [18] W. Lan and W. Xu, "Character-based neural networks for sentence pair modeling," in *NAACL-HLT (2)*. Association for Computational Linguistics, 2018, pp. 157–163.
- [19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *WWW*. ACM, 2017, pp. 173–182.
- [20] S. Rendle, "Factorization machines," in *ICDM*. IEEE Computer Society, 2010, pp. 995–1000.
- [21] H. Zhang, Y. Yang, H. Luan, S. Yan, and T. Chua, "Start from scratch: Towards automatically identifying, modeling, and naming visual attributes," in *ACM Multimedia*. ACM, 2014, pp. 187–196.
- [22] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *ACM Multimedia*. ACM, 2014, pp. 627–636.
- [23] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *AAAI*. AAAI Press, 2014, pp. 1112–1119.
- [24] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *AAAI*. AAAI Press, 2015, pp. 2181–2187.
- [25] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Beijing, China, Volume 1: Long Papers*, 2015, pp. 687–696.
- [26] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA*, 2016, pp. 985–991.
- [27] I. Balazevic, C. Allen, and T. M. Hospedales, "Multi-relational poincaré graph embeddings," in *NeurIPS*, 2019, pp. 4465–4475.
- [28] I. Chami, A. Wolf, D. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *ACL*. Association for Computational Linguistics, 2020, pp. 6901–6914.
- [29] M. Nickel, V. Tresp, and H. Kriegel, "A three-way model for collective learning on multi-relational data," in *ICML*. Omnipress, 2011, pp. 809–816.
- [30] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, 2015.
- [31] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR.org, 2016, pp. 2071–2080.
- [32] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," *arXiv preprint arXiv:1707.01476*, 2017.
- [33] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *WSDM*. ACM, 2019, pp. 96–104.
- [34] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, and H. Chen, "Iteratively learning embeddings and rules for knowledge graph reasoning," in *WWW*. ACM, 2019, pp. 2366–2377.
- [35] G. Wang, W. Zhang, R. Wang, Y. Zhou, X. Chen, W. Zhang, H. Zhu, and H. Chen, "Label-free distant supervision for relation extraction via knowledge graph embedding," in *EMNLP*. Association for Computational Linguistics, 2018, pp. 2246–2255.
- [36] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding," in *IJCAI*, vol. 18, 2018, pp. 4396–4402.
- [37] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *CoRR*, vol. abs/1906.08237, 2019. [Online]. Available: <http://arxiv.org/abs/1906.08237>
- [38] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.
- [39] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-bert: Enabling language representation with knowledge graph," *arXiv:1909.07606*, 2019.
- [40] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," *arXiv preprint arXiv:1905.07129*, 2019.
- [41] X. Wang, T. Gao, Z. Zhu, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *arXiv:1911.06136*, 2019.
- [42] M. E. Peters, M. Neumann, I. Logan, L. Robert, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, "Knowledge enhanced contextual word representations," *arXiv preprint arXiv:1909.04164*, 2019.
- [43] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, C. Cao, D. Jiang, M. Zhou *et al.*, "K-adapter: Infusing knowledge into pre-trained models with adapters," *arXiv preprint arXiv:2002.01808*, 2020.
- [44] Y. Su, X. Han, Z. Zhang, P. Li, Z. Liu, Y. Lin, J. Zhou, and M. Sun, "Contextual knowledge selection and embedding towards enhanced pre-trained language models," *arXiv preprint arXiv:2009.13964*, 2020.
- [45] D. Yu, C. Zhu, Y. Yang, and M. Zeng, "Jaket: Joint pre-training of knowledge graph and language understanding," *arXiv preprint arXiv:2010.00796*, 2020.
- [46] C. Rosset, C. Xiong, M. Phan, X. Song, P. Bennett, and S. Tiwary, "Knowledge-aware language model pretraining," *arXiv preprint arXiv:2007.00655*, 2020.
- [47] W. Xiong, H. Wang, and W. Y. Wang, "Progressively pretrained dense corpus index for open-domain question answering," *arXiv preprint arXiv:2005.00038*, 2020.
- [48] A. Lauscher, I. Vulić, E. M. Ponti, A. Korhonen, and G. Glavaš, "Informing unsupervised pretraining with external linguistic knowledge," *arXiv preprint arXiv:1909.02339*, 2019.

SEMI: A Sequential Multi-Modal Information Transfer Network for E-Commerce Micro-Video Recommendations

Chenyi Lei^{1,2}, Yong Liu³, Lingzi Zhang^{2,4}, Guoxin Wang², Haihong Tang²,
Houqiang Li¹, Chunyan Miao⁴

¹University of Science and Technology of China, Hefei, Anhui, China

²Alibaba Group, Hangzhou, Zhejiang, China

³Alibaba-NTU Singapore Joint Research Institute, Singapore

⁴School of Computer Science and Engineering, Nanyang Technological University, Singapore

leichy@mail.ustc.edu.cn, stephenliu@ntu.edu.sg, lingzi001@e.ntu.edu.sg, {xiaogong.wgx, piaoxue}@taobao.com

lihq@ustc.edu.cn, ascymiao@ntu.edu.sg

ABSTRACT

The micro-video recommendation system becomes an essential part of the e-commerce platform, which helps disseminate micro-videos to potentially interested users. Existing micro-video recommendation methods only focus on users' browsing behaviors on micro-videos, but ignore their purchasing intentions in the e-commerce environment. Thus, they usually achieve unsatisfied e-commerce micro-video recommendation performances. To address this problem, we design a **SE**quential **M**ulti-modal **I**nformation transfer network (SEMI), which utilizes product-domain user behaviors to assist micro-video recommendations. SEMI effectively selects relevant items (*i.e.*, micro-videos and products) with multi-modal features in the micro-video domain and product domain to characterize users' preferences. Moreover, we also propose a **C**ross-domain **C**ontrastive **L**earning (CCL) algorithm to pre-train sequence encoders for modeling users' sequential behaviors in these two domains. The objective of CCL is to maximize a lower bound of the mutual information between different domains. We have performed extensive experiments on a large-scale dataset collected from Taobao, a world-leading e-commerce platform. Experimental results show that the proposed method achieves significant improvements over the state-of-the-art recommendation methods. In addition, the proposed method has also been deployed on Taobao, and the online A/B testing results further demonstrate its practical value.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Micro-video recommendation; cross-domain recommendation; sequential recommendation

ACM Reference Format:

Chenyi Lei^{1,2}, Yong Liu³, Lingzi Zhang^{2,4}, Guoxin Wang², Haihong Tang², Houqiang Li¹, Chunyan Miao⁴. 2018. SEMI: A Sequential Multi-Modal Information Transfer Network for E-Commerce Micro-Video Recommendations. In *Woodstock '18: ACM Symposium on Neural Gaze Detection*, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Micro-videos are an emerging type of content on e-commerce platforms. They focus on introducing merchandise in various video display forms, *e.g.*, opinion leaders' analyses and syntheses, product experience sharing from grass-root users, and sitcom advertisements crafted by product providers or merchants. For e-commerce platforms, micro-videos have the following crucial advantages. Firstly, providing users with high-quality micro-videos enables different pan-entertainment experiences, which significantly increases the user stickiness to the platform. Secondly, well-created micro-videos would attract users' interests in the related merchandise and inspire their shopping behaviors. Thirdly, micro-videos provide a high-quality channel for merchants to perform brand promotion and online marketing. On Taobao, a world-leading e-commerce platform, there are more than 100 million micro-videos, and new micro-videos are generated at the speed of hundreds of thousands per day. More than 30 million users browse micro-videos every day, which leads to an estimated gross merchandise volume of million dollars. Accordingly, there is a great need for e-commerce providers to build effective micro-video recommendation systems to enhance the user experience, engagement, and retention.

Compared to videos on traditional online video sharing platforms, *e.g.*, YouTube and Youku, micro-videos usually have a much shorter length, *e.g.*, tens of seconds. A user can interact with more micro-videos at a lower decision cost. Therefore, users' interaction behaviors with micro-videos usually follow more dynamic patterns than their behaviors with traditional online videos. To a certain extent, this makes micro-video recommendation [10, 19, 24, 25, 36] more challenging than traditional video recommendation task [12, 13, 34]. On conventional micro-video sharing platforms, *e.g.*, Kuaishou, TikTok, and Vine, there are no constraints on the topics of micro-videos. However, the micro-videos on e-commerce platforms are mainly about products. The users' behavioral intentions on these platforms are usually different. For example, on TikTok, the users interact with micro-videos mainly for having fun. When

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD 2021, August 14–18, 2021, Singapore

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

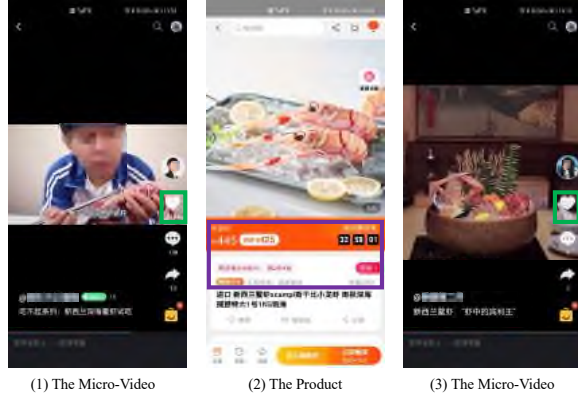


Figure 1: Screenshots of the Taobao mobile app (best viewed in color). (2) shows the product page about crawfish which consists of a series of promotion information (purple box), such as price and coupons. (1) and (3) are two micro-videos about crawfish, where (1) is more popular than (2) according to the number of “likes” (green box).

interacting with micro-videos on Taobao, besides entertainment, the users are more prepared to be “inspired” or purchase a specific kind of products. Therefore, existing video and micro-video recommendation methods usually achieve unsatisfied performances for micro-video recommendation on e-commerce platforms without considering the users’ purchasing intentions.

On e-commerce platforms, there is a large volume of user behavior data on products, which can help identify users’ preferences and purchasing intentions. In this work, we propose to leverage the product-related user behavior data for micro-video recommendation. It is a very challenging task, because there exists a large gap between users’ behaviors on products and micro-videos. Although micro-videos are about products, their forms are quite different, which is the crucial reason influencing users’ behavior patterns in these two domains [35]. Products are well structured with clearly defined attributes, e.g., price and keyword descriptions. However, micro-videos are the multimedia content created based on product information. Besides the introduced products, whether a user enjoys a micro-video is more dependent on her tastes on the multi-modal information of the micro-video. For better understanding, we illuminate an example in Figure 1, where all three contents describe the New Zealand crawfish. As shown in Figure 1 (2), the product page is structured with promotion information, clean cover pictures, and keyword-like titles, making users more inclined to purchase the product directly and quickly. On the contrary, micro-videos are more diverse with multimedia content. The video in Figure 1 (1) tends to attract users through a humorous tasting topic, while the video in Figure 1 (3) shows more ingredients in high-end restaurants. In addition, different multi-modal information display forms of micro-videos may also cause different popularity.

This study focuses on developing domain transfer and adaptation techniques to effectively exploit users’ sequential behaviors in product-domain for sequential micro-video recommendation. We propose a novel cross-domain recommendation method, named **SEquential Multi-modal Information transfer network (SEMI)** for

our task. Specifically, SEMI learns users’ cross-domain interests to overcome the differences of content structures and users’ sequential behavior patterns in different domains. Hierarchical transformer structure [33] is employed to analyze the intra-relationship between behaviors in each single domain and the inter-relationship between cross-domain behaviors. To better bridge the behavior pattern gap between micro-video domain and product domain, we further propose a **Cross-domain Contrastive Learning (CCL)** algorithm to pre-train encoders to describe sequential user behaviors. CCL imposes dual constraints on the cross-domain sequential behaviors by minimizing the gap between micro-videos and products’ representations in the same user behavior sessions and maximizing the distances between randomly sampled pairs.

The main contributions made in this work are as follows:

- To the best of our knowledge, this is the first study investigating e-commerce micro-video recommendation, whose importances and business values have been discussed before.
- We propose to use product-domain user behaviors to enhance the micro-video recommendation performances. To that end, we propose the **SEquential Multi-modal Information transfer network (SEMI)**, which effectively selects relevant items in both domains to characterize user preferences.
- To bridge the gap between micro-video domain and product domain, we propose a **Cross-domain Contrastive Learning (CCL)** algorithm to pre-train encoders for modeling users’ sequential behaviors from the multimodality view. CCL-based pre-training strategy significantly improves the performances of SEMI.
- We conduct extensive experiments on a large-scale real-world dataset collected from Taobao. The experimental results verify that our proposed method outperforms state-of-the-art recommendation methods. We have also deployed the proposed method on Taobao, and online A/B testing results demonstrate its practical values.

2 RELATED WORK

2.1 Video Recommendation

Personalized video recommendation has been well studied in the last decade. Previous video recommendation methods can be divided into three main categories, *i.e.*, collaborative filtering based methods [4, 17, 37], content-based methods [8, 12, 34], and hybrid approaches [5, 6]. Most of these approaches deal with traditional online videos instead of micro-videos. With the growing dominant trend of micro-videos in the mobile internet era, micro-video recommendation has attracted increasing research attentions [9, 10, 19, 22, 24, 25, 28, 36]. Compared with traditional video recommendation task, micro-video recommendation needs more targeted research to address its unique challenges, *i.e.*, diverse and dynamic user interests. Existing micro-video recommendation methods mainly focus on exploiting the multi-modal features of micro-videos and the users’ sequential behaviors. For example, [10] proposes the temporal hierarchical attention at category- and item-level network for the micro-video recommendation. [25] designs a user-video co-attention network to learn multi-modal information from both users and micro-videos for recommendation. [19] proposes an end-to-end multi-scale time-aware user interest modeling network to explore multi-scale time effects on the user interests.

Although these studies usually achieve promising performance improvements, they cannot be directly applied to the micro-video recommendation on e-commerce platforms. Because they only focus on the user-video interactions without considering the users' purchasing intentions.

2.2 E-Commerce Recommendation

Due to its great commercial value, personalized recommendation system has been widely deployed on e-commerce platforms [7, 15, 16, 23, 32, 38, 39]. For example, [39] proposes a deep attention model named Deep Interest Network (DIN), which learns the user representations from historical behaviors adaptively. [32, 38] further investigate the users' dynamic and long-term interests and propose the Deep Interest Evolution Network (DIEN) and Multi-channel user Interest Memory Network (MIMN), respectively. [23] studies the multi-task problem in e-commerce platforms and proposes a Pareto-efficient algorithm. Previous studies about e-commerce recommendation tasks mainly focus on recommending products and investigating users' sequential behaviors on products. To the best of our knowledge, none of them explores the micro-video recommendation on e-commerce platforms. As most micro-videos on e-commerce platforms are about products, and the e-commerce environment determines users' overall shopping mentality, it is natural to exploit users' behaviors on products to enhance the micro-video recommendation. Nevertheless, micro-videos have unique characteristics different from products, *e.g.*, diverse display forms and different users' behavior patterns. Thus, the micro-video recommendation on e-commerce platforms is still very challenging.

2.3 Cross-Domain Recommendation

The objective of cross-domain recommendation is to improve the target domain's recommendation performances by transferring knowledge from relevant source domains. It has been demonstrated to effectively alleviate the cold start problem and data sparsity issues [1, 21]. Cross-domain recommendation approaches can be broadly categorized into two groups, *i.e.*, traditional methods and deep learning-based methods. Traditional methods transfer knowledge from source domain using shallow models [18, 26]. For example, [26] utilizes several matrix factorization models separately on different domains to model user preferences. [18] proposes to use tensor-based factorization to share latent features between different domains. Compared with traditional methods, deep learning-based methods employ neural networks to learn high-level and complex information within multiple domains [2, 3, 14, 29–31]. For example, [2] introduces a collaborative cross-network to enable dual knowledge transfer across different domains. [14] proposes a multi-view deep learning framework to recommend items by combining rich user features from different domains. [31] transfers users' interests from the news domain to assist advertisement recommendation by the mixed interest network. Moreover, [29] investigates how to share the information from different domains of the same user account, and proposes a parallel information-sharing sequential recommendation model. In summary, none of these cross-domain recommendation methods are designed for e-commerce micro-video recommendation, and they lack the exploration of how to maximize

mutual information of sequential behaviors with the multi-modal view.

3 PROPOSED RECOMMENDATION MODEL

Figure 2 shows the proposed cross-domain e-commerce micro-video recommendation framework. In this section, we first introduce the problem formulation. Then, we introduce some basic components of the proposed framework. Next, we introduce the proposed Cross-domain Contrastive Learning (CCL) algorithm. Finally, we describe the details of the SEquential Multi-modal Information transfer network (SEMI).

3.1 Problem Formulation

The objective of personalized e-commerce micro-video recommendation is to build a prediction model to estimate the probability that a user would like to watch a specific micro-video. Particularly, in this work, the “watch” behavior stands for watching the micro-video completely. As the user's interests are dynamic, we use her recent behaviors to help predict. Here, a user's recent behaviors include her recent clicking behaviors on products and recent watching behaviors on micro-videos. For simplicity, we omit other personal data of users, *e.g.*, demographics.

We denote the target micro-video by v_t , the user's product clicking sequence and micro-video watching sequence by $P = \{p_1, p_2, \dots, p_l\}$ and $V = \{v_1, v_2, \dots, v_n\}$, respectively, where l and n denote the length of each sequence. The prediction problem can be formalized as follows,

$$Prob(v_t|P, V) \sim f(P, V, v_t) \quad (1)$$

where the user is represented by her recently interacted micro-videos V and products P , and $Prob(v_t|P, V)$ is the probability that the user would like to watch the target micro-video v_t . Moreover, $f(P, V, v_t)$ is the model used to estimate $Prob(v_t|P, V)$.

3.2 Basic Components

The proposed recommendation model includes the following three basic components, *i.e.*, feature extractor, multi-head attention block, and multi-modal sequence encoder.

3.2.1 Feature Extractor. In this work, we extract multi-modal features for micro-videos and products, which include micro-video titles, micro-video tags, micro-video frames, product titles, product covers, and product categories. Given the multi-modal information of micro-videos and products, the next step is to obtain their semantic representations. Inspired by the powerful information fusion ability of large-scale multi-modal pre-training models [11, 27], we adopt UniVL [27] to fuse the multi-modal information of micro-videos. Specifically, we utilize micro-video titles and frames as inputs and pre-train the UniVL model using about 100 million e-commerce micro-videos. The pre-training tasks of UniVL include masked language modeling, masked frame modeling, video-text alignment, and tag classifications. Finally, for each micro-video v , we have its multi-modal representation $v^m \in \mathcal{R}^{128}$, which describes the semantic information of micro-video frames, titles, and tags. Similarly, we have the semantic representation $p^m \in \mathcal{R}^{128}$ for each product p , which is extracted by the pre-trained UNITER model [11]. It describes the semantic information of product cover

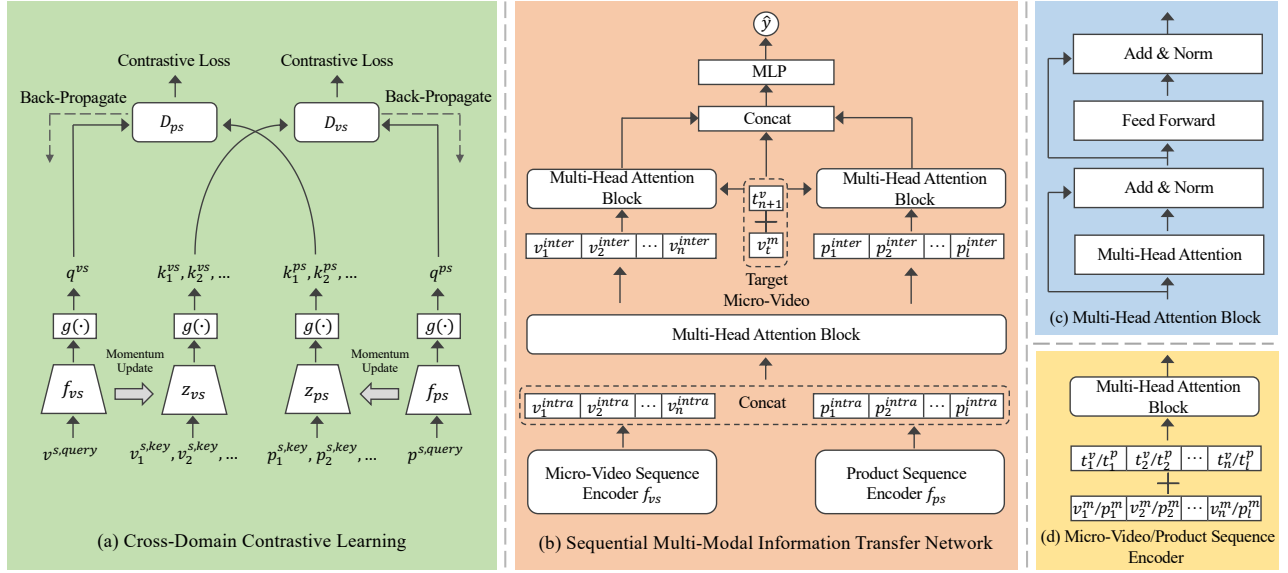


Figure 2: The framework of our proposed method (best viewed in color). Parts (c) and (d) are two basic structures used in our method (c.f. Section 3.2). The part (a) illustrates our proposed CCL algorithm to pre-train sequence encoders for different domains (c.f. Section 3.3). The part (b) describes the proposed SEMI transfer network for e-commerce micro-video recommendations (c.f. Section 3.4), where the sequence encoders of two domains are initialized by parameters pre-trained by CCL.

pictures, titles, and categories. In this work, the UNITER model is pre-trained by about 100 million products. More details about our feature extractors can be found in Appendix A.5.

3.2.2 Multi-Head Attention Block. Motivated by the success of multi-head attention mechanism [33] in sequential recommendation [7], we apply it to capture users' sequential behavior patterns on micro-videos and products. The structure of the attention mechanism module is illustrated in Figure 2 (c). Following [33], we linearly project an input vector n_h times (n_h -heads) to latent vectors with d_h dimensions, by different linear projections. Next, we perform scaled dot-product attention for each head, and concatenate all scaled dot-product attention output vectors along with each other as follows,

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) \quad (2)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

where $Q \in \mathcal{R}^d$, $K \in \mathcal{R}^d$, and $V \in \mathcal{R}^d$ represent Query, Key, and Value, respectively. $W_i^Q \in \mathcal{R}^{d \times d_h}$, $W_i^K \in \mathcal{R}^{d \times d_h}$, and $W_i^V \in \mathcal{R}^{d \times d_h}$ are weight matrices. The scaled dot-product attention function is defined as follows,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (3)$$

where d_k is the dimension of K and equals d_h in our work. Then, the multi-head attention network employs a feed-forward network for further strengthening the model performances. The feed-forward network includes two full-connection layers, and it is applied to each input vector separately and identically.

3.2.3 Multi-Modal Sequence Encoder. Figure 2 (d) shows the structure of the micro-video and product sequence encoders. As the multi-head attention mechanism is not aware of the order information of the sequence, we use timestamp embedding to preserve the sequential order information for capturing dynamic user preferences over time. Firstly, we split the user's historical behaviors into multiple time blocks. The length of each time block is empirically set to one hour. Then, we learn a timestamp embedding for each time block, which is denoted as $t_i^v \in \mathcal{R}^{128}$ and $t_i^p \in \mathcal{R}^{128}$ for micro-videos and products, respectively. Note that, t_i^v and t_i^p share the embedding for the same time block. Consequently, the input feature of each micro-video $v_i^{in} \in \mathcal{R}^{128}$ ($i \in [1, n]$) is derived by summing the pre-extracted multi-modal representation $v_i^m \in \mathcal{R}^{128}$ and the timestamp embedding $t_i^v \in \mathcal{R}^{128}$. Similarly, we have the input $p_j^{in} \in \mathcal{R}^{128}$ ($j \in [1, l]$) for each product. Taking the micro-video sequence as an example, following the inputs of micro-videos, there is a multi-head attention block, where the Query, Key, and Value are the same. In particular, they are $V^{in} = \{v_1^{in}, \dots, v_n^{in} | v_i^{in} \in \mathcal{R}^{128}\}$. Finally, we have the representation $V^{intra} \in \mathcal{R}^{n \times 128}$ of a micro-video sequence, which captures the user's sequential behaviors on micro-videos. Similarly, we have $P^{intra} \in \mathcal{R}^{l \times 128}$ as the representation of a product sequence.

3.3 Cross-Domain Contrastive Learning

On e-commerce platforms, there are a large number of users with very sparse micro-video behaviors (e.g., less than 3 watched micro-videos in the last 7 days). If we train the cross-domain recommendation model from scratch, these users' behavior data will affect the model's cross-domain knowledge transferability (c.f. the in-depth

Algorithm 1: Cross-Domain Contrastive Learning

Input: Collections of micro-video and product sequential behaviors V^s and P^s ; dictionary D_{vs} and D_{ps} ; encoders $f_{vs}, f_{ps}, z_{vs}, z_{ps}$; dictionary size K ; learning rate γ ; mini-batch size M ; momentum m .

Output: Optimal solution $\theta_q^{vs}, \theta_q^{ps}, \theta_k^{vs}, \theta_k^{ps}$

```

1 Initialize parameters,  $\theta_q^{vs}, \theta_q^{ps}, \theta_k^{vs}, \theta_k^{ps} \sim \text{Uniform}(0, 1)$ 
2 Load a dictionary at random for  $D_{vs}$ ,
    $D_{vs} \leftarrow \{v_1^s, \dots, v_K^s\} \sim \text{Random}(V^s)$ 
3 Load a dictionary at random for  $D_{ps}$ ,
    $D_{ps} \leftarrow \{p_1^s, \dots, p_K^s\} \sim \text{Random}(P^s)$ 
4 Encode dictionary samples,  $k_i^{vs} \leftarrow g(z_{vs}(v_i^s)), \forall v_i^s \in D_{vs}$ ,
    $D_{vs}, k_i^{ps} \leftarrow g(z_{ps}(p_i^s)), \forall p_i^s \in D_{ps}$ 
5 for each epoch do
6   for each mini-batch do
7     Load a mini-batch of micro-video sequence,
        $B_{vs} \leftarrow \{v_1^s, \dots, v_M^s\} \sim V^s$ 
8     Load a mini-batch of product sequence,
        $B_{ps} \leftarrow \{p_1^s, \dots, p_M^s\} \sim P^s$ 
       // Update dictionaries
9     Encode mini-batch samples,
        $k_i^{vs} \leftarrow g(z_{vs}(v_i^s)), \forall v_i^s \in B_{vs}$ 
10    Encode mini-batch samples,
        $k_i^{ps} \leftarrow g(z_{ps}(p_i^s)), \forall p_i^s \in B_{ps}$ 
11    Update  $D_{vs} \leftarrow \text{ENQUEUE}(\text{DEQUEUE}(D_{vs}, B_{vs}))$ 
12    Update  $D_{ps} \leftarrow \text{ENQUEUE}(\text{DEQUEUE}(D_{ps}, B_{ps}))$ 
       // Cross-domain contrastive coding
13    Encode mini-batch samples,
        $q_i^{vs} \leftarrow g(f_{vs}(v_i^s)), \forall v_i^s \in B_{vs}$ 
14    Encode mini-batch samples,
        $q_i^{ps} \leftarrow g(f_{ps}(p_i^s)), \forall p_i^s \in B_{ps}$ 
15    Compute the posterior,
        $\text{prob}(y_i^{vs} | v_i^s, p_i^s, D_{vs}) \leftarrow \text{Eq. 5}$ 
16    Compute the posterior,
        $\text{prob}(y_i^{ps} | p_i^s, v_i^s, D_{ps}) \leftarrow \text{Eq. 6}$ 
       // Update model parameters
17    Update  $\theta_q^{vs} \leftarrow \text{Eq. 7}$ ,  $\theta_q^{ps} \leftarrow \text{Eq. 8}$ 
18    Momentum update  $\theta_k^{vs}, \theta_k^{ps} \leftarrow \text{Eq. 9}$ 

```

analysis in Section 4.5). Therefore, we propose a Cross-domain Contrastive Learning (CCL) algorithm to pre-train the micro-video and product multi-modal sequence encoders. The motivation is that an active user's actions on the e-commerce platform within a session have a certain consistency of intention [15]. Hence, the objective of CCL is to encourage the representations of a micro-video sequence and a product sequence to be similar, if they come from the same session of an active user. Inspired by the success of MoCo [20], we design a momentum contrastive learning algorithm for the cross-domain scenario. As illustrated in Figure 2 (a), we build two queue-based dictionaries for micro-video and product sequence per mini-batch, where "keys" are randomly sampled from the data. The sequence encoders are trained to perform dictionary

look-up: an encoded "query" should be similar to the value of its matching key and dissimilar to others. This training objective maximizes a lower bound of the mutual information between different domains.

Let $V^s = \{v_1^s, \dots, v_L^s\}$ and $P^s = \{p_1^s, \dots, p_L^s\}$ denote the collections of users' behavior sequences, where v_i^s and p_i^s are a user's sequential actions on micro-videos and products, respectively. Note that each pair (v_i^s, p_i^s) is from the same session of a user. We define query encoders f_{vs} and f_{ps} and key encoders z_{vs} and z_{ps} for micro-video and product sequence behaviors, respectively. The learnable parameters of query encoders are $\{\theta_q^{vs}, \theta_q^{ps}\}$, and the parameters of key encoders are $\{\theta_k^{vs}, \theta_k^{ps}\}$. The structure of these sequence encoders are described in the Section 3.2.3. These encoders compute representations of micro-video and product sequences as queries and keys,

$$\begin{aligned} q^{vs} &= g(f_{vs}(v^{s, \text{query}})), & k^{vs} &= g(z_{vs}(v^{s, \text{key}})) \\ q^{ps} &= g(f_{ps}(p^{s, \text{query}})), & k^{ps} &= g(z_{ps}(p^{s, \text{key}})) \end{aligned} \quad (4)$$

where $g(\cdot)$ is the max pooling function. Finally, we have $\{q^{vs}, k^{vs}, q^{ps}, k^{ps}\} \in \mathcal{R}^{128}$.

Algorithm 1 describes the details of the proposed CCL algorithm. The dictionaries D_{vs} and D_{ps} are initialized with K randomly drawn samples from V^s and P^s , respectively (lines 2–4). For each iteration within an epoch, we load new mini-batch samples and enqueue them into dictionaries (lines 7–12). Given the updated D_{vs} and D_{ps} , we perform cross-domain contrastive coding. For micro-video sequence representations to product sequence representations, we compute the posteriors of all micro-video sequence samples $v_i^s \in B_{vs}$ with respect to the negative samples in the product sequence dictionary D_{ps} as follows,

$$\text{prob}(y_i^{vs} | v_i^s, p_i^s, D_{ps}) = \frac{\exp(q_i^{vs} \cdot k_i^{ps} / \tau)}{\sum_{j=1}^K \exp(q_i^{vs} \cdot k_j^{ps} / \tau)}, \quad \forall i \in [1, M] \quad (5)$$

where M is mini-batch size, and τ is a temperature term. The posterior is defined over a cross-domain space with one positive and K negative pairs. Similarly, we have posteriors for product sequence representations to micro-video sequence representations (lines 15–16),

$$\text{prob}(y_i^{ps} | p_i^s, v_i^s, D_{vs}) = \frac{\exp(q_i^{ps} \cdot k_i^{vs} / \tau)}{\sum_{j=1}^K \exp(q_i^{ps} \cdot k_j^{vs} / \tau)}, \quad \forall i \in [1, M] \quad (6)$$

Next, we back-propagate gradients only to the query encoders f_{vs} and f_{ps} according to the cross-entropy loss (line 17),

$$\theta_q^{vs} \leftarrow \theta_q^{vs} - \gamma \nabla_{\theta} \mathcal{L}_{CE}(-\log \text{prob}(y^{vs} | \cdot, y_{gt}^{vs}))|_{\theta=\theta_q^{vs}} \quad (7)$$

$$\theta_q^{ps} \leftarrow \theta_q^{ps} - \gamma \nabla_{\theta} \mathcal{L}_{CE}(-\log \text{prob}(y^{ps} | \cdot, y_{gt}^{ps}))|_{\theta=\theta_q^{ps}} \quad (8)$$

where γ is the learning rate, and y_{gt}^{vs} is the ground truth label indicating whether (v_i^s, p_j^s) is a pair. After that, we apply momentum update [20] to the parameters of key encoders z_{vs} and z_{ps} (line 18),

$$\theta_k^{vs} \leftarrow m\theta_k^{vs} + (1-m)\theta_q^{vs}, \quad \theta_k^{ps} \leftarrow m\theta_k^{ps} + (1-m)\theta_q^{ps} \quad (9)$$

where m is the momentum value. The momentum update allows the dictionaries to change their states slowly, thus making them consistent across iterations. In summary, we perform bi-directional contrastive coding and train the whole model end-to-end. Finally,

CCL generates pre-trained sequence encoders, which can learn discriminative micro-video and product sequence representations.

3.4 Sequential Multi-Modal Information Transfer Network

The pre-trained sequence encoders bridge the domain gap at the sequence representation level by maximizing the mutual information between different domains. In our problem setting, we need to predict whether the user likes a specific micro-video (e.g., target micro-video). Therefore, it requires us to further analyze the relationship between the target micro-video and the user’s cross-domain behaviors. This is challenging because users’ behaviors in micro-video domain are more sparse than their behaviors in the product domain. Moreover, users’ behavior patterns are also largely different in these two domains. To solve these problems, we propose a Sequential Multi-Modal Information Transfer Network (SEMI). Specifically, SEMI consists of a hierarchical attention mechanism that can fully integrate and enhance users’ interests in the two domains and automatically select relevant items in these domains to capture users’ preferences dynamically.

As shown in Figure 2 (b), we first initialize the parameters of two sequence encoders pre-trained by the CCL algorithm, and generate intra-domain representations $V^{intra} \in \mathcal{R}^{n \times 128}$ and $p^{intra} \in \mathcal{R}^{l \times 128}$ for micro-video and product sequential behaviors, respectively. Then, we feed $[V^{intra}, p^{intra}] \in \mathcal{R}^{(n+l) \times 128}$ into an additional multi-head attention block to capture the inter-domain dependencies, where $[\cdot]$ denotes concatenation. The Query, Key, and Value of this inter-domain multi-head attention block are the same, that is, $[V^{intra}, p^{intra}]$. We have the user cross-domain sequential representations $\{V^{inter} \in \mathcal{R}^{n \times 128}, p^{inter} \in \mathcal{R}^{l \times 128}\}$. Then, we need to evaluate the similarity between the target micro-video and the sequences in the two domains respectively, and generate the domain-specific user representation in each domain. Thus, we utilize another multi-head attention blocks over the target micro-video and $\{V^{inter}, p^{inter}\}$, respectively. The representation of target micro-video v_t^{in} is the sum of timestamp embedding and pre-extracted multi-modal features. In these two multi-head attention blocks, Query is the target micro-video, and Key and Value are the same, that is, V^{inter}/p^{inter} . Finally, we can obtain the user representations u_v and u_p in the micro-video domain and product domain, respectively.

In the experiments, we find that concatenation is the most efficient and powerful way to combine multiple representation vectors. Therefore, the concatenation of u_v , u_p , and v_t^{in} are fed into a multi-layer perceptron to predict the probability $\hat{y}$ that the user would like the target micro-video. The following cross-entropy loss is used to learn the model parameters,

$$\mathcal{L}(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log (1 - \hat{y}) \quad (10)$$

where $y \in \{0, 1\}$ is the ground-truth label indicating whether the user watches the micro-video completely or not.

4 EXPERIMENTS

To demonstrate the effectiveness of the proposed method, we perform extensive offline experiments on a large-scale dataset collected from Taobao, which is a world’s largest e-commerce platform. We

also deploy the proposed method on Taobao platform and perform online A/B testing to demonstrate its effectiveness.

4.1 Datasets

A large-scale real-world dataset is collected from one of the micro-video consumption scenarios, which serves about 10 million users every day, on Taobao. This dataset covers 8 consecutive days of users’ interacting records on micro-videos from the Taobao mobile App’s log during December 2020. Each record consists of a record ID, a user ID, a micro-video ID, a timestamp, an interaction flag (watched the micro-video completely or not), and two sequences of the user’s completely watched micro-videos and clicked products within recent 7 days, respectively. Moreover, each sequence consists of multiple tuples of (pre-extracted multi-modal features for micro-video/product, timestamp). Here, we only reserve the records with at least one watched micro-video or five clicked products. In addition, we truncate the sequence lengths of the micro-video domain and product domain into 50, respectively, if the sequence length is larger than 50. We keep all positive records (i.e., the user watched the micro-video completely) and randomly sample a subset of negative records (i.e., the user watched the micro-video but not completely) to keep the ratio of positive and negative records 1 : 1 nearly. We hold the first seven days as the training set and leave the eighth day for testing. Finally, we have about 10.39×10^8 and 2.49×10^8 samples for training and test datasets, respectively. Furthermore, we select samples in training data for CCL pre-training. Specifically, we keep records in the training set where the micro-video sequence length is larger than 10. Then, we do deduplication for these records according to the user ID dimension and keep one record with the longest average length of two domains’ sequences for each user. In this way, we have about 2.83×10^7 cross-domain sequence pairs with similar potential interests for CCL pre-training. More statistics of the training set, test set, and CCL pre-training set are listed in Appendix A.3.

4.2 Experimental Settings

4.2.1 Baseline Methods. We compare the proposed SEMI transfer network with CCL pre-training strategy (denoted as SEMI w. CCL) with the following baseline methods:

- **DIN** [39]: This method uses the embedding product attention mechanism to learn adaptive representations of users’ behaviors.
- **DIEN** [38]: This method introduces an interest-evolving layer on the top of DIN to capture users’ dynamic interests over time.
- **BST** [7]: This is a state-of-the-art method on modeling users’ sequential behaviors. It leverages the Transformer [33] structure with time information for e-commerce product recommendation.
- **YoutubeNet** [12]: This method is proposed to recommend videos in YouTube. It obtains user representations by simply averaging the item embeddings in the user behavior sequence.
- **BERT4SessRec** [8]: It is a video recommendation model with BERT-like structure and employs a pre-training strategy to better model users’ sequential behaviors.
- **THACIL** [10]: It is a state-of-the-art micro-video recommendation method modeling users’ sequential behaviors. It leverages item- and category-attention mechanisms to capture the diverse and fine-grained interests, respectively.

Table 1: Performances of different methods on e-commerce micro-video recommendations. * indicates the improvement over the second-best baseline is statistically significant with $p < 0.01$ using t-test.

Group	Method	Metrics	
		AUC	HR@3
Single-Domain	DIN	0.7027	0.4524
	DIEN	0.7067	0.4577
	BST	0.7128	0.4651
	YouTubeNet	0.6833	0.4265
	BERT4SessRec	0.7182	0.4700
	THACIL	0.7019	0.4519
Cross-Domain	CoNet	0.6749	0.4220
	π -Net	0.7241	0.4784
	MiNet	<u>0.7244</u>	<u>0.4785</u>
	Cross-BST	0.7202	0.4752
	SEMI w. CCL	0.7396*	0.4997*

- **CoNet** [2]: This method adds cross-connection units on two MLPs with shared user embeddings across domains to enable dual knowledge transfer.
- **π -Net** [29]: It uses RNNs to encode sequences and a gating mechanism to transfer information between sequences in different domains.
- **MiNet** [31]: This cross-domain recommendation method jointly models three types of user interests from different domains and utilizes two level of attentions to fuse these interests.
- **Cross-BST**: This method is extended from BST [7]. It is the baseline previously used in online services of Taobao. In this method, we directly fuse micro-videos and products as one behavior sequence, which are ordered by timestamp.

We run each method three times and compute the mean to eradicate any discrepancies. Hyper-parameter settings and implementation details of our method and baseline methods are introduced in Appendix A.1 and A.2, respectively.

4.2.2 Evaluation Metrics. We adopt two commonly used performance metrics for evaluation, *i.e.*, Area Under the receiver operating characteristic Curve (AUC) and Hit Ratio at rank K (HR@ K). Specifically, we calculate HR@ K according to the target micro-video and 300 randomly selected micro-videos. As the online system of Taobao requests three recommended results at a time, we set K to 3. The larger value of AUC and HR@ K indicates better performances.

4.3 Overall Performance Comparison

Table 1 summarizes the e-commerce micro-video recommendation performances achieved by different methods. From Table 1, we can make the following observations.

The single-domain methods can be divided into two sub-groups: classical e-commerce recommendation methods (DIN, DIEN, BST) and video/micro-video recommendation methods (YouTubeNet, BERT4SessRec, THACIL). Comparing these two sub-groups, we can note that the classical e-commerce recommendation methods are

Table 2: Results of ablation studies. SEMI-Product denotes only using product domain behaviors as inputs. SEMI-Micro-Video denotes only using micro-video domain behaviors as inputs. “w. CCL” indicates training the model with CCL pre-training. “w/o CCL” denotes training the model from scratch using without CCL pre-training.

Group	Method	Metrics	
		AUC	HR@3
Best Baselines	BERT4SessRec	0.7182	0.4700
	MiNet	0.7244	0.4785
w/o CCL	SEMI-Product w/o CCL	0.6797	0.4250
	SEMI-Micro-Video w/o CCL	0.7168	0.4694
	SEMI w/o CCL	0.7238	0.4790
w. CCL	SEMI-Product w. CCL	0.6990	0.4442
	SEMI-Micro-Video w. CCL	0.7161	0.4689
	SEMI w. CCL	0.7396	0.4997

comparable with the video/micro-video recommendation methods. This indicates that one crucial point for the micro-video recommendation task is to depict users’ sequential behaviors. In particular, BST and BERT4SessRec achieve better performance. The potential reason is that the transformer architecture is effective in capturing sequence patterns. Thus, we also adopt transformer architectures in our method.

Secondly, the cross-domain methods usually outperform single-domain methods, except for CoNet, which is a non-sequential method. This result verifies our motivation that product domain behaviors and micro-video domain behaviors are interrelated for e-commerce scenarios. The users’ behaviors in the product domain can supplement and enhance users’ interests in the e-commerce micro-video domain. Moreover, it is essential to design and learn a cross-domain sequential model.

Thirdly, according to the results, it is obvious that our proposed SEMI transfer network with CCL pre-training strategy (SEMI w. CCL) outperforms all the baselines, in terms of all the evaluation metrics. Specifically, compared to the second-best baseline (*i.e.*, MiNet), SEMI w. CCL achieves 2.10% and 4.43% better performances, in terms of AUC and HR@3, respectively. These experimental results demonstrate the effectiveness of the proposed method in the e-commerce micro-video recommendation task.

4.4 Ablation Study

Moreover, we also perform ablation study to investigate the impacts of different components of the proposed method. Table 2 summarizes the results of ablation studies.

Effect of Different Domain Behaviors. For both groups in Table 2, the cross-domain method (*i.e.*, SEMI) significantly outperforms the single-domain methods with only micro-video sequence behaviors (*i.e.*, SEMI-Micro-Video) or product sequence behaviors (*i.e.*, SEMI-Product). This further demonstrates the necessity of cross-domain interest modeling for the e-commerce micro-video recommendation task. Moreover, SEMI-Micro-Video w/o CCL gains 5.46% AUC and 10.45% HR@3 against SEMI-Product w/o CCL. It

confirms that there is a noticeable domain gap between micro-video and product domains.

Effect of SEMI. To study the performances of SEMI transfer network, we train the SEMI transfer network from scratch without CCL (denoted as SEMI w/o CCL). It can be observed that SEMI w/o CCL consistently and significantly outperforms the best single-domain baseline (BERT4SessRec). Compared with the best cross-domain baseline (MiNet), SEMI w/o CCL also achieves comparable results, especially in terms of HR@3. These results verify the effectiveness of SEMI transfer network in introducing cross-domain information.

Effect of CCL. Firstly, SEMI w. CCL achieves improvements as much as 2.18% for AUC and 4.32% for HR@3, compared to SEMI w/o CCL. This demonstrates the effectiveness of the CCL pre-training strategy for cross-domain interests transfer. Secondly, comparing SEMI-Product w. CCL to SEMI-Product w/o CCL, SEMI-Product w. CCL gains 2.84% AUC and 4.52% HR@3. This confirms that the CCL can help learn interests in the micro-video domain from the product domain behaviors. Finally, comparing SEMI-Micro-Video w. CCL to SEMI-Micro-Video w/o CCL, it does not achieve noticeable improvements and sometimes even exhibits worse. One possible reason is that CCL forces the micro-video sequence encoder to express some information in the product domain irrelevant to the micro-video recommendation task.

4.5 In-Depth Analysis for CCL

In this section, we analyze the impacts of different pre-training settings on the recommendation performances and present an in-depth analysis of the effectiveness of CCL.

Effect of Dictionary Size. Figure 3 shows how the dictionary size of CCL affects the performances of e-commerce micro-video recommendation task. Firstly, when the value is less than about 2×10^4 , the performance improves rapidly with the increase of the dictionary size. Then, CCL benefits a little from larger dictionaries up to a threshold (at about 5×10^4). Finally, CCL suffers after the dictionaries larger than about 5×10^4 . This observation is consistent with previous empirical findings of contrastive learning [20]. In particular, there are performance limits by increasing the dictionary size of the randomly-sampling strategy. A better way to build the contrastive learning dictionary is left for future study.

Effect of Pre-Training Dataset. We investigate the effects of the minimum micro-video sequence length of the pre-training dataset for CCL, which is denoted by MLEN in the following. Figure 4 shows that SEMI w. CCL achieves the best performance when MLEN equals 10. No matter whether MLEN is smaller or larger than 10, the performance would decline. One possible reason is that users' micro-video interests with very sparse micro-video behaviors may not be robust and convincing. If these data directly participate in CCL for pre-training or SEMI for training from scratch, the model's transfer learning ability for different domain behaviors may be hurt. For example, SEMI w. CCL even performs slightly worse than SEMI w/o CCL, when MLEN equals 1. On the contrary, for users with very dense micro-video behaviors (e.g., MLEN equals 20), their micro-video domain interests are relatively dynamic, and the related samples are relatively small. The sequence encoders pre-trained by CCL will be biased if only utilizing these active users in the micro-video domain.

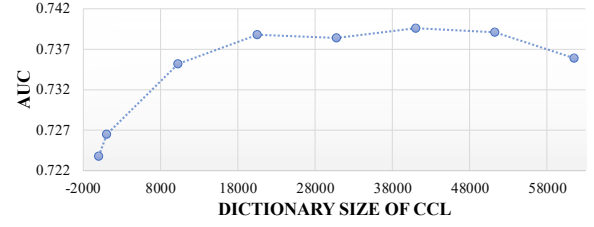


Figure 3: Recommendation performances w.r.t. different dictionary size of CCL. Particularly, when the size equals 0 (the leftmost node of the curve), it means SEMI w/o CCL.

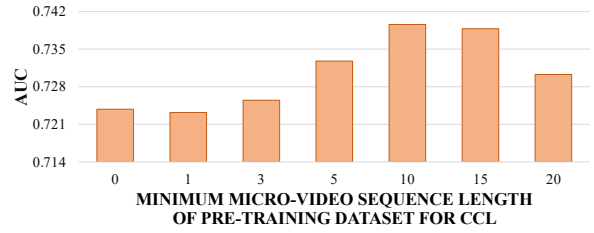


Figure 4: Recommendation performances w.r.t. minimum micro-video sequence length of pre-training dataset for CCL. Particularly, when the minimum length equals 0, it means SEMI w/o CCL.

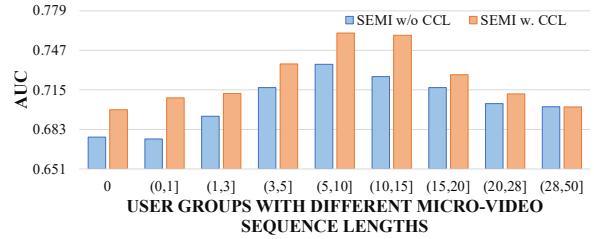


Figure 5: Recommendation performances w.r.t. user groups with different micro-video sequence lengths.

Improvements for Different User Groups. Figure 5 shows the performances of SEMI w. CCL and SEMI w/o CCL for users with different activeness in the micro-video domain. On the whole, the CCL pre-training strategy brings gains to all user groups. Especially for the inactive users in the micro-video domain, SEMI w. CCL achieves the highest gains. Take users with one or less micro-video action as an example, SEMI w. CCL achieves an average gain of 4.12%. For micro-video domain active users, the gains brought by CCL are relatively small. It is not surprising since the micro-video domain's rich behaviors for these active users play a leading role in the recommendation performances.

4.6 Online Experiments

We perform a careful online A/B testing on Taobao from 2020-11 to 2020-12, which is under the bucket tests. One bucket is selected for baseline and another bucket for our model. Each bucket serves about 0.5 million users per day. For the whole bucket, the goal is to increase the user stickiness and activity. During nearly two months

of A/B testing, our approach contributes up to 9.32% number of browsing videos per user, 10.45% dwell time per user, and 12.10% complete-watch ratio per video, compared with the online baseline (Cross-BST). These online benefits from our method are crucial for the micro-video commercial layout of Taobao. For example, advertising videos will have more broadcast opportunities with the increased number of users watching micro-videos. The proposed method has already been fully deployed online in December 2020 and serves over 30 million users on mobile Taobao every day. The details of the online deployment are introduced in Appendix A.4.

5 CONCLUSIONS

This paper studies the e-commerce micro-video recommendation task, which is a crucial application of e-commerce platforms. Considering the characteristics of micro-videos on the e-commerce platform, such as most e-commerce micro-videos are about products, we propose to utilize user behaviors in the product domain to assist the task. Specifically, we propose a sequential multi-modal information transfer network to identify and enhance user preferences in the micro-video domain. To better bridge the gap between the two domains, we propose a cross-domain contrastive learning pre-training strategy to learn the sequence behavior encoders. Experiments on large-scale industrial dataset and long-term online A/B testing results demonstrate the effectiveness of the proposed method, compared with state-of-the-art methods. Furthermore, the proposed method has been fully deployed online on Taobao and serves over 30 millions of users every day.

REFERENCES

- [1] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. 2013. Cross-System User Modeling and Personalization on the Social Web. In *User Modeling and User-Adapted Interaction*. Springer, 169–209.
- [2] Guangneng Hu and Yu Zhng and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *CIKM*. ACM, 667–676.
- [3] Guangneng Hu and Yu Zhng and Qiang Yang. 2020. CATN: Cross-Domain Recommendation for Cold-Start Users via Aspect Transfer Network. In *SIGIR*. ACM, 229–238.
- [4] Shumeet Baluja, Rohan Seth, Dharshi Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. 2008. Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph. In *WWW*. ACM, 895–904.
- [5] Bisheng Chen, Jingdong Wang, Qinghua Huang, and Tao Mei. 2012. Personalized Video Recommendation through Tripartite Graph Propagation. In *MM*. ACM, 1133–1136.
- [6] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. ACM, 335–344.
- [7] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-Commerce Recommendation in Alibaba. In *DLP-KDD*. ACM, 1–4.
- [8] Xusong Chen, Dong Liu, Chenyi Lei, Rui Li, Zheng-Jun Zha, and Zhiwei Xiong. 2019. BERT4SessRec: Content-Based Video Relevance Prediction with Bidirectional Encoder Representations from Transformer. In *MM*. ACM, 2597–2601.
- [9] Xusong Chen, Dong Liu, Zhiwei Xiong, and Zheng jun Zha. 2020. Learning and Fusing Multiple User Interest Representations for Micro-Video and Movie Recommendations. In *Transactions on Multimedia*, Vol. 23. IEEE, 484–496.
- [10] Xusong Chen, Dong Liu, Zheng-Jun Zha, Wengang Zhou and Zhiwei Xiong, and Yan Li. 2018. Temporal Hierarchical Attention at Category-and Item-level for Micro-video Click-through Prediction. In *MM*. ACM, 1146–1153.
- [11] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TEXT Representation Learning. arXiv:1909.11740
- [12] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for Youtube Recommendations. In *RecSys*. ACM, 191–198.
- [13] Peng Cui, Zhiyu Wang, and Zhou Su. 2014. What Videos are Similar with You?: Learning a Common Attributed Representation for Video Recommendations. In *MM*. ACM, 597–606.
- [14] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *WWW*. ACM, 278–288.
- [15] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-through Rate Prediction. In *IJCAI*. 2301–2307.
- [16] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical User Profiling for E-commerce Recommender Systems. In *WSDM*. ACM, 223–231.
- [17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.
- [18] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized Recommendation via Cross-domain Triadic Factorization. In *WWW*. ACM, 596–606.
- [19] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. 2020. What Aspect Do You Like: Multi-scale Time-aware User Interest Modeling for Micro-video Recommendation. In *MM*. ACM, 3487–3495.
- [20] Yuxin Wu Saining Xie Ross Girshick Kaiming He, Haoqi Fan. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*. IEEE.
- [21] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate?: Cross-Domain Collaborative Filtering for Sparsity Reduction. In *IJCAI*. 2052–2057.
- [22] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing Micro-videos via A Temporal Graph-guided Recommendation System. In *MM*. ACM, 384–400.
- [23] Xiao Lin, Hongjie Chen, Changhua Pei, Fei Sun, Xuanji Xiao, Hanxiao Sun, Yongfeng Zhang, Wenwu Ou, and Peng Jiang. 2019. A Pareto-Efficient Algorithm for Multiple Objective Optimization in E-Commerce Recommendation. In *RecSys*. ACM, 20–28.
- [24] Shang Liu and Zhenzhong Chen. 2019. Sequential Behavior Modeling for Next Micro-Video Recommendation with Collaborative Transformer. In *ICME*. IEEE, 460–465.
- [25] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. 2019. User-Video Co-Attention Network for Personalized Micro-video Recommendation. In *WWW*. ACM, 3020–3026.
- [26] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjani. 2014. Cross-Domain Collaborative Filtering with Factorization Machines. In *CER*. 456–661.
- [27] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation. arXiv:2002.06353
- [28] Jingwei Ma, Jiahui Wen, Mingyang Zhong, Weitong Chen, Xiaofang Zhou, and Jadwiga Indulska. 2019. Multi-Source Multi-Net Micro-Video Recommendation with Hidden Item Category Discovery. In *DASFAA*. Springer, 1464–1472.
- [29] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. pi-Net: A Parallel Information-sharing Network for Shared-account Cross-domain Sequential Recommendations. In *SIGIR*. ACM, 685–694.
- [30] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *IJCAI*. 2464–2470.
- [31] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction. In *CIKM*. ACM, 2669–2676.
- [32] Guorui Zhou Xiaoqiang Zhu Kun Gai Qi Pi, Weijie Bian. 2019. Practice on Long Sequential User Behavior Modeling for Click-Through Rate Prediction. In *KDD*. ACM, 2671–2679.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NIPS*. ACM, 6000–6010.
- [34] Peng Wang, Yunsheng Jiang, Chunxu Xu, and Xiaohui Xie. 2019. Overview of Content-Based Click-Through Rate Prediction Challenge for Video Recommendation. In *MM*. ACM, 2593–2596.
- [35] Shichao Wang, Danyang Song, Xi Chen, Tianqi Shi, and Haihong Tang. 2020. What Message an E-commerce Video Should and Should not Communicate? The Influence of Video Content on Its Business Performance. In *The Workshop on e-Business*.
- [36] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *MM*. ACM, 1437–1445.
- [37] Yanxiang Huang, Bin Cui, Jie Jiang, Kunqian Hong, Wenyu Zhang, and Yiran Xie. 2016. Real-Time Video Recommendation Exploration. In *SIGMOD*. ACM, 35–46.
- [38] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. In *AAAI*. 5941–5948.
- [39] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *KDD*. ACM, 1059–1068.

A APPENDIX

In the appendix, we first give the implementation notes of our proposed method and compared baselines to help in reproducibility. Then, we describe more details about our large-scale dataset collected from Taobao. After that, we introduce our online deployment system and pipeline. Finally, we provide some details of multi-modal feature extractors in this work.

A.1 Implementation Notes of Our Proposed Method

We set the number of heads and dimensions of feed-forward networks in all multi-head attention blocks as 8 and 128, respectively. The Multi-Layer Perceptron (MLP) used in SEMI is set to $128 \rightarrow 1$. For both SEMI and CCL, the batch size is set to 1024, and the optimizer is Adam. For CCL, we set initial learning rate $\gamma = 10^{-3}$, dictionary size $K = 40 \times 1024$, momentum $m = 0.999$, and temperature $\tau = 0.7$. For SEMI, we set the initial learning rate $\gamma = 10^{-3}$ and adopt a dropout with a rate 0.1 after the final MLP. Our proposed method is implemented with TensorFlow 1.4 in Python 2.7, is conducted on the Alibaba’s distributed cloud platform¹ which contains thousands of workers. Every two workers share an NVIDIA Tesla P100 GPU with 16GB of memory.

A.2 Implementation Notes of Baselines

We implement each baseline and report performance under its optimal settings.

- **DIN [39], DIEN [38]:** We implement DIN and DIEN according to original papers and public code² and code³. We use the same multi-modal embeddings with our method as inputs.
- **BST [7], Cross-BST:** We implement BST according to the original paper. We set the multi-head attention blocks in BST the same as SEMI, and set the number of blocks as 1. The difference of Cross-BST compared to BST is the cross-domain inputs.
- **YoutubeNet [12]:** We implement the model in our code based on the original paper.
- **BERT4SessRec [8]:** We implement BERT4SessRec [8] according to the original paper. We set the multi-head attention blocks in BERT4SessRec the same as SEMI, and set the number of blocks as 2. In the pre-training stage, the mask percentage is set as 0.1.
- **THACIL [10]:** We implement THACIL [10] according to original papers and public code⁴.
- **CoNet [2]:** We implement the model in our code based on the original paper and utilize the average pooling of different domains’ actions as the user embedding.
- **π -Net [29], MiNet [31]:** We implement these two baselines according to original papers and public code⁵ and code⁶. Since these two methods need the loss from the product domain, we utilize the latest clicked product in the sequence as the training target for the product domain in the training stage.

¹<https://data.aliyun.com>

²<https://github.com/zhougr1993/DeepInterestNetwork>

³<https://github.com/YafeiWu/DIEN>

⁴<https://github.com/Ocxs/THACIL>

⁵<https://github.com/mamuyang/PINet>

⁶<https://github.com/oywtece/minet>

Table 3: Statistics of the datasets for SEMI.

Index	Dataset	
	Training	Test
#Unique Instances	10.39×10^8	2.49×10^8
#Unique Users	4.42×10^7	1.08×10^7
#Unique Target Micro-Videos	5.30×10^6	2.33×10^6
#Unique Micro-Videos in Sequences	1.14×10^7	5.20×10^6
#Unique Products in Sequences	1.12×10^8	4.88×10^7

Table 4: Statistics of the micro-video and product sequences of the dataset for SEMI.

Number	Training		Test	
	Micro-Video	Product	Micro-Video	Product
Avg. Length	26.84	44.97	27.28	44.57
Avg. #Category	16.33	19.31	16.78	19.12
#Diversity	0.6084	0.4294	0.6154	0.4291
Avg. #Overlap	1.93		1.96	

Table 5: Statistics of the dataset for CCL.

Number	Min. Len. = (1, 5)		Min. Len. = (10, 5)		Min. Len. = (20, 5)	
	Micro-Video	Product	Micro-Video	Product	Micro-Video	Product
Avg. Length	23.06	44.96	31.19	46.03	35.38	46.30
Avg. #Category	13.67	18.60	19.31	19.38	22.36	19.68
#Diversity	0.5924	0.4137	0.6193	0.4211	0.6319	0.4251
Avg. #Overlap	2.02		2.41		2.47	
#Instances	4.64×10^7		2.83×10^7		1.64×10^7	

A.3 Details of the Datasets

We report detailed statistics of our large-scale dataset collected from Taobao. The details about how to collect our dataset can be found in Section 4.1. The categories of micro-videos in the following statistics are predicted by a content-based category-prediction model of Taobao. In the following tables, “Avg. Length” denotes the average sequence length. “Avg. #Category” denotes the average number of categories per sequence. “#Diversity” denotes the diversity of the sequence and equals “Avg. #Category”/“Avg. Length”. “Avg. #Overlap” denotes the average number of overlapped categories between two corresponding sequences. “#Instances” denotes the number of training samples. “Min. Len.” denotes the minimum sequence lengths of micro-videos and products (c.f. Section 4.1).

- **Statistics of the dataset for SEMI.** Table. 3 shows the statistics of the datasets for SEMI. The huge scale of the dataset also indicates the challenge of our task. Since we utilize multi-modal features in our method, our method can cover the problem that many videos in the test set have not appeared in the training set. Table. 4 further shows the statistics of the micro-video and product sequences of the dataset for SEMI. We have three observations from Table. 4. First, users on Taobao are more active in the product domain compare with the micro-video domain according to Avg. Length. Second, users’ interests are more dynamic in the micro-video domain than the product domain according to #Diversity. Third, user behaviors in different domains might reflect similar preferences according to Avg. #OverLap. However, the values of Avg. #OverLap are very small compared to the

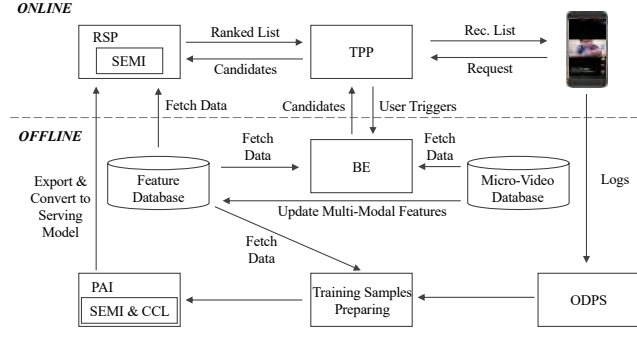


Figure 6: The deployment of SEMI w. CCL in Taobao APP.

values of Avg. #Category, which also indicates that user behavior patterns are quite different across the two domains.

- **Statistics of the dataset for CCL.** Table. 5 shows the statistics of the datasets for CCL based on different minimum micro-video sequence length (c.f. Section 4.1). Different data selection rules for CCL will impact the performance (c.f. Fig. 4).

A.4 Deployment System and Pipeline

Our proposed method has been fully deployed online in Taobao and serves over 30 million users every day. In this section, we introduce the online deployment pipeline of our proposed method. The Fig. 6 illustrates the related system and pipeline. For the online sub-system, the Taobao Personality Platform (TPP) and Ranking Service Platform (RSP) are the main components. A typical workflow is illustrated in the following:

- When a user launches Mobile Taobao App, TPP extracts the user’s latest information, including the latest clicked products, and watched micro-videos. Meanwhile, TPP utilizes the user’s information to retrieve a candidate set of micro-videos from Basic Engine (BE), which is an effective recall engine in Taobao, according to several recall approaches such as collaborative filtering and embedding similarity. These candidates are fed to RSP. RSP fetches related pre-extracted multi-modal features of candidates from Feature Database and ranks the candidate set of micro-videos with a trained SEMI transfer network and returns the ranked results to TPP. Particularly, under about 2500 service requests per second, SEMI on RSP can make inferences for about 500 candidates in batches with a delay of about 30 ms.
- During their visits to Taobao, users’ behaviors are collected and saved as log data for the offline sub-system on Open Data Processing Service (ODPS) distributed platform.

The workflow and sub-modules of the offline sub-system are described in the following:

- The log instances contain user historical sequential clicked products and watched micro-videos with timestamps and the observed micro-video currently. To date, about 2.3 billion log instances are generated by about 34 million users every day. All of these logs are saved on ODPS. Based on these logs and the ODPS platform, we prepare the training dataset for our method (c.f. Section 4.1). Besides, the related multi-modal features are also fetched from Feature Database.

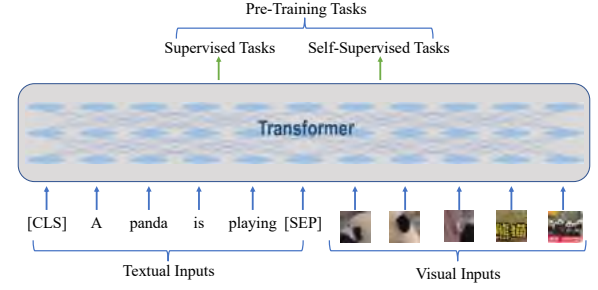


Figure 7: The general multi-modal feature extractor framework.

- To implement the proposed e-commerce micro-video recommendation algorithms, we train our models on Platform of Artificial Intelligence (PAI), which is a distributed machine learning platform in Taobao, with thousands of NVIDIA Tesla P100. For better fitting the changes in the data distribution, we re-train the models every week by the automatic dispatch system of PAI.
- To reduce the computational complexity on RSP, we pre-extract and update multi-modal features of all micro-videos and products for Feature Database. By doing this, the SEMI transfer network on RSP would avoid time cost on feature extractor during online inference.

A.5 Feature Extractors.

As described in Section 3.2.1, we adopt two pre-trained multi-modal feature fusing models to extract features for micro-videos and products. Fig. 7 shows the general multi-modal feature extractor framework [11, 27]. Generally, the inputs include textual and visual information. The main network structures are multi-layer transformers. The pre-training tasks include supervised tasks (e.g., tag classifications) and self-supervised tasks (e.g., visual-textual information matching). The pre-training tasks can guide the model to extract and fuse the corresponding multi-modal information. For example, the task of tag classifications will force the model to focus on those multi-modal signals related to tags. To date, the kind of this framework has become the SOTA of multi-modal information fusing and the standard tool in Taobao.

For micro-videos (UniVL [27]), the inputs of the model are micro-video titles and frames. Pre-training tasks include masked language modeling, masked frame modeling, video-text alignment, and tag classifications. These tasks will guide the model to learn the semantic information of micro-video frames, titles, and tags. Finally, the output of the model is a 128-dimension embedding. We pre-train UniVL with about 100 million e-commerce micro-videos.

For products (UNITER [11]), the inputs of the model are regions of cover pictures and titles. Pre-training tasks include masked language modeling, masked region modeling, image-text matching, and category classifications. These tasks will guide the model to learn the semantic information of product cover pictures, titles, and categories. Finally, the output of the model is a 128-dimension embedding. We pre-train UNITER with about 100 million e-commerce products.

AliCoCo2: Commonsense Knowledge Extraction, Representation and Application in E-commerce

Xusheng Luo, Le Bo, Jinhang Wu, Lin Li, Zhiyi Luo, Yonghua Yang and Keping Yang

Alibaba Group, Hangzhou, China

{lxs140564,bole.bl,jinhang.wjh,guoba.ll,zhiyi.luo,huazai.yyh,shaoyao}@alibaba-inc.com

ABSTRACT

Commonsense knowledge used by humans while doing online shopping is valuable but difficult to be captured by existing systems running on e-commerce platforms. While construction of commonsense knowledge graphs in e-commerce is non-trivial, representation learning upon such graphs poses unique challenge compared to well-studied open-domain knowledge graphs (e.g., Freebase). By leveraging the commonsense knowledge and representation techniques, various applications in e-commerce can be benefited. Based on AliCoCo[16], the large-scale e-commerce concept net assisting a series of core businesses in Alibaba, we further enrich it with more commonsense relations and present **AliCoCo2**, the first commonsense knowledge graph constructed for e-commerce use. We propose a multi-task encoder-decoder framework to provide effective representations for nodes and edges from AliCoCo2. To explore the possibility of improving e-commerce businesses with commonsense knowledge, we apply newly mined commonsense relations and learned embeddings to e-commerce search engine and recommendation system in different ways. Experimental results demonstrate that our proposed representation learning method achieves state-of-the-art performance on the task of knowledge graph completion (KGC), and applications on search and recommendation indicate great potential value of the construction and use of commonsense knowledge graph in e-commerce. Besides, we propose an e-commerce QA task with a new benchmark during the construction of AliCoCo2, for testing machine common sense in e-commerce, which can benefit research community in exploring commonsense reasoning.

CCS CONCEPTS

• Information systems → E-commerce infrastructure; • Computing methodologies → Semantic networks.

KEYWORDS

Common Sense; Knowledge Graph Embedding; E-commerce

ACM Reference Format:

Xusheng Luo, Le Bo, Jinhang Wu, Lin Li, Zhiyi Luo, Yonghua Yang and Keping Yang. 2021. *AliCoCo2: Commonsense Knowledge Extraction, Representation and Application in E-commerce*. In *Proceedings of the 27th ACM*

SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3447548.3467203>

1 INTRODUCTION

Commonsense knowledge reflects our natural understanding of the world and human behaviors, which is shared by all humans. During online shopping, common sense such as “we should wear warmly when it’s getting cold”, “business-style T-shirts usually have pointed collars” and “a pregnant woman needs anti-slip shoes and anti-radiation dress” influence the shopping decisions we made. However, such knowledge is implicitly acknowledged among human beings through real-life experience, making it difficult for machines to capture and leverage.

In order to provide intelligent and user-friendly e-commerce search and recommendation services for customers, the large-scale concept net AliCoCo [16] is constructed in Alibaba, the largest Chinese e-commerce platform in the world. In AliCoCo, a four-layer structure is introduced to describe complicated user needs in e-commerce, where millions of *primitive concepts* with a well-defined type system form a fundamental language for e-commerce. In this paper, we systematically enrich AliCoCo with abundant commonsense relations in the layer of primitive concepts and introduce **AliCoCo2**, the first commonsense knowledge graph constructed for e-commerce use. Figure 1 is a snapshot of AliCoCo2, where commonsense knowledge in e-commerce are represented by short-phrase concepts and relations among them. There are different types of commonsense relations defined in the graph: “isa” relations between concepts and classes, “complementary” relations between two categories, and “mapping” relations such as “style_to_design” and “season_to_function”. Particularly, there are extra constraints for some relations, making conventional $\langle s, p, o \rangle$ (*subject-predicate-object*) triple change to a four-element tuple $\langle s, p, o, c \rangle$ (*c: constraint*). Take the concept “Business Style” (bold bounding box in the center of Figure 1) as an example, the tuple $\langle \text{business\_style}, r:\text{style\_to\_design}, \text{pointed\_collar}, T\text{-shirts} \rangle$ describes the common sense that a business-style T-shirt usually has a pointed collar. The category constraint can not be lost because business-style is not always mapped to pointed collar. For a briefcase, business-style means the bag should be horizontal.

Knowledge graphs in different domains have gradually been employed in many downstream applications, e.g., question answering [9, 15], search [6] and recommendation [17, 26]. However, the discrete representation of triples can not support these applications well [28]. Knowledge graph embedding (KGE) methods, which map the discrete entities and relations into continuous vector spaces,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '21, August 14–18, 2021, Virtual Event, Singapore

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8332-5/21/08...\$15.00

<https://doi.org/10.1145/3447548.3467203>

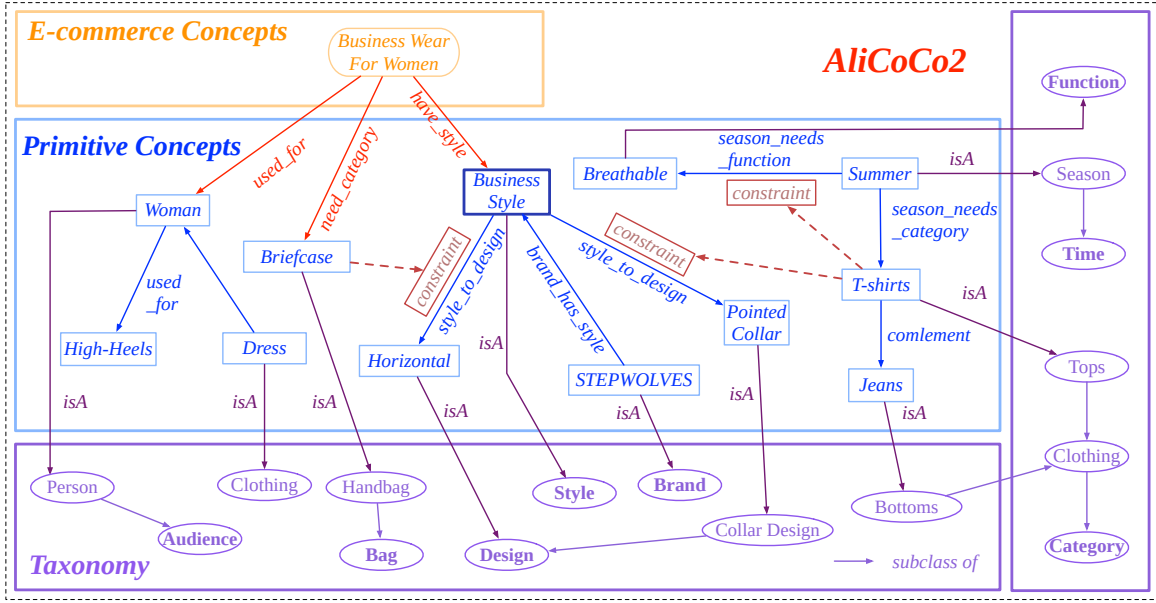


Figure 1: Overview of AliCoCo2, which mainly focuses on describing e-commerce commonsense knowledge in the layer of *Primitive Concepts*.

have recently emerged and been developed as a promising direction serving this purpose. The vector representations, i.e. embeddings, can preserve information in original triplets and can be used together with other machine learning techniques. However, most existing KGE models are targeting on open-domain factoid knowledge graphs such as Freebase. Relatively little work exists for commonsense knowledge graphs such as ConceptNet [21], not to mention e-commerce scenario. The key challenges in developing KGE models for commonsense knowledge graphs in e-commerce come from two perspectives: **1) Common Sense**: comparing to conventional knowledge graphs, commonsense knowledge graphs are much sparser since the nodes are usually represented by free-form phrases or even sentences. This conceptual diversity and expressiveness of the graph, implies that the number of nodes is orders of magnitude larger, and graphs are substantially sparser than conventional KBs. For instance, an encyclopedic KB like FB15K-237 [23] has 100x the density of ConceptNet [19]. **2) E-commerce**: as described above, the structures of commonsense relations are special when the knowledge graph is constructed for e-commerce use. For example, “subclass_of/isa” relations form the hierarchical structure, especially in category domain, so that general KGE models in euclidean space may lack the flexibility to embed tree-structured data properly. Another special relation structure is “n-ary” mapping relations between two concepts under some constraints. Existing KGE models based on binary relation structure can not be applied directly to such four-element structure, which poses unique challenge to embed the knowledge properly.

In this paper, we propose a multi-task encoder-decoder framework to embed nodes and relations of AliCoCo2 properly. To overcome the challenge of graph sparsity brought by “common sense”, we utilize large pre-trained language models to enhance semantic features and graph neural networks to capture structural features

in the encoding stage. For the challenge of unique and complicated relation structure brought by “e-commerce”, we leverage the idea of multi-task learning to model different types of commonsense relations separately in the decoding stage. The proposed method achieves state-of-the-art performance on knowledge base completion (KBC), the standard task for evaluating different KGE models. Based on AliCoCo2 and the embeddings learned from it, various downstream tasks in real e-commerce scenario can be benefited. Rich semantic relatedness carried in AliCoCo2 can help alleviate the semantic gap between query-item pair in the problem of search relevance, which is the core challenge of search engine in e-commerce. Commonsense relations like “complement” and “season_needs_xx” can help recommender systems jump out of historical behaviors to explore other implicit user interests. Item representation is critical for many e-commerce models, and relational knowledge contained in KGE embeddings learned from AliCoCo2 can help close the distances of those items with related but not identical properties in the embedding space and improve the quality of item embeddings. In this paper, We apply commonsense relations and embeddings learned from AliCoCo2 to three applications: query rewriting, search relevance and item representation and get promising experimental results.

We summarize the contributions of this paper as follows:

- We introduce AliCoCo2, the first commonsense knowledge graph in e-commerce. We define commonsense knowledge in e-commerce and show how to mine such relations in an effective way. During the construction of AliCoCo2, we propose a new challenging task of Question Answering in e-commerce, and release a new benchmark that can benefit the research community in exploring machine commonsense reasoning.

- We point out two main challenges of modeling KGE for commonsense knowledge graph in e-commerce, then we propose a multi-task encoder-decoder framework to effectively embed nodes and relations of the unique e-commerce commonsense knowledge graph. The performance on KGC task demonstrates the effectiveness of the method.
- We apply the mined commonsense knowledge and representation technique to downstream search and recommendation tasks in real industrial scenario and gain positive feedback, which may benefit further research on e-commerce common sense.

The rest of paper is organized as follows: First we formally define commonsense knowledge in e-commerce and give an overview of AliCoCo2 (Section 2). Next we introduce the way of mining various commonsense relations from text corpus via a form of machine reading comprehension (Section 3). Section 4 introduces a framework to embed nodes and edge of the commonsense KG. In Section 5, we apply the mined knowledge and learned embeddings in two e-commerce core businesses: search engine and recommender system. We report experimental results and give analysis in Section 6 and we give a conclusion in Section 7.

2 ALICOCO2

From AliCoCo to AliCoCo2, we define and enrich various commonsense relations in the layer of *Primitive Concepts* [16]. Commonsense relations in e-commerce can be categorized into two types: 1) Relations between two nodes of the same class, including *synonymy/hyponym* relations and *complementary* relations. 2) Relations between two nodes of different classes, such as “<winter, r:season, needs_material, cotton, c:jeans>”. Most of them describe a mapping relation between the head and tail concepts: *When people say the head concept, he actually means the tail concept*. In AliCoCo, the first type of relations are already constructed well enough, and we mainly focus on enriching the second type of relations in AliCoCo2. We define 91 kinds of meaningful commonsense relations across different classes and some examples are shown in Figure 1.

The main *structural* difference between e-commerce commonsense KG and traditional KG is that the former contains much more n-ary relations. N-ary relations also exist in traditional KG. For example, Freebase use *Mediators* as virtual nodes to connect multiple entities to represent n-ary relationships. Such structure brings extra difficulty to KGE modeling, so that two standard datasets FB15k and FB15k-237 replace one n-ary relation into n binary relations to make subsequent KGE modeling simpler. However, this way can not be used directly in AliCoCo2, since the meaning of three binary relations does not equal to the original relation. One possible way of split n-ary relations is to add new concepts and virtual relations (we discuss the details in Section 6.3). In this way, the semantic meaning remains the same but the size of nodes and edges increase dramatically, which increases the redundancy of the KG and brings extra computation and memory overload when training KGE models. Therefore, we remain the n-ary structure in AliCoCo2.

Type of informal term	Templates used to generate questions
Space	What kind of \$cate is used under the circumstances of \$term?
Crowd	What kind of \$cate is used for the group of \$term?
Time	What kind of \$cate is used when the time is \$term?
Function	What kind of \$cate has the functionality of \$term?
Event	What kind of \$cate do you need when you do \$term?

Table 1: Example templates used to transform informal queries to questions, where \$term represents the informal term and \$cate represents the category word.

3 EXTRACTION

In this section, we introduce an effective way to extract commonsense relations between two nodes of different classes under certain category constraint. Given pre-defined relation, head concept and constrained category, we transform the extraction of tail concepts to the task of identifying answer spans from the context [7, 13]. Let us revisit the “winter jeans” case, we transform it into a question like “What material of *jeans* is suitable for *winter*?”, then find answers from a shopping guidance saying “...*winter* is coming, it’s time to prepare warm *cotton* sweater...*cashmere* jeans for sale...”. This Question Answering (QA) formalization comes with several key advantages:

- The question encodes important information for the head concept and category constraint we focus on, and QA provides a natural way of jointly modeling entity, relation and other constraint;
- It allows us to exploit the well developed machine reading comprehension (MRC) models, which achieved significant improvements in recent years;
- The generated natural language questions can be directly presented to crowdsourcing workers, alleviating the difficulty of annotating raw commonsense relation triples.

The whole process to mine the commonsense relations is iterative and semi-automated. The most important step in practice is to collect a high-quality seed training dataset and we will describe this step in detail, including question-answer collection, annotation, and content extraction. We first collect question-answer pairs from users’ search-purchase behaviors and product reviews on Alibaba. Second, we annotate question-answer pairs through crowdsourcing efforts to generate a seed dataset. Then, related contents are extracted from product descriptions written by editors in e-commerce business. This way of building dataset generates a large number of candidates while the crowdsourcing annotation task is relatively simple, comparing to directly labeling triples from noisy long texts. After the seed dataset with related contents is constructed, a MRC model can be trained to predict more commonsense relations and continue to enhance the training process.

3.1 Question-Answer Collection

Abundant query logs and user behaviors in e-commerce platform is an initial source to mine seed relation instances. We leverage search logs and user behaviors including purchase and review to mine the correlations between concepts of different classes. Specifically, for each pair of query-behaved item, we establish possible relationship between the head concept in the query and the tail concept appearing in corresponding item title or reviews. Then we use conventional retrieval approach such as TF-IDF score to

calculate the degree of correlations. Note that only a small number of desired pairs can be mined in this way, since the cold start problem exists in search engine and we set a strict threshold to prune the candidates with low correlation scores. Once a seed dataset is constructed, we need to train models to extract more commonsense relationship from large-scale corpora.

Instead of asking people to find answers directly from long texts, crowdsourcing is used to judge whether the answer is plausible according to their common sense. Each candidate is a triple, consisting of head concept, product category and tail concept. We transform each candidate triple to a “YesNo” question according to the classes of head and tail concepts defined in AliCoCo2. Part of templates we used to generate questions are shown in Table 1. Then, we ask crowdsourcing annotators to judge whether the question is plausible according to common sense. The guideline for annotators are 1) The information described in the question fits most situations. 2) The expression of the subject is reasonable. 3) The question is fluent.

In order to control the quality of our dataset, we label 300 question-answer pairs and get the ground truth. The workers will be hired if they pass an examination on a small dataset, and the accuracy of workers’ annotation should be higher than 90%. For avoiding randomly-choose behaviors of annotators, the labeled question-answer pairs are distributed randomly to annotators to judge if they seriously participate in answering or not. Furthermore, we distribute each question to five people at least, and regard answers with a consistency rate of 80 or higher as valid.

3.2 Content Extraction

After accumulating a number of high-quality question-answer pairs, we search the large-scale product descriptions written by editors on Alibaba platform as passages to construct SQuAD-style dataset. The human-written product descriptions are in natural language form and contain abundant commonsense knowledge for online shopping. We collect a corpus containing 25.1M product-description pairs and 20.2M different descriptive contents. We extract the passage corresponding to a question-answer pair through matching the head concept to the product titles in the corpus, and the answer appears in the description of the product in the mean time. For each QA pair, 87.4 candidates of content can be matched, and up to more than 1000 candidates which contain a number of similar texts. We employ Jaccard and pre-trained language model to compute the similarity and the perplexity of each candidate, and retain 20 dissimilar high-quality contents as candidates. In total, we generate 420k distinct <question, contents, answer> triples as the seed dataset for following model training.

3.3 Reading Comprehension

The MRC model is based on BERT [5], the powerful pre-trained model which already achieved significant performance on MRC. After fine-tuning on a large-scale e-commerce corpora containing 20.3M product descriptions written by human editors, we train the MRC model on the training set. For each possible head concept from AliCoCo, we retrieve related products by title matching and generate questions using templates. Then the trained MRC model

automatically extracts possible answers from corresponding human-written product descriptions. The extracted candidate pairs are then distributed to crowdsourcing annotators, and the whole process runs iteratively.

4 REPRESENTATION

For knowledge representation learning, we formally define “E-commerce Commonsense Knowledge Graph (ECKG)” and “Knowledge Graph Embedding (KGE) for ECKG” as follows:

Definition 1. *E-commerce Commonsense Knowledge Graph (ECKG) is a triple $\langle E, R, I \rangle$, where: E is a finite set of all concept nodes in ECKG; R is a finite set of all relation names in ECKG; I is a finite set of relation instances with the form $\langle e_1, r, e_2, c \rangle$, where $e_1, e_2, c \in E$ and $r \in R$.*

Definition 2. *Given ECKG $\langle E, R, I \rangle$, the goal of Knowledge Graph Embedding (KGE) for ECKG is to represent each concept $e \in E$ and relation $r \in R$ into a low-dimensional vector space and the learned embeddings can improve performances of downstream tasks.*

We propose a multi-task encoder-decoder representation framework shown in Figure 2 to solve the challenges mentioned in the third paragraph of Section 1. To address the challenge of sparsity of commonsense KG, we use pre-trained language models to enhance contextual features and use graph neural networks to learn local graph structural features in the encoding stage. To address the challenge of special data structure in e-commerce, we adopt the idea of multi-task learning to decode different type of relations respectively using different models, then jointly optimize in training process.

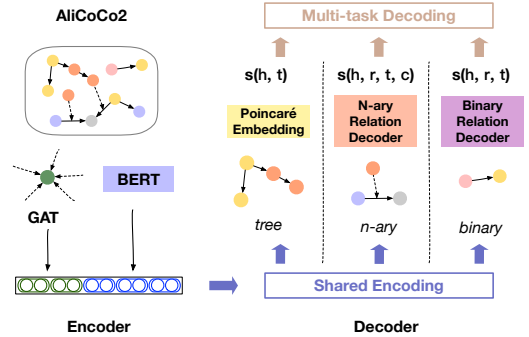


Figure 2: The proposed multi-task encoder-decoder framework for commonsense knowledge graph embedding in e-commerce.

4.1 Encoding from Language and Graph

To make the best use of semantic information from the textual description of nodes in the graph, we apply the widely used pretrained language model BERT [5] as the textual encoder. We finetune BERT with the masked language modeling loss to get rich semantic representations of all the concepts based on their text phrase. We list all the concept phrases as the input of finetuning in a format of $[CLS] + e' + [SEP]$, where e' is the textual phrase of $e \in E$. Then we use representations of the $[CLS]$ token from the last layer of the BERT model as node representations and obtain the language embedding matrix $T \in \mathbb{R}^{|E| \times M}$, where M is the dimension of BERT

embeddings. We use $\mathbf{t}_i$ to represent the semantic encoding for node e_i from the language model.

Graph Neural Networks (GNNs) aim to learn node embeddings that can capture structural information of the graph and it has become a fast-growing research area in recent years. A number of graph models are proposed including Graph Convolutional Networks (GCNs) [12], which achieves state-of-the-art classification results on a number of benchmark graph datasets. GCN is a multi-layer neural network that operates directly on a graph and induces embedding vectors of nodes based on properties of their neighborhoods. In practice, we use Graph Attention Networks (GATs) [25], a improved version of GCNs, where the information from the one-hop neighborhood is aggregated by self-attention layer. Given the graph G with R relation types and a GAT with L layers, the representation of a node e^l in layer $l + 1$ is computed as follows:

$$\mathbf{h}_i^{l+1} = \sigma(\sum_{r \in R} \sum_{j \in N_i} \alpha_{1r} \alpha_{2ij}^l \mathbf{W}^l \mathbf{h}_j^l + \mathbf{W}_0^l \mathbf{h}_i^l) \quad (1)$$

where N_i is the 1-hop neighbors of the node e_i in the graph. The second term represents the self-connection for the node, and is used to propagate information from one layer to the next. α_{1r} is the weight for the relation r and α_2 is the attention weight indicating the importance of each neighbor:

$$\beta_{ij}^l = \text{LeakyReLU}(\mathbf{W}_a[\mathbf{h}_i^l; \mathbf{h}_j^l]) \quad (2)$$

$$\alpha_{2ij}^l = \frac{\exp(\beta_{ij}^l)}{\sum_{k \in N_i} \exp(\beta_{ik}^l)} \quad (3)$$

where $\mathbf{W}_a$ is a shared attention matrix and $[\cdot]$ represents concatenation. The output of graph encoding is also an embedding matrix $\mathbf{H} \in \mathbf{R}^{|E| \times D}$, where D is the dimension of graph embeddings.

4.2 Decoding via Multi-task Learning

There are mainly three different types of relation structures to be modeled separately in AliCoCo2: hierarchical tree-structure, n-ary relation structure with category constraint, and normal triplet structure. We model these three structures respectively using three different models and decode them in a multi-task fashion.

For modeling normal binary relations, almost every existing model can be used as the decoder in our framework. We use the classical TransE [3] model here, and the probability of a given triple $\langle e_i, r, e_j \rangle$ is modeled as:

$$s(e_i, r, e_j) = \|\mathbf{e}_i + \mathbf{r} - \mathbf{e}_j\|_{l_1/2} \quad (4)$$

where $\mathbf{e}_i$ is the output of the encoding module:

$$\mathbf{e}_i = \text{MLP}([\mathbf{t}_i; \mathbf{h}_i]) \quad (5)$$

where $\mathbf{t}_i$ and $\mathbf{h}_i$ are semantic and graph embeddings for node e_i . Upon computing scores for all candidate tuples $s(e_i, r, e_j)$, we use a binary cross entropy loss to train the model. All relation instances in the training set are treated as positive samples and we randomly corrupt one node in a triple as negative samples.

Poincaré Embedding for Tree Structure

The hierarchical structure constituted by “subclass_of” and “isa” require additional modeling than other relation structures. To learn the distributed representations of the hierarchical categories, which are originally tree-structured symbols, we employ the Poincaré

embedding [4], which uses hyperbolic space such as the Poincaré Ball to represent hierarchy structures instead of Euclidean vector space. The main difference induced by operating on Poincaré Ball lies in the distance metric:

$$s(e_i, e_j) = \text{arccosh}(1 + 2 \frac{\|\mathbf{e}_i - \mathbf{e}_j\|_2^2}{(1 - \|\mathbf{e}_i\|_2^2)(1 - \|\mathbf{e}_j\|_2^2)}) \quad (6)$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are output embeddings of the encoding module for node e_i and e_j .

Decoding for N-ary Structure

To model the n-ary relation properly, we propose an especially designed decoder. Given a four element tuple $\langle e_i, r, e_j, c \rangle$, we first calculate a score for measuring the compatibility of node e with category c :

$$\text{sim}(e, c) = \sigma(\mathbf{W}_{sim}^T [\mathbf{t}; \mathbf{t}_c]) \quad (7)$$

where $\mathbf{t}_i$ and $\mathbf{t}_c$ are the semantic embeddings calculated from BERT and $\mathbf{W}_{sim} \in \mathbf{R}^{(M+D) \times 1}$ is a parameter matrix. In most cases, the head or tail concept node can be regarded as a possible property value of the category, so that this score measures the probability of e being a suitable property value for category c . Then the constrained head or tail concept representation is calculated as:

$$\mathbf{e}_c = \text{MLP}(\mathbf{A}^{(c_1)} \text{sim}(e, c) \cdot [\mathbf{t}; \mathbf{h}]) \quad (8)$$

where $\mathbf{A} \in \mathbf{R}^{|C_1| \times X \times (M+D)}$ is a parameter cube and C_1 is the collection of all first-level categories. c_1 is the first-level category of c , and $\mathbf{A}^{(c_1)}$ is the corresponding transformation matrix for c_1 . We choose first-level categories instead of leaf categories because the number of leaf nodes is around 100k, which is too large as the dimension of a parameter. Meanwhile, the number of first-level categories is smaller than 100, and categories within each first-level category are relatively similar so they can be modeled using the same embedding space. In this way, the relation between head and tail node can be modeled in the unique embedding space of current constrained category, and arbitrary binary decoder can be plugged into this framework to get the final probability.

Joint Training

While the encoding stage is shared across all tasks, we get different scoring functions in decoding stage, and each of them is treated as a unique task. For each training epoch, we randomly choose one task and sample batches from the corresponding dataset. The sample-then-training process is repeated until the metrics for each task stop improving on validation dataset.

5 APPLICATION

Various downstream e-commerce tasks can be benefited once a commonsense knowledge graph is constructed. In this section, we introduce detailed ways of applying the knowledge and the learned representations of AliCoCo2 in several e-commerce industrial tasks.

5.1 Query Rewriting

Query rewriting is the fundamental component of a search engine, aiming to explore potentially relevant documents to enlarge the retrieved candidate size for a following ranking step. One of the main challenges is that there exists a semantic gap between user queries and documents since they are usually described by different

language styles and vocabularies. This problem is more serious in e-commerce, due to the fact that language used by merchants to describe items is very professional, while users attend to express shopping needs by daily vocabulary and queries can be sometimes informal. Informal queries contain terms that hardly appear in the descriptions of items. For example, an informal user query can be “winter jeans (冬季牛仔褲)”, where “winter” is an informal term representing a direct user need but hardly appearing in the descriptions of items. In this case, terms used to describe relevant items are formal attributes such as “cashmere (羊絨)”, “cotton (棉花)” and “warm (保暖)”. According to a query analysis on Alibaba e-commerce platform, there are 12% queries still suffer from incomplete understanding of such informal terms.

The mined commonsense relations in AliCoCo2 can be directly applied to informal query rewriting since many informal terms can be linked to the head concepts of relation instances in the graph and the corresponding tail concepts are similar to formal attributes used to describe items. In detail, if an informal term in the query matches to some head concepts of relation instances in AliCoCo2, we rewrite the informal term as the corresponding tail concept terms. If the relation instance has category constraint, then the predicted category of the query should be matched. In this way, we add more rewriting query candidates in the recall phrase of the search engine, and it can retrieve more relevant items for a following ranking step.

5.2 Search Relevance

Relevance is the core problem of a search engine and the main challenge is the vocabulary gap between user queries and documents as well. Semantic matching is a key technique to bridge the gap in between to improve relevance. We apply concept embeddings learned by our KGE model (Section 4) to the deep relevance model, aiming to improve the model performance by introducing rich commonsense knowledge of AliCoCo2.

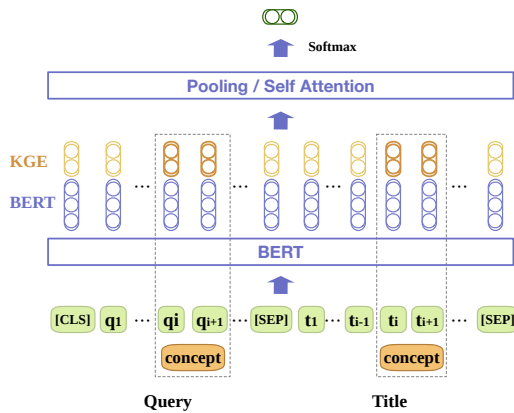


Figure 3: Commonsense knowledge enhanced deep semantic relevance model.

A typical deep semantic relevance model is shown in Figure 3, which is based on the BERT architecture. Concept embeddings are concatenated to the BERT word embeddings if the word in query

or title matches to the concept term. For those query or title words do not have a concept to match, an UNK embedding is concatenated.

5.3 Item Representation

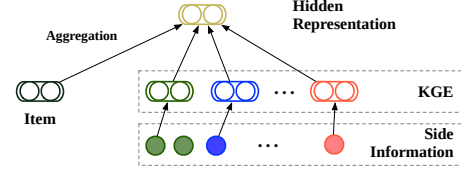


Figure 4: Commonsense knowledge enhanced graph embedding model.

Item embedding is widely used in the match stage of an industrial recommender system. By learning item embeddings via graph embedding (GE) methods like GraphSAGE [8], similar items are generated for each trigger item as a candidate set for a following ranking stage. Side information such as category and properties is important when training item embeddings in e-commerce scenario using GE models [27], since items with similar side information should be close in the embedding space. Shown in Figure 4, side information is usually modeled via extra feature aggregation in the GE model. In order to learn better item embeddings, we introduce commonsense knowledge from AliCoCo2 into the GE model based on the assumption that relational knowledge contained in KGE embeddings can help close the distances of those items with related but not identical properties in the embedding space. Therefore, initial embeddings for each item side information (category or property) are replaced with our KGE embeddings learned from AliCoCo2, and the embeddings can be finetuned in the training process.

6 EVALUATIONS

In this section, we first give a statistical comparison of AliCoCo2 and AliCoCo. Then we design and conduct experiments to answer the following questions:

- Does the QA task with the collected dataset mentioned in Section 3 suitable for testing machine common sense in e-commerce?
- How does the proposed KGE framework perform on the task of knowledge graph completion in e-commerce?
- Can commonsense relations and learned embeddings help improve downstream tasks in e-commerce?

6.1 AliCoCo2 vs. AliCoCo

Commonsense relations are enriched mainly in the layer of primitive concepts of AliCoCo, so we only report the numbers in this layer. The relations among primitive concepts can be naturally spread to the layer of e-commerce concepts. Table 2 shows the statistic of AliCoCo2. Comparing to AliCoCo, the number of primitive concepts covered by meaningful relations increase from 57, 125 to 163, 460. The number of relation types increase from 2 to 91 and the number of relation instances increase by **516%**. The density of the graph is largely improved so that AliCoCo2 has the ability to support more complicated downstream applications.

	AliCoCo	AliCoCo2
# primitive concepts*	57,125	163,460
# relation type	2	91
# relation instance	131,968	813,315

Table 2: Statistics of the primitive concept layer of AliCoCo2 at the time of writing, as a comparison to AliCoCo. * means concepts connected by relations other than “isa_(class)”.

6.2 E-commerce QA

The dataset generated in Section 3 can be used as a benchmark for testing machine common sense in e-commerce, we further clean the collected data, resulting to a dataset containing around 60k questions, 400k product descriptions as contents, and 90k distinct answers in total. We split it into training/validation/testing sets (8:1:1), ensuring that questions in validation or testing sets do not appear in training set. We named the dataset “com2qa” (common sense in e-commerce) and we are interested in the performance of existing MRC models on this dataset.¹ There are two types of question-answer pairs in com2qa dataset: squad-style type and YesNo type. Squad-style type contains single-span answer (60%) and no answer (10%), while YesNo type (30%) expects simply yes or no as answers. YesNo type of questions are said to be challenging [10] to existing MRC models so we sample a portion and add into com2qa, making our dataset more diverse. To quantify the difficulty level of our dataset and generate incremental question-answer pairs, we evaluate two types of the baseline MRC systems, and compare them with human performance respectively.

Method	SQuAD-style Type		YesNo Type
	EM	F1	F1
BiDAF[20]	52.1	54.3	-
QANet[31]	56.2	57.1	-
BERT	62.4	63.2	76.2
HUMAN	87.8	88.2	94.8

Table 3: Performance of different models on com2qa.

For SQuAD-style questions, we compare all three baselines together with human performance. We use two evaluation metrics similar to those used by SQuAD: Exact Match (EM) and F1 over words in the answers. The model performances on two types of questions are summarized in Table 3. We can see that model performances are still far from human judgements, showing that there is a large margin for machines to achieve the ability of judging commonsense reasoning as humans do on this e-commerce QA task. We hope this MRC benchmark can facilitate further researches on how to enhance the commonsense reasoning ability of models in e-commerce.

6.3 Knowledge Graph Completion

Knowledge Graph Completion (KGC) aims to infer missing facts by exploiting information already present in a knowledge graph, which is the standard task for evaluating the performances of different KGE models.

Datasets

¹The dataset can be found at <https://github.com/alicogintel/com2qa>

The main challenge of conduct KGC experiments on AliCoCo2 is that almost all the existing KGE models are developed based on KGs of binary relation structure, so that they can not be directly applied for n-ary relation structure in AliCoCo2. Therefore, we add virtual nodes and edges to our dataset (mentioned in the end of Section 2), in order to make the comparison as fair as possible. For each n-ary relation instance $\langle h, r, t, c \rangle$ in AliCoCo2, we replace it with 1 binary relation instance $\langle h_c, r, t_c \rangle$ and add additional 6 virtual binary edges shown in Table 4 with an example. The two newly added virtual relation are “constrain” and “modify”. In this way, the semantic information contained in the total 7 binary relations is equal to 1 n-ary relation.

$\langle h, r, t, c \rangle$	$\langle \text{business\_style}, \text{styleToCollar}, \text{pointed\_collar}, \text{T-shirts} \rangle$
$\langle h_c, r, t_c \rangle$	$\langle \text{business\_style}, \text{T-shirts}, \text{styleToCollar}, \text{pointed\_collar\_T-shirts} \rangle$
$\langle h, r_1, h_c \rangle$	$\langle \text{business\_style}, v\_constrain, \text{business\_style\_T-shirts} \rangle$
$\langle c, r_1, h_c \rangle$	$\langle \text{T-shirts}, v\_constrain, \text{business\_style\_T-shirts} \rangle$
$\langle t, r_1, t_c \rangle$	$\langle \text{pointed\_collar}, v\_constrain, \text{pointed\_collar\_T-shirts} \rangle$
$\langle c, r_1, t_c \rangle$	$\langle \text{T-shirts}, v\_constrain, \text{pointed\_collar\_T-shirts} \rangle$
$\langle h, r_2, c \rangle$	$\langle \text{business\_style}, v\_modify, \text{T-shirts} \rangle$
$\langle t, r_2, c \rangle$	$\langle \text{pointed\_collar}, v\_modify, \text{T-shirts} \rangle$

Table 4: Replacing 1 n-ary instance with 7 binary instances, where bold fonts are newly added virtual nodes and relations.

The scale of the complete AliCoCo2 is too large for efficient experiments, let alone a large number of virtual nodes and edges need to be added for fair comparison. Therefore, we split a small and clean dataset from AliCoCo2 for evaluating KGC task. We choose nodes and relation types with top popularity and then split the dataset into training/validation/testing sets by the following rules: 1) all the duplicated and reverse instances should be removed to avoid information leakage [23]; 2) make sure that all the nodes in validation/testing sets appear in the training set; 3) only add virtual edges in the training set. The statistic of the dataset is shown in Table 5, where 20% of relation instances are in n-ary structure and 10% are in tree structure.

	Graph	Train		Valid		Test	
	raw	raw	+virtual	raw	+virtual	raw	+virtual
# instances	79,192	64,516	129,037	2,260	-	2,325	-
# nodes	21,453	20,933	42,306	2,361	2,731	2,407	2,845
# relations	42	42	44	38	-	34	-

Table 5: Statistics of the dataset for commonsense knowledge graph completion in e-commerce. “+virtual” indicates the number after adding virtual nodes and edges.

Baselines and Metrics

We compare with various methods from several places, including the PyKEEN [1] repository which covers implementations of TransE [3], TransH [29], TransR[14], TransD [11], DistMult [30], and ComplEx [24], as well as the source code releases of ComplEx, RotatE [22], TuckER [2] and ComKGE [18]. Note that all the baseline models run on the dataset with virtual nodes and edges and only our model can directly deal with original data. We report MRR and Hits@K as previous works do, and we filter out all remaining nodes valid for a $\langle h, r \rangle$ pair from the ranking when computing scores for a gold target node. The scores are measured in both directions, where we compute the ranking of t given $\langle h, r \rangle$ and

the ranking of h given $\langle t, r^{-1} \rangle$. For n-ary relation instances, we predict t given $\langle h, r, c \rangle$.

Model	MRR	Hits@1	Hits@3	Hits@10	Hits@10 (-virtual)
TransE	0.099	0.035	0.112	0.263	0.223
TransH	0.137	0.058	0.153	0.320	0.296
TransR	0.137	0.072	0.163	0.314	0.282
TransD	0.155	0.076	0.158	0.311	0.287
RotatE	0.109	0.062	0.132	0.239	0.228
TuckER	0.067	0.032	0.095	0.226	0.202
DistMult	0.103	0.042	0.112	0.273	0.252
ComKGE	0.272	0.175	0.312	0.479	0.369
Ours	0.358	0.228	0.479	0.577	-
ComKGE (tree)	0.282	0.178	0.313	0.447	-
Ours (tree)	0.342	0.198	0.395	0.568	-
ComKGE (n-ary)	0.228	0.108	0.300	0.431	-
Ours (n-ary)	0.474	0.331	0.569	0.805	-

Table 6: Experimental results of different KGE models on testing set. The last four lines show the results on tree and n-ary structural instances.

Results

Table 6 shows the performances of different KGE models. Our proposed method out-performed all the baselines by a large margin. The performances of ComKGE is clearly better than other baselines, indicating the effectiveness of rich semantic features learned from BERT and local structural features. Our method continuously outperforms ComKGE, demonstrating the superiority of modeling different relation structures respectively and the last four lines in Table 6 show that our joint optimization is able to improve each single performance on tree structure and n-ary structure. We also conduct experiments by removing all six virtual edges for each n-ary relation instance in the training set. The rightmost column in Table 6 shows the performances drop if the virtual edges are removed, indicating that virtual edges do carry important information and are necessary to make the comparison fair enough.

6.4 Query Rewriting

We integrate about 100k mapping relations from AliCoCo2 into the online query rewriting module in Alibaba’s search engine with a standard A/B testing configuration. Comparing to the baseline without informal query rewriting, online results show that the new strategy achieves higher sales volume under the hit queries. Gross Merchandise Volume (GMV) is improved by **0.6%**² on average in a continuous 10 days, which demonstrates that newly mined relations are able to help retrieve more relevant products.

6.5 Search Relevance

We conduct experiments on a human labeled query-item relevance dataset, consisting of 286,299 training samples and 22,545 testing samples. Experimental results are shown in Table 7. AUC is improved by 1.78% after integrating node embeddings learned by our KGE model into the baseline model, which demonstrates the effectiveness of knowledge representation learning. Comparing to a pooling operation after embedding concatenation, using a self-attention layer further increases AUC to 0.8556, resulting in an

improvement of 2.03%. indicating the interaction of learned embeddings is meaningful and able to better represent the query or the title.

Model	AUC	Improvement
Baseline	0.8386	-
+ KGE	0.8536	1.78%
+ KGE + SelfAtt.	0.8556	2.03%

Table 7: Experimental results on the task of search relevance.

We also look into some cases in the testing set that are solved after commonsense knowledge is introduced. The baseline model fails to predict the relevance between “query: 冰丝裤女夏季 (ice silk pants woman **summer**)” and “title: 空气阔腿裤女薄款高腰坠垂感缎面醋酸铜氨丝直筒冰丝雪纺凉凉云朵裤 (air wide-leg pants women’s **thin** high-waist drape satin acetate cupramide silk straight ice silk chiffon cool cloud pants)” since information of “夏季 (summer)” in the query is not explicitly included in the title. After commonsense knowledge is introduced, the model is able to correctly discover the relevance of this pair since we wear “thin” pants in “summer” in our common sense and such knowledge is contained in AliCoCo2.

6.6 Item Representation

We evaluate the quality of learned item embeddings using a recommendation task and report Hit Rate (HR) as evaluation metric. We use the last day’s user behaviors as trigger items to generate similar items as recommended predictions using the learned item embeddings and then test on the next day’s true user behaviors. Results are shown in Table 8. Comparing to the raw GraphSAGE, adding side information improves the performance by 5.70%. By integrating the commonsense knowledge graph embeddings, HR is continuously improved by 8.81%, demonstrating the effectiveness of the commonsense knowledge contained in our KGE model.

Model	HR	Improvement
GraphSAGE	0.0193	-
+ SideInfo.	0.0204	5.70%
+ SideInfo. + KGE	0.0210	8.81%

Table 8: Experimental results on the task of recommendation.

7 CONCLUSION

In this paper, we continue to enrich AliCoCo, the large-scale e-commerce concept net assisting various core businesses in Alibaba, by focusing on the extraction, representation and application of commonsense knowledge in e-commerce. We investigate potential effectiveness of commonsense knowledge in several real applications, then define commonsense relations based on the schema of AliCoCo. We mine commonsense relation instances from multiple sources and present AliCoCo2, the first commonsense knowledge graph in e-commerce. We then propose a novel framework to effectively embed each node and edge into low-dimensional space and validate its superiority in the task of knowledge graph completion. For two core e-commerce scenarios: search and recommendation, we adopt three detailed tasks either offline or online to demonstrate

²Any online improvement larger than 0.5% is considered as meaningful

the use of mined commonsense knowledge and corresponding representations. Experimental results show positive feedback, indicating the great value of the construction and use of commonsense knowledge graph in e-commerce. In the future, we will continue to explore better ways of incorporating commonsense knowledge into downstream tasks, to fulfill the goal of letting machines have the ability of commonsense reasoning in industrial scenarios.

REFERENCES

- [1] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2020. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *arXiv preprint arXiv:2007.14175* (2020).
- [2] Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590* (2019).
- [3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.
- [4] Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. 2017. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359* (2017).
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [6] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 601–610.
- [7] Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019. Question Answering is a Format; When is it Useful? *arXiv preprint arXiv:1909.11291* (2019).
- [8] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.
- [9] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 221–231.
- [10] Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, et al. 2017. Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint arXiv:1711.05073* (2017).
- [11] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*. 687–696.
- [12] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [13] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. *arXiv preprint arXiv:1905.05529* (2019).
- [14] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion.. In *AAAI*, Vol. 15. 2181–2187.
- [15] Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu. 2018. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2185–2194.
- [16] Xusheng Luo, Luxin Liu, Yonghua Yang, Le Bo, Yuanpeng Cao, Jinghang Wu, Qiang Li, Keping Yang, and Kenny Q Zhu. 2020. AliCoCo: Alibaba E-commerce Cognitive Concept Net. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 313–327.
- [17] Xusheng Luo, Yonghua Yang, Kenny Qili Zhu, Yu Gong, and Keping Yang. 2019. Conceptualize and Infer User Needs in E-commerce. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 2517–2525.
- [18] Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2019. Exploiting structural and semantic context for commonsense knowledge base completion. *arXiv preprint arXiv:1910.02915* (2019).
- [19] Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense Knowledge Base Completion with Structural and Semantic Context.. In *AAAI*. 2925–2933.
- [20] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).
- [21] Robert Speer and Catherine Havasi. 2013. ConceptNet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP*. Springer, 161–176.
- [22] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* (2019).
- [23] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. 57–66.
- [24] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. *International Conference on Machine Learning (ICML)*.
- [25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [26] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippletNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 417–426.
- [27] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 839–848.
- [28] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.
- [29] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes.. In *Aaai*, Vol. 14. Citeseer, 1112–1119.
- [30] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).
- [31] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541* (2018).

Live-Streaming Fraud Detection: A Heterogeneous Graph Neural Network Approach

Zhao Li*
lizhao.lz@alibaba-inc.com
Alibaba Group, China

Haishuai Wang*
hwang@fairfield.edu
Fairfield University, USA

Peng Zhang†
zhangpeng04@gmail.com
Alibaba Group, China

Pengrui Hui
pengrui.hpr@alibaba-inc.com
Alibaba Group, China

Jiaming Huang
jimmy.hjm@alibaba-inc.com
Alibaba Group, China

Jian Liao
jian.liaoj@alibaba-inc.com
Alibaba Group, China

Ji Zhang
Ji.Zhang@usq.edu.au
The University of Southern
Queensland, Australia

Jiajun Bu
bjj@zju.edu.cn
Zhejiang University, China

ABSTRACT

Live-streaming platforms have recently gained significant popularity by attracting an increasing number of young users and have become a very promising form of online shopping. Similar to the traditional online shopping platforms such as Taobao, live-streaming platforms also suffer from online malicious fraudulent behaviors where many transactions are not genuine. The existing anti-fraud models proposed to recognize fraudulent transactions on traditional online shopping platforms are inapplicable on live-streaming platforms. This is mainly because live-streaming platforms are characterized by a unique type of heterogeneous live-streaming networks where multiple heterogeneous types of nodes such as users, live-streamers, and products are connected with multiple different types of edges associated with edge features. In this paper, we propose a new approach based on a heterogeneous graph neural network for Live-streaming Fraud detection (called **LIFE**). LIFE designs an innovative heterogeneous graph learning model that fully utilizes various heterogeneous information of shopping transactions, users, streamers, and items from a given live-streaming platform. Moreover, a label propagation algorithm is employed within our LIFE framework to handle the limited number of labeled fraudulent transactions for model training. Extensive experimental results on a large-scale Taobao live-streaming platform demonstrate that the proposed method is superior to the baseline models in terms of fraud detection effectiveness on live-streaming platforms. Furthermore, we conduct a case study to show that the proposed method

is able to effectively detect fraud communities for live-streaming e-commerce platforms.

CCS CONCEPTS

• **Applied computing** → **Secure online transactions**.

KEYWORDS

Fraud detection, Heterogeneous graph neural networks, Live-streaming.

1 INTRODUCTION

The recent rapid development of live-streaming technologies creates new opportunities for online business and the traditional online e-commerce platforms such as *Taobao* have been heavily investing live-streaming applications. Live-streaming enables streamers, who are typically the popular online celebrities or Key Opinion Leaders (KOLs), to harness the power of live media to sell products and/or services worldwide in real time with enhanced engagement and interaction in consumers' shopping experiences [9, 13]. After the outbreak of COVID-19 particularly, live-streaming platforms have enjoyed unprecedented growth in the number of users as well as their revenues. For example, the annual financial report from Alibaba Group shows that the Gross Merchandise Volume (GMV) of Alibaba's live-streaming platform *Taobao Live*¹ in the calendar year of 2020 reaches a staggering figure of 60 billion dollars.

However, live-streaming platforms face a variety of threats from malicious fraudulent transactions which might compromise the fairness and security of their business ecosystem. As shown in Figure 1, a fraudulent streamer is broadcasting her products, while spam accounts and fraudulent users are manipulating to inject a vast amount of malicious traffic to the live room, giving "like" comments, making fake reviews, and creating unrealistic followers with fake purchase transactions. Moreover, some streamers provide pornography or non-compliant contents to attract more customers and receive more exposures [6]. The propose of executing these fraudulent behaviors is to mislead e-commerce systems in calculating ranking scores or reputation factors for certain live-streamers.

*The first two authors contributed equally.

†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '21, August 14–18, 2021, Virtual Event, Singapore

© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08...\$15.00
<https://doi.org/10.1145/3447548.3467065>

¹<https://taobaolive.taobao.com/>



Figure 1: An example of a fraudulent streamer on the Taobao live-streaming platform. Three types of fraudulent behaviors are shown: (a) malicious behaviors from auto-controlled terminal devices, (b) fake “like” and comments from a special equipment, and (c) fraudulent transactions conducted by an organized team.

Recent years have witnessed a growing interest in utilizing the graph information with graph neural networks for fraud detection [14, 26, 29]. Unfortunately, the existing fraud detection models built on traditional e-commerce platforms are inapplicable to deal with live-streaming scenarios due to three major challenges:

- **Heterogeneous tripartite networks.** There are three types of nodes in a live-streaming network, i.e., users, streamers and items. Compared with bipartite networks where only users and items are considered, the tripartite live-streaming network requires a new network feature learning schema. In addition, the tripartite heterogeneous networks consist of different types and features of both nodes and edges;
- **Rich social information.** Live-streaming is by nature a social network, where users can thumb up a streamer, post comments, and discuss in the streamer’s room in real-time. As shown in Figure 2, the nodes of users, streamers and items are connected with multiple types of edges, such as users follow streamers, users like streamers, streamers promote items, and users buy items through a streamer;
- **Limited domain knowledge.** The labeling of nodes and edges on live-streaming is challenging as most existing knowledge on labeling “black seeds” (i.e., typical fraudulent users) on e-commerce platforms is inapplicable to live-streaming.

In this paper, we present a novel anti-fraud model for live-streaming platforms using a heterogeneous graph neural network approach. Specifically, we propose a new model, called LIFE (Live-streaming Fraud detection), to detect fraudulent transactions in a tripartite live-streaming network. LIFE first learns embeddings of both nodes and edges associated with features and types, and then builds a fully-connected layer on top of the concatenation of all the embeddings as the fraud classifier. The embeddings include the nodes of users, streamers and items, the edges between each pair of nodes, and shopping transactions. Experimental results demonstrate the performance of LIFE on a large-scale Taobao Live-streaming dataset. The contributions of this paper are summarized as follows:

- To our best knowledge, this is the first reported effort in literature to study the problem of live-streaming fraud detection using Graph Neural Networks;
- A new method, called LIFE, is proposed for live-streaming fraud detection. Built on a tripartite heterogeneous network, LIFE conducts embedding learning of both nodes and edges in the heterogeneous network and builds a fraud classifier on top of the concatenation of all the embeddings;
- As a byproduct, a new label propagation approach is presented to address the challenge of limited number of labeled “black seeds” for training our model;
- Extensive experiments are conducted on a real-world ten-million-scale live-streaming dataset. Our method significantly outperforms a couple of baseline approaches. Furthermore, specific fraudulent communities are revealed based on physical location tracing, which contributes to the overall safety and health of live-streaming ecosystem.

The rest of this paper is organized as follows. Section 2 surveys the related work in the area of fraud detection and graph neural networks. Section 3 presents in detail our innovative fraud detection method, LIFE, for live-streaming platforms. Section 4 reports the results of experimental evaluation as well as a case study in real-world live-streaming applications. Section 5 draws the conclusion.

2 RELATED WORK

Most existing fraud detection on e-commerce platforms mainly focus on automatically learning the intrinsic patterns from historic fraud data based on machine learning models [4, 15, 19, 32]. Oentaryo *et al.* [16] first extracted statistical features from multiple sources such as user’s profiles and historical behaviours and then use classical classifiers for fraud detection. Dhankhad *et al.* [2] identified the most important features from online payment data for fraudulent transaction detection. Cao *et al.* [1] advanced the technique of aggregating behavioral patterns to find out fraudulent accounts in an unsupervised machine learning framework. A sybil defense based fraud detection scheme SD2 was proposed [22] and this work further designed an unsupervised fraud detection strategy named UNIK to make it highly robust against spam attacks. However, these pattern-centric approaches ignore relations between users and items which are critical in detecting frauds because fraudulent transactions are often detected in groups.

Given that observation, some researchers have leveraged rich network information for fraud detection [14, 26, 29]. Recent years have seen the graph neural networks gaining prominent progress for network representation learning by passing and aggregating messages [3], such as graph convolutional networks (GCN) [10], GraphSAGE [5], and graph attention networks (GAT) [24]. GNN aggregates features from one-hop neighbors by the graph convolution operator. GAT introduces attention mechanism into GCN that allows information aggregation using the most relevant neighbors (i.e., high ranking attention scores). GraphSAGE is an inductive framework that utilizes efficient sampling and aggregation methods to learn node embeddings. The work [8] first extracted features based on graph properties and then built classifiers to identify fraudulent online accounts. Shchur *et al.* [18] formulated the fraud detection problem in temporal graphs and used dense sub-block discovery to detect suspicious activities. A graph neural network framework

for malicious accounts detection in heterogeneous networks was proposed in [12]. Wang *et al.* [26] made a comprehensive analysis of fraud activities and presented FdGars, a GCN-based approach for fraudster detection in online app review system. A multi-view graph convolution network model was proposed [11] to predict hidden fraudulent links.

Most of the graph neural networks work well on homogeneous graphs while do not have the capability of learning the heterogeneity in graphs. Heterogeneous graph neural networks have been proposed to learn from heterogeneous graphs. Zhang *et al.* [31] presented heterogeneous graph neural network (HetGNN) that adopts a node type-aware neighborhood aggregation by different RNNs to integrate multi-modal features. Heterogeneous attention network (HAN) [27] extended GAT by maintaining different weights for different meta-path-defined edges. HAN also use high-level semantic attention to differentiate and aggregate information from different meta paths. Hu *et al.* [7] proposed heterogeneous graph transformer (HGT) that applies the meta relations of heterogeneous graphs to parameterize weight matrices for the heterogeneous mutual attention, message passing, and propagation steps.

Although heterogeneous graphs have been widely applied in e-commerce applications such as spam detection [26], malicious users detection [12], anti-money laundry [28], there have been only a few studies for fraud detection using various types of nodes and edges in heterogeneous graphs. Rao *et al.* [17] employed heterogeneous graph transformer to incorporate heterogeneous mutual attention and heterogeneous message passing with key, value and query vector operations.

However, all the above fraud detection methods do not fit our problem setting in live-streaming fraud detection because we aim to detect frauds at the transaction level that have more fine-grained information in the heterogeneous graph such as the edge and transaction features and their contextual information. Therefore, they cannot detect the existence of frauds in live-streaming scenarios.

3 PROPOSED METHOD

In this section, we present the details of our model, named LIFE, for fraud detection online-streaming platforms. We first introduce the preliminaries focusing on heterogeneous graphs. We then discuss the construction of the live-streaming graph, the key heterogeneous graph for modeling large-scale live-streaming applications. Effective strategies of embedding-based representation learning on the live-streaming graph are also presented, which supports the classifier based on the concatenation of all the embeddings. Finally, an innovative method is proposed to deal with the problem of a limited number of labeled training samples.

3.1 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, E, X, \mathcal{R}, \phi, \varphi)$ be a heterogeneous tripartite network, where each node $v \in \mathcal{V}$ is associated with a node feature vector and one particular type $\phi(v)$, while each edge $e \in E$ is associated with an edge feature vector and one particular type $\varphi(e)$. In the heterogeneous tripartite network, two nodes are linked through multiple types of edges [20, 21]. Each node v of type $\phi(v)$ is associated with attribute X_v , and each edge $e = (v, v')$ of type $\varphi(e)$ connecting two neighboring nodes v and v' has an attribute X_e . There is a tuple $\langle \phi(v), \varphi(e), \phi(v') \rangle$ based on the edge type.

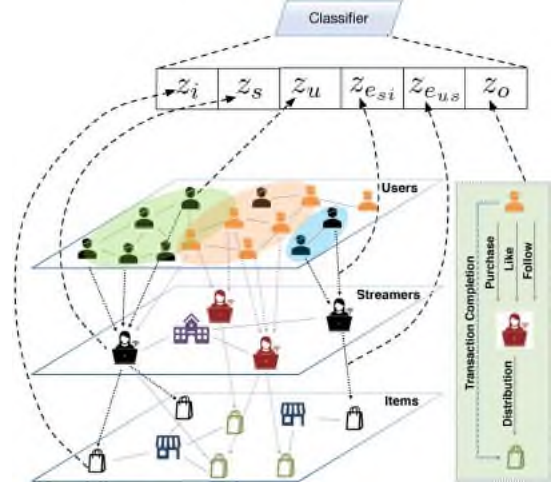


Figure 2: An illustration of LIFE on a tripartite live-streaming network. There are three types of nodes and edges. LIFE learns user embedding z_u , streamer embedding z_s , item embedding z_i , and edge embedding $z_{e_{si}}$ of edge e_{si} , and edge embedding $z_{e_{us}}$ of edge e_{us} . Moreover, transaction embedding z_o is extracted from each purchase transaction. A fraud classifier is built on top of the concatenation of all the embeddings, i.e., $y = \text{classifier}(\text{concat}(z_o, z_i, z_s, z_u, z_{e_{us}}, z_{e_{si}}))$.

In general, for a graph neural network with L layers, the l -th layer ($l = 1, 2, \dots, L$) of edge e can be written as:

$$H_e^l = \sigma \left(W^l \cdot \text{AGG}(H_e^{l-1}, H_v^{l-1}, H_{v'}^{l-1}) \right), \quad (1)$$

where v' is the node adjacent to v through edge $e = (v, v')$, AGG is a function used for aggregating the neighborhood message by the attention weight, which can be customized by specific models (e.g., max-pooling, mean-pooling) to calculate the score (or importance) of each neighboring node.

3.2 Live-streaming Graph Construction

In this paper, we aim to identify fraudulent transactions on live-streaming platforms, which by nature can be formulated to an edge classification problem on a directed heterogeneous tripartite graph with attributed nodes and edges.

As shown in Figure 2, we first construct a heterogeneous tripartite graph to model live-streaming e-commerce scenarios. In such a heterogeneous graph which consists of multiple types of nodes and links, a node $v \in \mathcal{V}$ has a type $\phi(v) \in \mathcal{A}$, where $\mathcal{A} := \{\text{user}, \text{streamer}, \text{item}\}$, referring to users, streamers and items, respectively. We use U to denote a set of users, I to denote a set of items, S to denote a set of streamers, E_{us} to denote a set of edges between U and S , E_{si} to denote a set of edges between S and I . An edge $e_{us} \in E_{us}$ from a user $u \in U$ to a streamer $s \in S$ exists if u makes interactions (i.e., 'likes', 'buy', etc.) to s , and an edge $e_{si} \in E_{si}$ from an item $i \in I$ to a streamer $s \in S$ exists if s helps to promote the product sale for i . If e_{us} or e_{si} or the user u is labeled as being fraudulent, this transaction is considered as a fraud, otherwise it is legitimate. This heterogeneous graph $G = (U, S, I, E_{us}, E_{si})$ is named a **Live-streaming Graph**.

Given a node $u \in U$ or $s \in S$ or $i \in I$, let $N(u)$, $N(s)$, $N(i)$ be the set of nodes in node u 's, s 's, i 's one-hop neighbors, i.e., $N(u) = \{u' \in U | (u, u') \in E_{us}\}$, $N(s) = \{s' \in S | (s, s') \in E_{us} \cup E_{si}\}$, $N(i) = \{i' \in I | (i, i') \in E_{si}\}$, respectively.

In the Live-streaming Graph G , it includes edge types, node types, node features and edge features. In addition, we have purchase transactions that include the contextual information of purchases, such as order price, order time, IP and payment type.

3.3 Live-streaming Graph Representation Learning

Our model aims to identify fraud transactions between users and items. This is equivalent to classifying edges between users and items through streamers. To this end, effective graph representation learning is needed that embeds the heterogeneous live-streaming graph and the contextual information of purchase transactions into a latent space such that the representations of the fraudulent purchase transactions are as similar as possible, while the embeddings of the normal users are distributed uniformly in the remaining latent space. The major challenge of learning representations in the live-streaming graphs lies in capturing the heterogeneous structure information from multiple types of nodes and edges.

In our model, embedding of various nodes and edges in the graph are involved. Specifically, let $z_{e_{us}}$ be the embedding of a user-streamer edge, $z_{e_{si}}$ be the embedding of a streamer-item edge, $z(u)$ be the embedding of a user, $z(s)$ be the embedding of a streamer, and $z(i)$ be the embedding of an item. Note that the above embeddings are aggregations of all the L -layers hidden states, i.e., $z_{e_{us}} = H_{e_{us}}^L$, $z_{e_{si}} = H_{e_{si}}^L$, $z_u = H_u^L$, $z_s = H_s^L$, $z_i = H_i^L$ respectively. To fit the general framework Eq. (1) to the Live-streaming Graph, three types of aggregation functions are defined for edges, streamers and items.

For edges e_{us} and e_{si} , the hidden states are updated as the concatenation of the previous hidden states of the edges and two nodes it links to. Then, the hidden state of $H_{e_{us}}^l$ and $H_{e_{si}}^l$ in the l^{th} layer are defined as

$$H_{e_{us}}^l = \sigma \left(W_{e_{us}}^l \cdot AGG_{e_{us}}^l(H_{e_{us}}^{l-1}, H_{U(e_{us})}^{l-1}, H_{S(e_{us})}^{l-1}) \right), \quad (2)$$

$$H_{e_{si}}^l = \sigma \left(W_{e_{si}}^l \cdot AGG_{e_{si}}^l(H_{e_{si}}^{l-1}, H_{S(e_{si})}^{l-1}, H_{I(e_{si})}^{l-1}) \right), \quad (3)$$

where the aggregation functions $AGG_{e_{us}}^l$ and $AGG_{e_{si}}^l$ are concatenation $concat(\cdot)$.

For each user node $u \in U$, streamer node $s \in S$, and item node $i \in I$, we aggregate messages from their respective neighboring nodes and the associated edges. Then, the aggregated embedding $H_{N(u)}^l$, $H_{N(s)}^l$ and $H_{N(i)}^l$ are calculated as follows

$$H_{N(u)}^l = \sigma \left(W_u^l \cdot AGG_U^l(\mathcal{H}_{UE}^{l-1}) \right), \quad (4)$$

$$H_{N(s)}^l = \sigma \left(W_s^l \cdot AGG_S^l(\mathcal{H}_{SE}^{l-1}) \right), \quad (5)$$

$$H_{N(i)}^l = \sigma \left(W_i^l \cdot AGG_I^l(\mathcal{H}_{IE}^{l-1}) \right), \quad (6)$$

where

$$\mathcal{H}_{UE}^{l-1} = \left\{ concat(H_u^{l-1}, H_{e_{us}}^{l-1}), \forall e_{us} = (u, s) \in E_{us} \right\}, \quad (7)$$

$$\mathcal{H}_{SE}^{l-1} = \left\{ concat(H_s^{l-1}, H_{e_{us}}^{l-1}, H_{e_{si}}^{l-1}), \forall e_{us} = (u, s) \in E_{us}, \forall e_{si} = (s, i) \in E_{si} \right\}, \quad (8)$$

$$\mathcal{H}_{IE}^{l-1} = \left\{ concat(H_i^{l-1}, H_{e_{si}}^{l-1}), \forall e_{si} = (s, i) \in E_{si} \right\}, \quad (9)$$

The parameters W_u^l , W_s^l and W_i^l are defined by three types of nodes (i.e., users, streamers and items). During layer-wise message propagation, we use attention mechanism. Based on the recent work on Transformer [23] and Heterogeneous Graph Transformer [7], we use heterogeneous mutual attention given that different types of nodes have different feature distributions. Given a target node v' , and all its neighbors $v \in N(v')$, we calculate the mutual attention with respect to edge type, i.e., the triplet of $\langle \phi(v), \psi(e), \phi(v') \rangle$. We map each target node v' into a query vector representation, and source node v into a key vector representation in order to calculate the attention scores by their dot products. For the $(l-1)$ -th layer, the query vector for target node v' and the key vector for source node v can be calculated by Eqs. (10) and (11),

$$Q_{v'} = Q-Linear_{\phi(v')} H_{v'}^{l-1}, \quad (10)$$

$$K_v = K-Linear_{\phi(v)} H_v^{l-1}, \quad (11)$$

Based on the key vector and the query vector in Eqs. (10) and (11), we can calculate $AGG_U^l(\mathcal{H}_{UE}^{l-1})$, $AGG_S^l(\mathcal{H}_{SE}^{l-1})$ and $AGG_I^l(\mathcal{H}_{IE}^{l-1})$.

For $AGG_U^l(\mathcal{H}_{UE}^{l-1})$, the target node v' is from the set of users U (i.e., $v' \in U$), and the source node v is from all its neighbors in the set of streamers S (i.e., $v \in S$). Then, we have

$$\begin{aligned} AGG_U^l(\mathcal{H}_{UE}^{l-1}) &= ATT_{N_u} \left(H_u^{l-1}, \mathcal{H}_{UE} \right) \\ &= Softmax_{\forall v \in N(v')} \left(\left(K_v W_{\phi(e_{us})}^{ATT} Q_{v'}^T \right) \cdot \frac{\mu \langle \phi(v), \psi(e_{us}), \phi(v') \rangle}{\sqrt{d}} \right), \end{aligned} \quad (12)$$

For $AGG_S^l(\mathcal{H}_{SE}^{l-1})$, the target node v' is from the set of streamers S (i.e., $v' \in S$), the source node v is from all its neighbors in the sets of users U and items I (i.e., $v \in S \cup I$), and the edge $e \in e_{us} \cup e_{si}$ ($\forall e_{us} = (u, s) \in E_{us}, \forall e_{si} = (s, i) \in E_{si}$). Then we have

$$\begin{aligned} AGG_S^l(\mathcal{H}_{SE}^{l-1}) &= ATT_{N_s} \left(H_s^{l-1}, \mathcal{H}_{SE} \right) \\ &= Softmax_{\forall v \in N(v')} \left(\left(K_v W_{\phi(e)}^{ATT} Q_{v'}^T \right) \cdot \frac{\mu \langle \phi(v), \psi(e), \phi(v') \rangle}{\sqrt{d}} \right), \end{aligned} \quad (13)$$

For $AGG_I^l(\mathcal{H}_{IE}^{l-1})$, the target node v' is from the set of items I (i.e., $v' \in I$), and the source node v is from all its neighbors in the set of streamers S (i.e., $v \in S$). Then we have

$$\begin{aligned} AGG_I^l(\mathcal{H}_{IE}^{l-1}) &= ATT_{N_i} \left(H_i^{l-1}, \mathcal{H}_{IE} \right) \\ &= Softmax_{\forall v \in N(v')} \left(\left(K_v W_{\phi(e_{si})}^{ATT} Q_{v'}^T \right) \cdot \frac{\mu \langle \phi(v), \psi(e_{si}), \phi(v') \rangle}{\sqrt{d}} \right), \end{aligned} \quad (14)$$

where $W_{\phi(e_{us})}^{ATT}$ and $W_{\phi(e_{us})}^{ATT}$ are distinct edge-based matrix for different edge types, d is the dimension of K_v , and μ is a prior tensor used to denote the general significance of each edge type. Since different node types contribute differently to the target node, μ is considered to be an adaptive scaling to the attention. Furthermore,

to improve stability of the learning process, we can leverage the strength of multi-head attentions. We compute multiple different attention maps and finally aggregate all the learned representations.

After aggregating neighbors' information of various types of nodes in the graph, we use a combination strategy as in GraphSAGE [5] for the node representation:

$$H_u^l = \text{concat} \left(M_u^l \cdot H_u^{l-1}, H_{N(u)}^l \right), \quad (15)$$

$$H_s^l = \text{concat} \left(M_s^l \cdot H_s^{l-1}, H_{N(s)}^l \right), \quad (16)$$

$$H_i^l = \text{concat} \left(M_i^l \cdot H_i^{l-1}, H_{N(i)}^l \right), \quad (17)$$

where H_u^l , H_s^l and H_i^l are the user hidden state, streamer hidden state, and item hidden state of the l -th layer. M_u^l , M_s^l and M_i^l denote the trainable weight matrices for user nodes, streamer nodes and item nodes, respectively.

3.4 Fraud Detection Model

Based on the embeddings generated on the Live-streaming Graph, we introduce an algorithm that employs a heterogeneous neural network for live-streaming fraud detection. The architecture of the proposed model LIFE is demonstrated in Figure 2. LIFE consists of three parts. First, a heterogeneous graph neural network that outputs user embedding z_u , streamer embedding z_s , item embedding z_i , edge embedding between user and streamer $z_{e_{us}}$, and edge embedding between streamer and item $z_{e_{si}}$. Second, a wide and deep architecture that learns embedding z_o from the contextual information of the current purchase transactions. Third, a fully-connected layer is built on top of the concatenation of all the embeddings, i.e.,

$$y = \text{classifier}(\text{concat}(z_o, z_i, z_s, z_u, z_{e_{us}}, z_{e_{si}})), \quad (18)$$

where the classifier is a fully connected layer with a sigmoid function that produces fraud prediction.

3.5 Label Propagation for Limited Labeled Training Samples

For training our detection model, we can directly label fraudulent purchase transactions in the Live-streaming Graph through multiple information channels, including filed complaints and user reports on live-streaming platforms. However, the number of labeled samples is too small to build an accurate model. Thus, we propagate the labels of the labeled seeds in the network via label propagation, recalling more suspicious samples. The label propagation is based on the observation that the fraudulent users often share the same or highly similar behavior patterns, such as frequently purchasing the same product at the same time. Therefore, we can discover more fraudulent users in the network from the suspicious seeds through label propagation.

Since a purchase transaction is from a user to an item, we can build a user-item bipartite graph (U - I graph) based on the transactions for our label propagation algorithms. Based on the U - I graph, a user-user graph (U - U graph) can then be generated by the U - I graph, where the weight between users can be calculated by the amount of the co-buy items. Finally, the label propagation algorithms can be applied in the U - U graph. However, this strategy is usually time-consuming and leads to extremely dense graph in

practice. This is because that a popular item on e-commerce platforms can be connected to millions of users. Thus, a U - U graph may contain many complete sub-graphs and the number of edges can be extremely large.

Inspired by the work [25], we solve this problem by the HGP approach, which propagates labels on (U - I) graph directly. In short, the calculation of U 's label can be divided into two steps. At the first step, for each item, we aggregate information from its neighboring users. We denote the temporary label vector for an item $i \in I$ as ρ_i^l . At the t -th round, it is updated by $\rho_i^l = \sum_{j \in N(i)} f_t^j$, where $N(i)$ is an indexed set of users that connects to i , and f_t^j is the label of users at the t -th step. At the second step, HGP propagates the labels back to users. The two steps iterate until either the convergence of labels or a given maximal number of iterations reached. Indeed, the label updating rule is equivalent to propagating information in the U - U graph using the Adamic-Adar weight [25].

Algorithm 1: Representation Learning on Heterogeneous Live-streaming Graphs.

Input: The Live-streaming Graph $G = (U, S, I, E_{us}, E_{si})$;
The node features X_u, X_s, X_i ;
The edge features $X_{e_{us}}, X_{e_{si}}$;
The number of layers L .

Output: The user embedding z_u ;
The streamer embedding z_s ;
The item embedding z_i ;
The edge embeddings $z_{e_{us}}$ and $z_{e_{si}}$.

// Initialization;

$H_u^0 \leftarrow X_u; H_s^0 \leftarrow X_s; H_i^0 \leftarrow X_i$;

$H_{e_{si}}^0 \leftarrow X_{e_{si}}; H_{e_{us}}^0 \leftarrow X_{e_{us}}$;

// Go through neural network;

for $l = 1, \dots, L$ **do**

for $e_{us} \in E_{us}$ **do**

$H_{e_{us}}^l = \sigma \left(W_{e_{us}}^l \cdot \text{AGG}_{e_{us}}^l (H_{e_{us}}^{l-1}, H_u^{l-1}, H_s^{l-1}) \right)$

end

for $e_{si} \in E_{si}$ **do**

$H_{e_{si}}^l = \sigma \left(W_{e_{si}}^l \cdot \text{AGG}_{e_{si}}^l (H_{e_{si}}^{l-1}, H_s^{l-1}, H_i^{l-1}) \right)$

end

for $u \in U$ **do**

$H_{UE}^{l-1} \leftarrow \text{Eq.}(7); H_{N(u)}^l \leftarrow \text{Eq.}(4); H_u^l \leftarrow \text{Eq.}(15)$

end

for $s \in S$ **do**

$H_{SE}^{l-1} \leftarrow \text{Eq.}(8); H_{N(s)}^l \leftarrow \text{Eq.}(5); H_s^l \leftarrow \text{Eq.}(16)$

end

for $i \in I$ **do**

$H_{IE}^{l-1} \leftarrow \text{Eq.}(9); H_{N(i)}^l \leftarrow \text{Eq.}(6); H_i^l \leftarrow \text{Eq.}(17)$

end

end

for $e_{us} \in E_{us}, e_{si} \in E_{si}, u \in U, s \in S, i \in I$ **do**

$z_{e_{us}} = H_{e_{us}}^L; z_{e_{si}} = H_{e_{si}}^L; z_u = H_u^L; z_i = H_i^L; z_s = H_s^L$

end

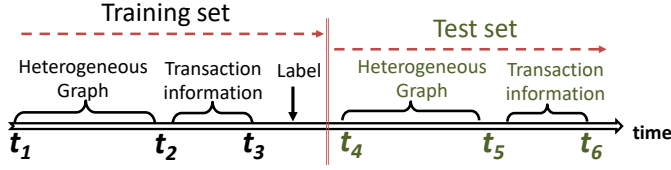


Figure 3: Training and test data collection from the Taobao live-streaming platform. We first collect the training set as a heterogeneous graph from time t_1 to t_2 and the purchase transactions occurring at time t_3 . Then, the model is validated on the test set comprised of a heterogeneous graph from t_4 to t_5 and purchase transactions at time t_6 .

4 EXPERIMENTAL EVALUATION

In this section, we conduct extensive experiments to evaluate the effectiveness of our proposed method on a large real-world dataset collected from Taobao Live platform. We also conduct a case study to demonstrate the performance of our method on detecting and tracing fraudulent communities on Taobao Live platform.

4.1 Dataset

The large real-world dataset collected from Taobao Live platform, which is presented as a live-streaming graph, consists of three types of nodes: users, streamers, and items. In addition, the nodes and edges are associated with attributes. For example, features of the users include user preference and registration information, features of streamers include registration and the times of live-streaming, and features of items include information such as product prices and the number of historical sells and purchases. Edge types between users and streamers include ‘like’, ‘follow’, and ‘buy’. Edge type between users and items is ‘buy’. Edge type between streamers and items is ‘promotion’. Each edge has its own features such as the number of purchases. The purchase transaction include order time, IP, price, payment methods, etc.

For the training set, we construct a live-streaming graph by collecting user behavior information within a one-week period (i.e., t_1 to t_2 in Figure 3), followed by the purchase transaction information. For the test set, another live-streaming graph is constructed based on the data within a one-week period as well (i.e., t_3 to t_4 in Figure 3), followed by the purchase transaction information at t_6 .

The training graph has a total of 50,721,800 edges published by 4,728,458 users and 118,384 streamers on 1,169,693 items, while the test graph has a total of 54,582,707 edges published by 4,862,550

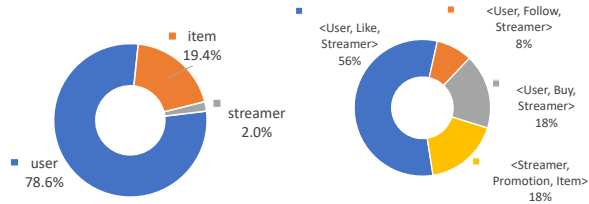


Figure 4: The proportion of different types of nodes and edges in the Taobao Live-streaming Graph.

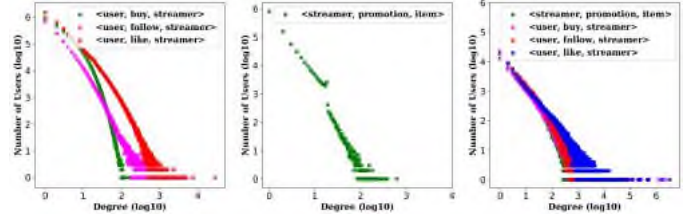


Figure 5: Degree distribution of different types of edges in the Taobao Live-streaming Graph. The edge type distribution follows a power law distribution.

Table 1: Results comparison of fraud detection in terms of AUC, Macro-F1, Micro-F1, recall at 90% precision (R90P) on the Taobao Live-streaming Graph

Method	AUC	F1	Precision	Recall	R90P
GCN	0.900	0.824	0.777	0.877	0.546
GAT	0.918	0.840	0.824	0.857	0.728
HAN	0.923	0.846	0.845	0.846	0.751
HetGNN	0.923	0.854	0.837	0.872	0.749
HGT	0.944	0.868	0.828	0.912	0.802
LIFE	0.966	0.909	0.922	0.896	0.896

users and 122,022 streamers on 1,257,292 items. Figure 4 shows the different numbers and distributions of node and edge types in the training graph. Most of the node types are users and the edge types are $\langle user, like, streamer \rangle$. Figure 5 also provides the analysis regarding the degree distribution of each edge type. Degree distribution significantly influences the performance of the heterogeneous graph embedding algorithm [30] due to the widely used node sampling process.

4.2 Baselines and Evaluation Metrics

4.2.1 Baselines. To demonstrate the effectiveness of the proposed method for fraud detection in live-streaming, we choose several baseline methods for fraud detection, including the state-of-the-art graph neural networks for both homogeneous and heterogeneous graphs. The details of the baseline methods are presented as below.

- Graph Convolutional Networks (GCN) [10]: A semi-supervised graph convolutional network, where features of nodes are aggregated from local neighbors. GCN was originally designed for homogeneous graphs;
- Graph Attention Networks (GAT) [24]: A semi-supervised neural network which incorporates attention into GCN. GAT was originally designed for homogeneous graphs;
- Heterogeneous Graph Attention Networks (HAN) [27]: A heterogeneous graph neural network based on hierarchical attention that aggregates features from meta-path based neighbors in a hierarchical manner;
- Heterogeneous Graph Neural Networks (HetGNN) [31]: A heterogeneous graph neural network model that aggregates feature information of the sampled neighboring nodes by different Bi-LSTMs;

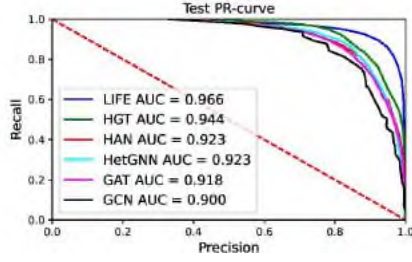


Figure 6: P-R chart of the methods. LIFE performs the best in terms of AUC.

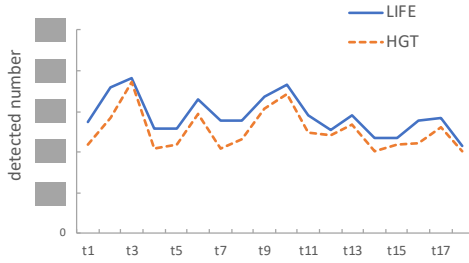


Figure 7: Online comparison between LIFE and HGT in terms of the number of detected fraud transactions at 90% precision during a time window of 17 days from t_1 to t_{17} . LIFE outperforms HGT consistently in terms of the amount of recalled fraudulent transactions.

Table 2: Ablation study to evaluate the effectiveness of core components in LIFE.

Method	AUC	F1	Precision	Recall	R90P
LIFE _e	0.874	0.797	0.767	0.829	0.530
LIFE _g	0.944	0.868	0.828	0.912	0.802
LIFE	0.966	0.909	0.922	0.896	0.896

- Heterogeneous Graph Transformer (HGT) [30]: A representation learning method for heterogeneous networks which applies the meta relations of heterogeneous graphs to parameterize weight matrices for the heterogeneous mutual attention, message passing, and propagation steps. HGT only considers the transaction information and the heterogeneous graph with node feature, node type and edge type;
- LIFE: Our method which includes the transaction information, the heterogeneous tripartite graph and edge features.

4.2.2 Evaluation Metrics. We evaluate LIFE and the baseline methods in terms of AUC, F1 score, precision, recall and recall at 90% precision (R90P). The metric of recall rate at 90% precision is chosen because that we need to ensure a high precision for our model to minimize disturbance to normal users. The precision over 90% is an essential pre-condition for a fraud detection model to be deployed in practice, so recall at 90% precision becomes an important criteria for comparing different models in the experiments.



Figure 8: The location distribution of fraudulent users detected by LIFE in China. The fraudulent users in Taobao Live-streaming are mainly distributed in the Yunnan and Guangdong Provinces.

4.3 Experimental Results

4.3.1 Fraud Detection on the Taobao Live-streaming Graph. We summarize the experimental results of the proposed model and baselines on the Taobao Live-streaming Graph in Table 1.

From the experimental results, we have the following observations: 1) The proposed LIFE method outperforms all the baselines in terms of AUC, F1 score, precision and R90P; 2) LIFE outperforms all baselines by 5% in terms of AUC, 7% in terms of F1 score, 12% in terms of precision, and 3% in terms of recall; 3) The recall at 90% precision is improved from 0.715 (i.e., the average R90P of baselines) to 0.896, which means 18.10% more frauds can be detected and removed from e-commerce platforms.

We further analyze the results by comparing each method more specifically and have the following observations: 1) The comparison of HGT and LIFE shows that the performance gain of LIFE significantly attributes to the introduction of the edge features; 2) When comparing GCN, GAT and LIFE, the performance improvement achieved by LIFE is prominent. This is because GCN and GAT are originally designed for homogeneous networks and is difficult to be extended to heterogeneous graphs with different types of nodes and edges, making GCN and GAT unsuitable for the live-streaming scenario in question; 3) When comparing HAN, HetGNN and LIFE, LIFE performs better because of the fully utilization of the properties of heterogeneous graphs such as the attributes obtained from multiple different types of nodes and edges. HAN and HetGNN use either node type or edge type to determine the weight matrices of GNN, the node or edge counts of different types, however, vary largely [7].

To evaluate the effectiveness of different core components in the Live-streaming Graph used by LIFE, we further conduct an ablation study by creating a few variants of LIFE, as shown in Table 2. LIFE_g is a variant of LIFE which only considers the purchase transaction information and the heterogeneous graph (i.e., node types, edge types, and node features), while LIFE_e is a variant of LIFE which only considers the edge features. We can observe that LIFE outperforms the other two variants because LIFE is able to learn the richer propagated information as well as the edge features in the Live-streaming Graph. Note that LIFE_g is equivalent to HGT so they feature the same performance.

Overall, the results show that LIFE outperforms all the baseline methods, which indicates that our model learns better embeddings from richer network information from the Taobao Live-Streaming Graph. Under the fixed threshold of 90% accuracy, our method is able to achieve a higher recall for fraud detection in the live-streaming scenario.

4.3.2 Fraud Propagation on the Taobao Live-streaming Graph. We first evaluate the effectiveness of the label propagation approach to expand the fraudulent users in the heterogeneous Taobao Live-streaming Graph to solve the issue of a limited number of labeled training data. We identified some black seeds (i.e., typical fraudulent users) through various channels including public opinions, user reports, complaints, black production cracking, and strategy mining on the Taobao Live platform. As a result, a total of 17,818 black accounts have been found. However, the number of determined black accounts is too small to train an accurate model, so we go back to label more black accounts from the original data. A straightforward approach is to randomly select samples from the original data and then manually check the fraudulent users in the sampled set. We repeated this process for five times but found that the average percentage of black users in the sample set is less than 2%, which makes this strategy rather ineffective to discover the fraudulent users. As discussed in Section 3.5, black accounts are related to each other through various forms of transaction behaviors and account association on the e-commerce platform, so in order to cover as many black and gray (i.e., suspected fraud) users as possible, we conduct fraud label propagation in the Taobao Live-streaming Graph to increase the number of labeled samples.

After fraud propagation, a total of 291,422 black and gray accounts were found. A further manual screening found that 144,693 accounts had fraudulent transactions or fake comments on the Taobao Live platform. Compared to the 2% accuracy achieved by manual labeling, the fraud propagation procedure in LIFE can improve the detection accuracy of fraud accounts to 49.65%, showing that it is a much efficient strategy to increase the number of labeled training samples.

4.3.3 Online Experiments on Taobao Live. We conduct online experiments on our Taobao Live platform. The experiments are designed to compare LIFE and HGT with respect to the number of fraudulent transactions detected by the models at 90% precision during a time period of 17 days. The detected transactions are then verified offline by a team of human experts, so the results are reliable.

Figure 7 shows the comparison results, where only the tendencies of the number of detected transactions are listed to avoid the sensitive information of the actual number of detected fraudulent transactions during the comparison time window. We can observe that the proposed LIFE model outperforms HGT consistently in terms of the amount of detected fraudulent transactions, which further indicates the effectiveness of introducing rich information learned by LIFE in the live-streaming scenario.

4.4 Case Study

In this part, we conduct a case study to demonstrate the practicality of LIFE in fraudulent community detection in live-streaming applications. Specifically, based on the detected fake transactions of

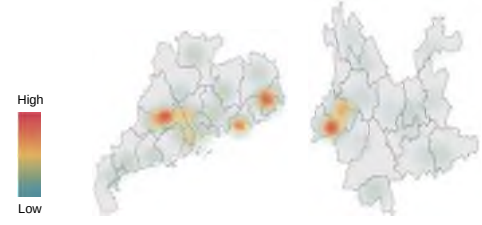


Figure 9: The heat map of fraudulent users in Guangdong (left) and Yunnan (right) provinces. In Yunnan, the fraudulent users are concentrated in one area, while the fraudulent users are scattered in multiple regions in Guangdong.



Figure 10: A fraudulent community detected by our model. This community contains 17 users, 56 streamers, and 828 edges between users and streamers. In particular, we found that 13 users share the same IP address, and there is a device cluster to simulate virtual users behind the fraudulent community.

fraudulent users, we undertake further fraud tracing to identify the exact locations of high-density fraudulent communities on Taobao live-streaming platform. This goal is achievable thanks to the availability of various spatial and temporal information associated with the transactions and the IP addresses of users, through which it is possible for precise positioning of the fraudulent communities.

To detect fraudulent communities, rather than individual fraudulent users, we conducted data analysis and mining of the location information of the detected fraudulent users and cluster them based on their location information. After analyzing the grouped communities by their locations, we found that some fraudulent communities on Taobao Live platform exhibit distinctive characteristics in their geographic location distributions, which are considered as high-risk communities. To further validate those high-risk communities, we carried out further investigations through physical visits and surveys to determine whether those communities are indeed fraudulent communities.

For the detected high-risk communities, we visualize their physical locations in Figure 8. We can observe that the fraudulent users mainly concentrate in Guangdong and Yunnan Provinces of China. Furthermore, through more accurate positioning using IP addresses of users, we found that the fraudulent users in Yunnan Province are mostly concentrated in one area, while those in Guangdong Province are scattered in multiple regions, as shown in Figure 9.

Figure 10 shows a real case of what happens behind the scene for the detected high-risk fraudulent communities through physical visits and surveys. Some fraudulent groups are using multiple devices to simulate a large number of users to create false traffic, which will further attract more users to the live-streamer.

This case study clearly demonstrates that our model is capable of providing solid clues and evidences in helping actually locate physical fraudulent communities that are responsible for fraudulent transactions on live-streaming platforms. This is believed to effectively help clean up the contaminated live-streaming environment and offer more healthy and fair live-streaming platforms to users.

5 CONCLUSIONS

In this paper, we propose an innovative heterogeneous graph neural network (HGNN)-based anti-fraud model, called LIFE, which incorporates heterogeneous nodes and edges to detect fraudulent transactions as well as communities on live-streaming e-commerce platforms. A live-streaming heterogeneous graph is constructed to model the nature of tripartite heterogeneity of live-streaming e-commerce platforms. Embedding of various entities in the graph are performed to support representation learning for HGNN in fraud detection. In addition, a label propagation algorithm is proposed to address the problem of a limited number of labeled training fraudulent users. We conduct extensive experiments on a large-scale Taobao Live dataset and the results demonstrate that LIFE outperforms mainstream baselines in a variety of performance metrics. A case study is also carried out to further validate the practicality of LIFE in detecting and physically tracing fraudulent communities on live-streaming e-commerce platforms.

ACKNOWLEDGMENTS

This work was supported by Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Zhejiang Lab (111007-PI2001), Zhejiang Provincial Natural Science Foundation (LZ21F030001).

REFERENCES

- [1] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 477–488.
- [2] Sahil Dhankhad, Emad Mohammed, and Behrouz Far. 2018. Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 122–125.
- [3] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*. PMLR, 1263–1272.
- [4] Qingyu Guo, Zhao Li, Bo An, Pengrui Hui, Jiaming Huang, Long Zhang, and Mengchen Zhao. 2019. Securing the deep fraud detector in large-scale e-commerce platform via adversarial machine learning approach. In *The World Wide Web Conference*. 616–626.
- [5] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).
- [6] Graeme Horsman. 2018. A forensic examination of the technical and legal challenges surrounding the investigation of child abuse on live streaming platforms: A case study on Periscope. *Journal of information security and applications* 42 (2018), 107–117.
- [7] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.
- [8] Junxian Huang, Yinglian Xie, Fang Yu, Qifa Ke, Martin Abadi, Eliot Gillum, and Z Morley Mao. 2013. Socialwatch: detection of online service abuse via large-scale social graphs. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. 143–148.
- [9] Kai Kang, Jinxuan Lu, Lingyun Guo, and Wenlu Li. 2021. The dynamic effect of interactivity on customer engagement behavior through tie strength: Evidence from live streaming commerce platforms. *International Journal of Information Management* 56 (2021), 102251.
- [10] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [11] Zhao Li, Zhanlin Liu, Jiaming Huang, Geyu Tang, Yucong Duan, Zhiqiang Zhang, and Yifan Yang. 2019. MV-GCN: Multi-View Graph Convolutional Networks for Link Prediction. *IEEE Access* 7 (2019), 176317–176328.
- [12] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2077–2085.
- [13] Zhicong Lu, Haijun Xia, Seongkook Heo, and Daniel Wigdor. 2018. You watch, you give, and you engage: a study of live streaming practices in China. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.
- [14] Jun Ma, Danqing Zhang, Yun Wang, Yan Zhang, and Alexey Pozdnoukhov. 2018. GraphRAD: a graph-based risky account detection system. In *Proceedings of ACM SIGKDD conference, London, UK*. 9.
- [15] Renxin Mao, Zhao Li, and Jinhua Fei. 2015. Fraud transaction recognition: A money flow network approach. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1871–1874.
- [16] Richard Oentaryo, Ee-Peng Lim, Michael Finegold, David Lo, Feida Zhu, Clifton Phua, Eng-Yeow Cheu, Ghim-Eng Yap, Kelvin Sim, Minh Nhut Nguyen, et al. 2014. Detecting click fraud in online advertising: a data mining approach. *The Journal of Machine Learning Research* 15, 1 (2014), 99–140.
- [17] Susie Xi Rao, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Zhiyao Chen, Yinan Shan, Yang Zhao, and Ce Zhang. 2020. xFraud: Explainable Fraud Transaction Detection on Heterogeneous Graphs. *arXiv preprint arXiv:2011.12193* (2020).
- [18] Aleksandr Shchur, Aleksandar Bojchevski, Mohamed Farghal, Stephan Günnemann, and Yusuf Saber. 2018. Anomaly Detection in Car-Booking Graphs. In *2018 IEEE International Conference on Data Mining Workshops*. 604–607.
- [19] Ning Su, Yiqun Liu, Zhao Li, Yuli Liu, Min Zhang, and Shaoping Ma. 2018. Detecting Crowdturfing: Add to Favorites' Activities in Online Shopping. In *Proceedings of the 2018 World Wide Web Conference*. 1673–1682.
- [20] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.
- [21] Yizhou Sun, Brandon Norrick, Jiawei Han, Xifeng Yan, Philip S Yu, and Xiao Yu. 2012. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1348–1356.
- [22] Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. 2013. Unik: Unsupervised social network spam detection. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 479–488.
- [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [24] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [25] Haobo Wang, Zhao Li, Jiaming Huang, Pengrui Hui, Weiwei Liu, Tianlei Hu, and Gang Chen. 2020. Collaboration based multi-label propagation for fraud detection. In *IJCAI*.
- [26] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xion. 2019. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *Proceedings of The 2019 World Wide Web Conference*. 310–316.
- [27] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.
- [28] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591* (2019).
- [29] Rui Wen, Jianyu Wang, Chunming Wu, and Jian Xiong. 2020. Asa: Adversary situation awareness via heterogeneous graph convolutional networks. In *Companion Proceedings of the Web Conference 2020*. 674–678.
- [30] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2020. Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [31] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.
- [32] Mengchen Zhao, Zhao Li, Bo An, Haifeng Lu, Yifan Yang, and Chen Chu. 2018. Impression Allocation for Combating Fraud in E-commerce Via Deep Reinforcement Learning with Action Norm Penalty. In *IJCAI*. 3940–3946.

Embedding-based Product Retrieval in Taobao Search

Sen Li, Fuyu Lv*

Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng
Alibaba Group
Hangzhou, China
{lisen.lisen,fuyu.lfy,taiwei.jtw,
guli.lingl,shaoyao,yuanhan}@alibaba-inc.com

Xiao-Ming Wu¹, Qianli Ma²

¹The Hong Kong Polytechnic University
Hong Kong, China

²South China University of Technology
Guangzhou, China

csxmwu@comp.polyu.edu.hk, qianlima@scut.edu.cn

ABSTRACT

Nowadays, the product search service of e-commerce platforms has become a vital shopping channel in people's life. The retrieval phase of products determines the search system's quality and gradually attracts researchers' attention. Retrieving the most relevant products from a large-scale corpus while preserving personalized user characteristics remains an open question. Recent approaches in this domain have mainly focused on embedding-based retrieval (EBR) systems. However, after a long period of practice on Taobao, we find that the performance of the EBR system is dramatically degraded due to its: (1) low relevance with a given query and (2) discrepancy between the training and inference phases. Therefore, we propose a novel and practical embedding-based product retrieval model, named Multi-Grained Deep Semantic Product Retrieval (MGDSPR). Specifically, we first identify the inconsistency between the training and inference stages, and then use the softmax cross-entropy loss as the training objective, which achieves better performance and faster convergence. Two efficient methods are further proposed to improve retrieval relevance, including smoothing noisy training data and generating relevance-improving hard negative samples without requiring extra knowledge and training procedures. We evaluate MGDSPR on Taobao Product Search with significant metrics gains observed in offline experiments and online A/B tests. MGDSPR has been successfully deployed to the existing multi-channel retrieval system in Taobao Search. We also introduce the online deployment scheme and share practical lessons of our retrieval system to contribute to the community.

CCS CONCEPTS

• Information systems → Retrieval models and ranking.

KEYWORDS

Embedding-based retrieval system; E-commerce search

ACM Reference Format:

Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, and Xiao-Ming Wu¹, Qianli Ma². 2021. Embedding-based Product Retrieval in Taobao

*Equal contribution

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8332-5/21/08...\$15.00

<https://doi.org/10.1145/3447548.3467101>

Search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, August 14–18, 2021, Virtual Event, Singapore. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3447548.3467101>

1 INTRODUCTION

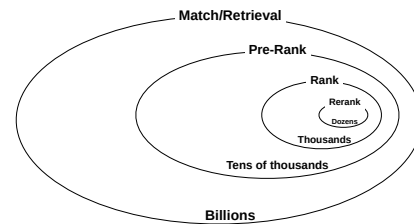


Figure 1: Overview of the product search system in Taobao. The head of each circle denotes different phase. The bottom is the scale of the corresponding candidate set.

Nowadays, online shopping has become a daily habit in people's lives. The top E-commerce shopping platforms (such as eBay, Amazon, Taobao, and JD) have hundreds of millions of daily active users and thus facilitate billion-level transaction records [17, 26, 34]. Therefore, product search engines are designed to discover products that satisfy users, which is also our working goal. As shown in Figure 1, our search system uses the “match-prerank-rank-rerank” architecture to screen and sort thousands of products from billions of candidates to possess controllable and high-efficiency characteristics. We finally return dozens of products to display to users. Obviously, the match (retrieval) phase plays an important role in determining the quality of the item candidate set fed to the follow-up ranking stage. The problem gradually receives more and more attention from academia and industry.

Search retrieval in e-commerce poses different challenges than in web (document) search: the text in e-commerce is usually shorter and lacks grammatical structure, while it is important to consider the massive historical user behaviors [2, 3]. The lexical matching engine (typically an inverted index [21, 24, 38]), despite its widely criticized semantic gap issue [14, 24, 31], remains a vital part of current retrieval systems due to its reliability and controllability of search relevance (exactly matching query terms). However, it hardly distinguishes users' interests in the same query and cannot flexibly capture user-specific characteristics. Hence, how to effectively retrieve the most relevant products satisfying users while considering the relationship between query semantics and historical user behaviors is the main challenge facing e-commerce platforms.

With the development of deep learning [35], Amazon [21] and JD [34] built their respective two-tower embedding-based retrieval (EBR) systems to provide relevant or personalized product candidates in their e-commerce search engines. Both of them reported the success of EBR without further discussing its low controllability of search relevance (compared to the lexical matching engine). We have also built an EBR system that can dynamically capture the relationship between query semantics and personalized user behaviors, and launched it on Taobao¹ Product Search for quite a long time. In the first deployment, it can achieve a good improvement in various metrics. However, after long observation, we have found that the embedding-based method's controllability of relevance is relatively low due to the inability to exactly matching query terms [11], resulting in increasing user complaints and bad cases that cannot be fixed. To improve its controllability of relevance (*i.e.*, resolving bad cases), we have adopted a relevance control module to filter the retrieved products. The control module only keeps those products that meet the relevance standards of exact matching signals and feed them to the follow-up ranking module. However, we statistically find it usually filters out thirty percent of candidates due to the low relevance of retrieved products. It is quite a waste of online computing resources because the filtered products cannot participate in the ranking stage, thus degrading the EBR system's performance. Therefore, the practical challenge for our search system is to enable the embedding-based model to retrieve more relevant products and increase the number of participants in the subsequent ranking stage.

Moreover, random negative samples are widely used to train large-scale deep retrieval models to ensure the sample space in training is consistent with that of the inference phase [14, 34]. Nevertheless, there remains a discrepancy in existing e-commerce product search methods [21, 34] due to the inconsistent behavior between the training and inference stages. Specifically, during inference, the model needs to select the top- K products closest to the current query from all candidates, requiring the ability for global comparison. However, [21] and [34] both adopt hinge (pairwise) loss as the training objective, which can only do local comparison.

This paper introduces the design of the proposed Multi-Grained Deep Semantic Product Retrieval (MGDSPR) model, its effect on each stage of the search system, and the lessons learned from applying it to product search. To tackle the above problems, we first use the softmax cross-entropy loss as the training objective to equip the model with global comparison ability, making training and inference more consistent. We further propose two effective methods without extra training procedures to enable MGDSPR to retrieve more relevant products. Specifically, we smooth the relevance noise introduced by using user implicit feedback (*i.e.*, click data) logs as training data [29, 31] by including a temperature parameter to the softmax function. Also, we mix the positive and random negative samples to generate relevance-improving hard negative samples. Moreover, we adapt the relevance control module to enhance the EBR system's controllability of search relevance. The effectiveness of MGDSPR is verified by an industrial dataset collected from the Taobao search system and online A/B tests.

The main contributions of this work are summarized as follows:

- We propose a Multi-Grained Deep Semantic Product Retrieval (MGDSPR) model to dynamically capture the relationship between user query semantics and his/her personalized behaviors and share its online deployment solution.
- We identify the discrepancy between training and inference in existing e-commerce retrieval systems and suggest using the softmax cross-entropy loss as the training objective to achieve better performance and faster convergence.
- We propose two methods to make the embedding-based model retrieve more relevant products without additional knowledge and training time. We further adapt the relevance control module to improve the EBR system's controllability of relevance.
- Experiments conducted on a large-scale industrial dataset and online Product Search of Taobao demonstrate the effectiveness of MGDSPR. Moreover, we analyze the effect of MGDSPR on each stage of the search system.

2 RELATED WORK

2.1 Deep Matching in Search

With the booming interest in deep NLP techniques, various neural models have been proposed to address the semantic gap problem raised by traditional lexical matching in the last few years. Those approaches fall into two categories: representation-based learning and interaction-based learning. The two-tower structure is the typical characteristic of representation-based models, such as DSSM [15], CLSM [25], LSTM-RNN [22], and ARC-I [13]. Each tower uses a siamese/distinct neural network to generate semantic representations of query/document. Then a simple matching function (*e.g.*, inner product) is applied to measure the similarity between the query and document. Interaction-based methods learn the complicated text/relevance patterns between the query and document. Popular models include MatchPyramid [23], Match-SRNN [28], DRMM [11], and K-NRM [32]. Other than semantic and relevance matching, more complex factors/trade-offs, *e.g.*, user personalization [2, 3, 10] and retrieval efficiency [5], need to be considered when applying deep models to a large-scale online retrieval system.

2.2 Deep Retrieval in Industry Search

Representation-based models with an ANN (approximate near neighbor) algorithm have become the mainstream trend to efficiently deploy neural retrieval models in industry. For social networks, Facebook developed an EBR system to take text matching and searcher's context into consideration [14]. They introduced various tricks and experiences (*e.g.*, hard mining, ensemble embedding, and inverted index-based ANN) to achieve hybrid retrieval (fuzzy matching). For display advertising, Baidu proposed MOBIUS [8] for CPM (cost per mile) maximization in the web ads retrieval phase, reducing the objective distinction between ranking and matching. For web search, Google [30] adopted transfer learning to learn semantic embeddings from data in recommendation systems to alleviate the cold start problem. Due to more text features and fewer user behaviors, their search scenarios are characterized by strong semantic matching and weak personalization. For e-commerce search, Amazon developed a two-tower model to address the semantic gap issue in a lexical matching engine for semantic product retrieval [21],

¹<https://www.taobao.com/>

where one side uses n-gram query features and the other side exploits item features, without considering user personalization. Recently, JD [34] proposed a deep personalized and semantic retrieval model (DPSR) to combine text semantics and user behaviors. However, DPSR aggregates user behaviors through average pooling, weakening personalization characteristics. Furthermore, neither Amazon nor JD studies the problem of insufficient product relevance caused by the EBR method. This paper will discuss the low relevance issue of the EBR system encountered in Taobao Product Search and propose our solution.

3 MODEL

Here, we introduce our model called Multi-Grained Deep Semantic Product Retrieval (MGDSPR) to simultaneously model query semantics and historical behavior data, aiming at retrieving more products with good relevance. The general structure of MGDSPR is illustrated in Figure 2. We first define the problem and then introduce our design of the two-tower model, including the user tower and the item (product) tower. Finally, we elaborate on the training objective and proposed methods to retrieve more relevant products.

3.1 Problem Formulation

We first formulate the e-commerce product retrieval problem and our solution as well as the notations used in this paper. Let $\mathcal{U} = \{u_1, \dots, u_u, \dots, u_N\}$ denote a collection of N users, $\mathcal{Q} = \{q_1, \dots, q_u, \dots, q_N\}$ denote the corresponding queries, and $\mathcal{I} = \{i_1, \dots, i_i, \dots, i_M\}$ denote a collection of M items (products). Also, we divide the user u 's historical behaviors into three subsets according to the time interval from the current time t : real-time (denoted as $\mathcal{R}^u = \{i_1^u, \dots, i_t^u, \dots, i_T^u\}$, before the current time step), short-term (denoted as $\mathcal{S}^u = \{i_1^u, \dots, i_t^u, \dots, i_T^u\}$, before $\mathcal{R}$ and within ten days) and long-term sequences (denoted as $\mathcal{L}^u = \{i_1^u, \dots, i_t^u, \dots, i_T^u\}$, before $\mathcal{S}$ and within one month), where T is the length of the sequence.

We now define the task. Given the historical behaviors ($\mathcal{R}^u, \mathcal{S}^u, \mathcal{L}^u$) of a user $u \in \mathcal{U}$, after he/she submits query q_u at time t , we would like to return a set of items $i \in \mathcal{I}$ that satisfy his/her search request. Typically, we predict top- K item candidates from $\mathcal{I}$ at time t based on the scores z between the user (query, behaviors) and items, i.e.,

$$z = \mathcal{F}(\phi(q_u, \mathcal{R}^u, \mathcal{S}^u, \mathcal{L}^u), \psi(i)), \quad (1)$$

where $\mathcal{F}(\cdot)$, $\phi(\cdot)$, $\psi(\cdot)$ denote the scoring function, query and behaviors encoder, and item encoder, respectively. Here, we adapt the two-tower retrieval model for efficiency. We instantiate $\mathcal{F}$ with the inner product function. In the following, we introduce the design of the user and item towers, respectively.

3.2 User Tower

3.2.1 Multi-Granular Semantic Unit. Queries in Taobao search are usually in Chinese. After query segmentation, the average length of the segmentation result is less than three. As such, we propose a multi-granular semantic unit to discover the meaning of queries from multiple semantic granularities and enhance the representation of queries. Given a query's segmentation result $q_u = \{w_1^u, \dots, w_n^u\}$ (e.g., {红色, 连衣裙}), each $w^u = \{c_1^u, \dots, c_m^u\}$ (e.g., {红, 色}), and its historical query $q_{his} = \{q_1^u, \dots, q_k^u\} \in \mathbb{R}^{k \times d}$ (e.g., {绿色, 半身裙, 黄色, 长裙}), where $w_n \in \mathbb{R}^{1 \times d}$, $c_m \in \mathbb{R}^{1 \times d}$

and $q_k \in \mathbb{R}^{1 \times d}$, we can obtain its six granular representations $Q_{mgs} \in \mathbb{R}^{6 \times d}$. It is done by concatenating q_u 's unigram mean-pooling $q_{1\_gram} \in \mathbb{R}^{1 \times d}$, 2-gram mean-pooling $q_{2\_gram} \in \mathbb{R}^{1 \times d}$, word segmentation mean-pooling $q_{seg} \in \mathbb{R}^{1 \times d}$, word segmentation sequence $q_{seg\_seq} \in \mathbb{R}^{1 \times d}$, historical query words $q_{his\_seq} \in \mathbb{R}^{1 \times d}$, and mixed $q_{mix} \in \mathbb{R}^{1 \times d}$ representations. d , n , and m denote the embedding size, the number of word segmentation, and the number of words in each segment, respectively. Formally, the Multi-Granular Semantic representation Q_{mgs} is obtained as follows:

$$q_{1\_gram} = \text{mean\_pooling}(c_1, \dots, c_m), \quad (2)$$

$$q_{2\_gram} = \text{mean\_pooling}(c_1 c_2, \dots, c_{m-1} c_m), \quad (3)$$

$$q_{seg} = \text{mean\_pooling}(w_1, \dots, w_n), \quad (4)$$

$$q_{seg\_seq} = \text{mean\_pooling}(\text{Trm}(w_1, \dots, w_n)), \quad (5)$$

$$q_{his\_seq} = \text{softmax}(q_{seg} \cdot (q_{his})^T) q_{his}, \quad (6)$$

$$q_{mix} = q_{1\_gram} + q_{2\_gram} + q_{seg} + q_{seg\_seq} + q_{his\_seq}, \quad (7)$$

$$Q_{mgs} = \text{concat}(q_{1\_gram}, q_{2\_gram}, q_{seg}, q_{seg\_seq}, q_{his\_seq}, q_{mix}), \quad (8)$$

where Trm , mean_pooling , and concat denote the Transformer [27], average, and vertical concatenation operation, respectively. We average all the outputs of the last layer of the Transformer in Eq. (5).

3.2.2 User Behaviors Attention. User behaviors are recorded by their history of items clicked (or bought). Taking user u 's short-term behaviors $\mathcal{S}^u$ as an example, $i_t^u \in \mathcal{S}^u$ denotes the user clicks on item i at time t , and each item i is described by its ID and side information $\mathcal{F}$ (e.g., leaf category, first-level category, brand and shop) [18]. Specifically, each input item $i_t^u \in \mathcal{S}^u$ is defined by:

$$e_i^f = W_f \cdot x_i^f, \quad (9)$$

$$i_t^u = \text{concat}(\{e_i^f | f \in \mathcal{F}\}), \quad (10)$$

where W_f is the embedding matrix and x_i^f is a one-hot vector. $e_i^f \in \mathbb{R}^{1 \times d_f}$ is the corresponding embedding vector of size d_f and $\cdot$ denotes matrix multiplication. We concatenate the embeddings of item i 's ID and side information in Eq. (10). We use the same way to embed items in real-time $\mathcal{R}^u$ and long-term $\mathcal{L}^u$ sequences.

Unlike the target-item attention [9, 36, 37] used in advertising and recommendation, here we use query attention to capture user history behaviors related to the current query semantics. Moreover, inspired by [2], we put an all-zero vector into user behavior data to remove potential noise and deal with situations where the historical behaviors may not be related to the current query. In the following, we introduce the fusion of real-time, short-term, and long-term sequences, respectively.

For real-time sequences $\mathcal{R}^u = \{i_1^u, \dots, i_t^u, \dots, i_T^u\}$, we apply Long Short-Term Memory (LSTM) [12] to capture the evolution and collect all hidden states $\mathcal{R}_{lstm}^u = \{h_1^u, \dots, h_t^u, \dots, h_T^u\}$. Next, we use multi-head self-attention to aggregate multiple potential points of interest [18] in $\mathcal{R}_{lstm}^u$ to get $\mathcal{R}_{self\_att}^u = \{h_1^u, \dots, h_t^u, \dots, h_T^u\}$. Then, we add a zero vector at the first position of $\mathcal{R}_{self\_att}^u$, resulting in $\mathcal{R}_{zero\_att}^u = \{0, h_1^u, \dots, h_t^u, \dots, h_T^u\} \in \mathbb{R}^{(T+1) \times d}$. Finally, the real-time personalized representation $H_{real} \in \mathbb{R}^{6 \times d}$ related to the current query is obtained by the attention operation (Q_{mgs} is analogous to

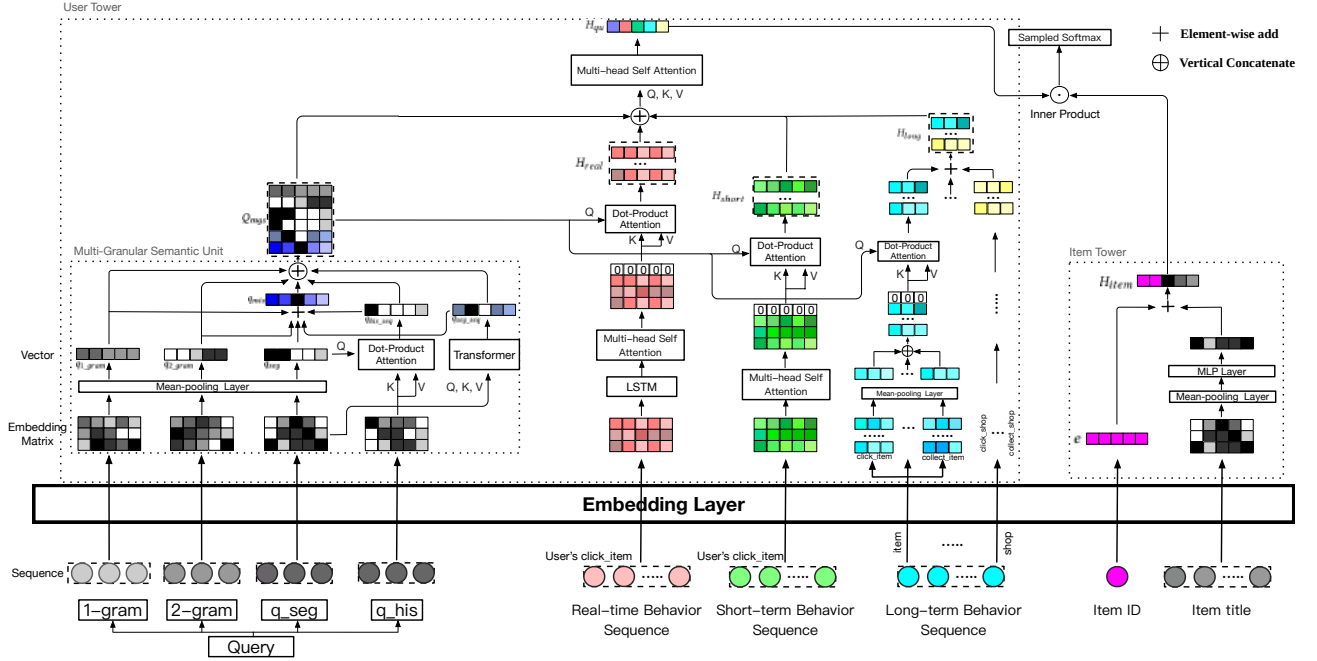


Figure 2: General architecture of the proposed Multi-Grained Deep Semantic Product Retrieval model (MGDSPR).

Q in the attention mechanism), which is defined by:

$$H_{real} = \text{softmax}(Q_{mgs} \cdot R_{zero\_att}^T) \cdot R_{zero\_att}^T \quad (11)$$

For short-term sequences $S^u = \{i_1^u, \dots, i_t^u, \dots, i_T^u\}$, we apply multi-head self-attention to aggregate S^u into $S_{self\_att}^u = \{h_1^u, \dots, h_t^u, \dots, h_T^u\}$. We add a zero vector at the first position of $S_{self\_att}^u$, resulting in $S_{zero\_att}^u = \{0, h_1^u, \dots, h_t^u, \dots, h_T^u\} \in \mathbb{R}^{(T+1) \times d}$. Finally, the short-term personalized representation $H_{short} \in \mathbb{R}^{6 \times d}$ is defined by:

$$H_{short} = \text{softmax}(Q_{mgs} \cdot S_{zero\_att}^T) \cdot S_{zero\_att}^T \quad (12)$$

The real-time and short-term sequences are composed of click sequences.

We use four attribute behaviors to describe the long-term sequence (within one month), including *item* ($\mathcal{L}_{item}^u$), *shop* ($\mathcal{L}_{shop}^u$), *leaf category* ($\mathcal{L}_{leaf}^u$) and *brand* ($\mathcal{L}_{brand}^u$). Each attribute behavior is described by a user's click, buy and collecting actions. For example, $\mathcal{L}_{item}^u$ consists of multiple action sequences: $\mathcal{L}_{click\_item}$, $\mathcal{L}_{buy\_item}$ and $\mathcal{L}_{collect\_item}$. Entries in each action sequence are embedded by Eq. (9) and aggregated into a vector through mean-pooling with consideration of quick response in online environment, resulting in $\mathcal{L}_{item}^u = \{0, h_{click}, h_{buy}, h_{collect}\}$. The representation of item attribute behavior $H_{a\_item} \in \mathbb{R}^{6 \times d}$ is then defined by:

$$H_{a\_item} = \text{softmax}(Q_{mgs} \cdot L_{item}^T) \cdot L_{item}^T \quad (13)$$

Finally, the long-term personalized representation $H_{long} \in \mathbb{R}^{6 \times d}$ is defined as follows:

$$H_{long} = H_{a\_item} + H_{a\_shop} + H_{a\_leaf} + H_{a\_brand} \quad (14)$$

where H_{a_shop} , H_{a_leaf} , and H_{a_brand} denote the representation of the attribute behaviors of *shop*, *leaf category*, and *brand* respectively.

3.2.3 Fusion of Semantics and Personalization. To retrieve products relevant to the current user's query and preserve personalized characteristics, we take the multi-granular semantic representation Q_{mgs} and personalized representations ($H_{real}, H_{short}, H_{long}$) as the input of self-attention to dynamically capture the relationship between the two. Specifically, we add a "[CLS]" token at the first position of the input $I = \{[CLS], Q_{mgs}, H_{real}, H_{short}, H_{long}\}$ of self-attention and regard the output as the user tower's representation $H_{qu} \in \mathbb{R}^{1 \times d}$, which is defined as follows:

$$H_{qu} = \text{Self\_Att}^{first}([CLS], Q_{mgs}, H_{real}, H_{short}, H_{long}). \quad (15)$$

3.3 Item Tower

For the item tower, we experimentally use item ID and title to obtain the item representation H_{item} . Given the representation of item i 's ID, $e_i \in \mathbb{R}^{1 \times d}$, and its title segmentation result $T_i = \{w_1^i, \dots, w_N^i\}$, $H_{item} \in \mathbb{R}^{1 \times d}$ is calculated as follows:

$$H_{item} = e + \tanh(W_t \cdot \frac{\sum_{i=1}^N w_i}{N}), \quad (16)$$

where W_t is the transformation matrix. We empirically find that applying LSTM [12] or Transformer [27] to capture the context of the title is not as effective as simple mean-pooling since the title is stacked by keywords and lacks grammatical structure.

3.4 Loss Function

To make the sample space where the model is trained consistent with that of online inference, Huang et al. [14], Nigam et al. [21], and Zhang et al. [34] use random samples as negative samples. However, they use pairwise (hinge) loss as the training objective, making training and testing behavior inconsistent. Specifically,

during inference, the model needs to pick the top- K items that are closest to the current query from all candidates, which requires the model to have the ability of global comparison. However, hinge loss can only do local comparison. Also, hinge loss introduces a cumbersome tuning margin, which has a significant impact on performance [14]. Here, we adapt the softmax cross-entropy loss as the training objective, achieving faster convergence and better performance without additional hyper-parameter tuning.

Given a user u and his/her query q_u , the positive item i^+ is the item clicked by u under q_u . The training objective is defined by:

$$\hat{y}_{(i^+|q_u)} = \frac{\exp(\mathcal{F}(q_u, i^+))}{\sum_{i' \in I} \exp(\mathcal{F}(q_u, i'))}, \quad (17)$$

$$L(\nabla) = - \sum_{i \in I} y_i \log(\hat{y}_i), \quad (18)$$

where $\mathcal{F}$, I , i^+ , and q_u denote the inner product, the full item pool, the item tower's representation H_{item} , and the user tower's representation H_{qu} , respectively. Note that Eq. (17) endows the model with global comparison ability. The softmax involves calculating an expensive partition function, which scales linearly to the number of items. In practice, we use sampled softmax (an unbiased approximation of full-softmax) [4, 16] for training. Similar to [34], we also experimentally find that using the same set of random negative samples for every training example in the current batch results in similar performance as using a different set for each one. We adopt the former training method to reduce computing resources.

To improve the EBR system's relevance in retrieval and increase the number of products participating in the follow-up ranking stage while maintaining high efficiency, we propose two efficient methods without relying on additional knowledge to make our model retrieve more relevant products.

3.4.1 Smoothing Noisy Training Data. In e-commerce search, users' click and purchase records are used as supervisory signals to train a model. However, these signals are noisy since they are influenced not only by query-product relevance but also by images, prices, and user preferences [29, 31]. Hence, we introduce a temperature parameter τ into softmax to smooth the overall fitted distribution of the training data. If $\tau \rightarrow 0$, the fitted distribution is close to one-hot distribution, which means that the model completely fits the supervisory signals. The model will be trained to push positive items far away from negative ones, even if the relevance of a positive item is low. If $\tau \rightarrow \infty$, the fitted distribution is close to a uniform distribution, indicating that the model does not fit the supervisory signals at all. We can increase τ to reduce the noise in training data and thus alleviate the impact of low relevance caused by fully fitting users' click records, which does not require additional knowledge and is verified by our experiments. Formally, the softmax function with the temperature parameter τ is defined as follows:

$$\hat{y}_{(i^+|q_u)} = \frac{\exp(\mathcal{F}(q_u, i^+)/\tau)}{\sum_{i' \in I} \exp(\mathcal{F}(q_u, i')/\tau)}. \quad (19)$$

3.4.2 Generating Relevance-improving Hard Negative Samples. Unlike prior works [20] that require additional annotated training data and training process, we propose a method to generate relevance-improving hard negative samples in the embedding space. Specifically, given a training example (q_u, i^+, i^-) , where i^- denotes a set of

random negative samples sampled from item pool I . For simplicity, we use q_u , i^+ , and i^- to refer to their respective representations. We first select the negative items of i^- that have the top- N inner product scores with q_u to form the hard sample set I_{hard} , and then mix $i^+ \in \mathbb{R}^{1 \times d}$ and $I_{hard} \in \mathbb{R}^{N \times d}$ by interpolation to obtain the generated sample set $I_{mix} \in \mathbb{R}^{N \times d}$, which is defined as follows:

$$I_{mix} = \alpha i^+ + (1 - \alpha) I_{hard}, \quad (20)$$

where $\alpha \in \mathbb{R}^{N \times 1}$ is sampled from the uniform distribution $U(a, b)$ ($0 \leq a < b \leq 1$). The closer α is to 1, the closer the generated sample is to the positive samples i^+ in the embedding space, indicating the harder the generated sample is. We take I_{mix} as the set of relevance-improving hard negative samples and include it in the denominator of the softmax function to make the model distinguish the positive sample i^+ and its nearby samples. Formally, the softmax function with relevance-improving hard samples I_{mix} is defined as follows:

$$\hat{y}_{(i^+|q_u)} = \frac{\exp(\mathcal{F}(q_u, i^+)/\tau)}{\sum_{i' \in I \cup I_{mix}} \exp(\mathcal{F}(q_u, i')/\tau)}. \quad (21)$$

Note that we can tune the maximum b and minimum a of the uniform distribution U to determine the "hardness" of the generated relevance-improving negative samples. This generation process only needs a linear interpolation after calculating the inner product scores between the current query q_u and the negative samples of i^- , which is quite efficient.

4 SYSTEM ARCHITECTURE

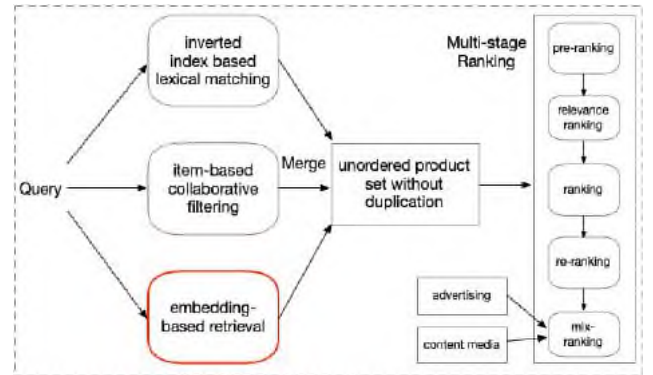


Figure 3: Overview of Taobao search engine.

As illustrated in Figure 3, at a high level, the Taobao search engine works as follows: a user issues a query, which triggers a multi-channel retrieval system, producing an unordered candidate set without duplication. Before the most relevant items are finally displayed to users, the candidates are passed through multi-stages of ranking, including pre-ranking, relevance ranking (removing products that are inconsistent with the predictions of the query's category), ranking, re-ranking, and mix-ranking. Our embedding-based retrieval module is the third matching channel as a supplement to the existing two-channel retrieval. In the following, we introduce how to deploy MGDSPR in the production environment once we have trained the model and the relevance control module on the EBR system.

4.1 Offline Training and Indexing

We use search logs in the past one week to train the model by distributed Tensorflow [1] and update the model parameters daily. Note that we do not use sequential training [33] to do the A/B test. Since the base model has been trained by lots of data (several months or even one year), it is difficult to catch up with the same volume of data for the new testing model. As illustrated in Figure 4, the deployment system of MGDSPR is an offline to online architecture. At the offline phrase, *build service* optimizes and constructs a user/query network extracted from the user tower, which is passed to *real-time prediction* platform. All the item embeddings are simultaneously exported from the item tower and transmitted to an approximate near neighbor (ANN) indexing system. The total number of items is about one hundred millions. They are placed in multiple columns (6 in our system) because of the enormous amounts. Each column of the ANN builds indexes of embeddings by HC (hierarchical clustering) algorithm with K-means and INT8 quantization to promote storage and search efficiency. The training sample size is 4 million for HC, and the max scan ratio is 0.01.

4.2 Online Serving

The user/query network and item embedding indexes are published in an online environment after offline indexing. When a user issues a query, user history behaviors and the query are fed into a real-time prediction platform for online inference. The *ANN search module* then distributively seeks top- K ($K = 9600$ in our system) results from indexes of multi-columns (referred to as $n = 6$ columns). Each column returns the same size of K/n . The indexing retrieval accuracy is 98% accompanied with 10 milliseconds of retrieval latency.

4.3 Relevance Control

After a long period of practice, we find that although embedding-based retrieval has advantages in personalization and fuzzy matching, it often leads to more search bad cases due to lack of exact matching [?] to the key terms of a query. The *key terms of a query* are referred to as words of brand, type, color, etc., which are significant to product search relevance. For instance, a user is searching for *Adidas sports shoes*. Items of *Nike sports shoes* are similar to the query in the embedding space and hence will appear in the top- K results with high probability. However, this is not the user intent and will harm user experience. Hence, we add an inverted index based *boolean matching* module on top of the ANN results. Boolean matching aims to filter out items that do not contain key query terms in their titles. The final search results can then be expressed as:

(ANN results) and (Brand: Adidas) and (Category: Shoes).

Generally, we predefine the rule of key terms according to query understanding, e.g., words of brand, color, style, and audience. Note that Facebook [14] uses an embedding-based method to enhance boolean expression and achieves fuzzy matching, while we use boolean matching to improve retrieval relevance of the EBR system.

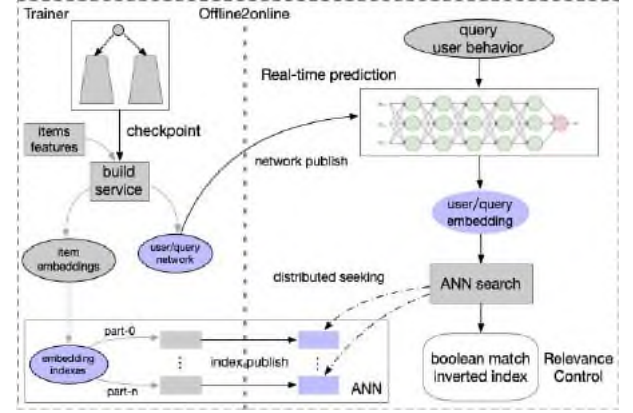


Figure 4: Deployment system of our MGDSPR model.

5 EXPERIMENTS

Here, we introduce evaluation metrics, implementation details, datasets, and offline and online experimental results of our method including its effect in the search system.

5.1 Evaluation Metrics

5.1.1 Offline Evaluation. We use the metric of Recall@ K to evaluate the offline performance. Specifically, given a query q_u , the items clicked or purchased by the user u are regarded as the target set $T = \{t_1, \dots, t_N\}$, and the top- K items returned by a model are regarded as the retrieval set $I = \{i_1, \dots, i_K\}$. Recall@ K is defined as

$$\text{Recall@}K = \frac{\sum_{i=1}^K \mathbb{I}(i \in T)}{N}. \quad (22)$$

Empirically, during the retrieval phase, we find that the AUC metric has no positive correlation with the online Gross Merchandise Volume (GMV) metric, while the recall metric does. Also, we add T_{rec} , the records relevant to the current query that were not purchased in search but elsewhere (e.g., recommender system) in Taobao Mobile App, to the testing set.

In Taobao search, we also pay attention to the relevance of retrieved products (related to user experience). Due to the large amount of test data, we use an online well-trained relevance model (its AUC for human-labeled data is **0.915**) instead of expensive human evaluation to calculate the proportion of products with good relevance (abbreviated as good rate and denoted as P_{good}) in the retrieval set I , which is defined as

$$P_{good} = \frac{\sum_{i=1}^K \mathbb{I}(i)}{K}, \quad (23)$$

where $\mathbb{I}(\cdot)$ is an indicator function. When item i is rated as good by the relevance model, the function value is 1, otherwise 0. It is not appropriate to use the AUC metric to evaluate whether our model can return more products with good relevance because it evaluates the order of the set elements rather than the number of the “good” elements. K is experimentally set to be 1,000.

Meanwhile, to analyze the effect of our model on each stage of the search system, we also count the number of items in the retrieval set I that participate in each follow-up stage. Given a

retrieval set $I = \{i_1, \dots, i_K\}$, every time it goes through a stage (such as the relevance control module, pre-ranking, and ranking), the number of items will decrease, resulting in $I_{left} = \{i_1, \dots, i_k\}$, $k < K$. Therefore, we calculate the number of items in I_{left} after going through each phase, and use Num_{prank} and Num_{rank} to denote the number of items that enter the pre-ranking and ranking stages. For a total retrieval set $\mathcal{I} = \{I_1, \dots, I_i, \dots, I_N\}$ of N queries, the calculation of Num_{prank} and Num_{rank} are averaged by N .

5.1.2 Online Evaluation. We consider the most important online metrics: GMV , P_{good} , and P_{h_good} . GMV is the Gross Merchandise Volume, which is defined as

$$GMV = \#pay \text{ amount}. \quad (24)$$

In addition to the amount of online income, we also consider user search experience by the P_{good} and P_{h_good} metrics (defined in Eq. (23)). Precisely, both P_{good} and P_{h_good} calculate the good rate of the item set displayed to users, but P_{good} is determined by the relevance model while P_{h_good} is determined by humans.

5.2 Implementation Details

The maximum length T of real-time, short-term, and long-term sequences are 50, 100, and 100, respectively. We use attention with a mask to calculate those sequences whose length is less than T . The dimensions of the user tower, item tower, behavior sequence, and hidden unit of LSTM are all set to 128. The batch size is set to 256. We use LSTM of two layers with dropout (probability 0.2) and residual network [19] between vertical LSTM stacks. The number of heads in self-attention is set to 8. The parameters a and b of uniform distribution U and the number of generated samples N are set to 0.4, 0.6 and 684, respectively. The temperature parameter τ of softmax is set to 2. All parameters are orthogonally initialized and learned from scratch. The experiments are run on the distributed TensorFlow platform [1] using 20 parameter servers and 100 GPU (Tesla P100) workers. The AdaGrad optimizer [7] is employed with an initial learning rate of 0.1, which can improve the robustness of SGD for training large-scale networks [6]. We also adopt gradient clip when the norm of gradient exceeds a threshold of 3. The training process converges at about 35 million steps for about 54 hours.

5.3 Datasets

5.3.1 Large-scale Industrial Offline DataSet. We collect search logs of user clicks and purchases for 8 consecutive days from online **Mobile Taobao App** in December 2020, and filter the spam users. The training set comprises samples from the first 7 consecutive days (a total of 4.7 billion records). For evaluation, we randomly sample 1 million search records T and 0.5 million purchase logs T_{rec} from the recommender system in the 8-th day. We have also tried to extend the timeframe of training data to 10 days, but there is no significant benefit, indicating billions of data can effectively prevent the model from overfitting. The size of the candidate item set is consistent with the online environment, *i.e.*, about 100 million.

5.3.2 Online Dataset. We deploy a well-trained MGDSPR in the Taobao search production environment containing hundreds of millions of user query requests. The size of the item candidate set is about 100 million, covering the most active products at Taobao.

Table 1: Comparison with the strong baseline α -DNN on a large-scale industrial offline dataset. Num_{prank} is the number of products that flow into the follow-up pre-ranking phase. P_{good} is the good rate. Relative improvements are shown in parentheses.

Methods	Recall@1000	P_{good}	P_{f_good}	Num_{prank}
α -DNN [5]	82.6%	70.6%	83.2%	769
MGDSPR	84.7%(+2.5%)	80.0%(+13.3%)	84.1%(+1.1%)	815(+6.0%)

5.4 Offline Experimental Results

Previously, our embedding-based retrieval system adopts the DNN architecture proposed in [5], but uses more user behaviors and statistical features (inherited from the ranking model), which has been experimentally verified to be effective to some extent. Specifically, we concatenate the vectors of user behaviors (obtained by mean-pooling) and statistical features (*e.g.*, Unique Visitor (UV), Item Page View (IPV)) and feed it into a multi-layer feed-forward neural network. We refer to it as a strong baseline α -DNN. In addition, adding statistical features to MGDSPR has no benefit in the metric of *recall*, so we delete them but keep the user behavior sequences.

5.4.1 Comparison with the Strong Baseline. As mentioned in Section 5.1.1, we report the metrics of Recall@K, good rate, and Num_{prank} . Note that we report P_{good} on both the retrieval set I (denoted as P_{good}) and the filtered set I_{left} (denoted as P_{f_good}). As shown in Table 1, MGDSPR improves over α -DNN by 2.5%, 13.3% and 6.0% in Recall@1000, P_{good} and P_{f_good} respectively, indicating it can retrieve more products with good relevance and improve the quality of the retrieval set. Comparing P_{good} and P_{f_good} shows our relevance control module enhances retrieval relevance.

Table 2: Ablation study of MGDSPR.

Methods	Recall@1000	P_{good}
MGDSPR	85.6%	71.2%
MGDSPR + mgs	86.0%	71.6%
MGDSPR + trm	86.4%	71.4%
MGDSPR + τ	85.5%	79.0%
MGDSPR + mgs + trm + τ	86.8%	79.2%
MGDSPR + I_{mix}	83.6%	75.6%
MGDSPR + all	84.7%	80.0%

5.4.2 Ablation Study. We study the effectiveness of each component of MGDSPR by adding only one component at a time. Specifically, MGDSPR have the following four components: 1) Multi-Granular Semantic unit (*i.e.*, Eq. (8), denoted as mgs); 2) dynamic fusion of semantics and personalization (*i.e.*, Eq. (15), denoted as trm); 3) the temperature parameter τ of softmax (denoted as τ); 4) the relevance-improving hard negative samples (denoted as I_{mix}). Note that here we focus on the model’s performance, so good rate P_{good} is calculated on the retrieval set I instead of I_{left} .

As shown in Table 2, both the multi-granular semantics unit mgs and trm can improve the metrics of Recall@1000 and P_{good} , indicating the effectiveness of multi-granular semantics and dynamic

fusion. The temperature parameter τ and relevance-improving hard negative samples I_{mix} make the model retrieve more relevant products in terms of much higher good rate P_{good} . Comparing MGDSR+all and MGDSR or MGDSR+ $mgs+trm+\tau$, we observe there is a trade-off between recall and relevance even in search scenarios, which may indicate excessive personalization in our system.

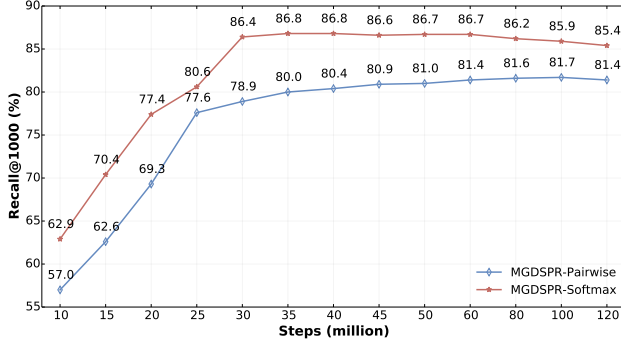


Figure 5: Convergence comparison of the softmax cross-entropy and hinge (pairwise) loss functions. The X-axis denotes the number of training steps, and the Y-axis denotes the corresponding test Recall@1000 score.

5.4.3 Convergence Analysis. We investigate the performance of MGDSR using softmax cross-entropy and pairwise loss [21, 34] as the training objective, respectively. We report the test Recall@1000 score with respect to the number of training steps. As shown in Figure 5, the softmax function’s global comparison capability makes training and testing more consistent, achieving faster convergence and better performance. In fact, it only takes about three days for the softmax loss to converge while about six days for the pairwise loss. Note that the margin parameter used in the hinge loss has been carefully tuned.

5.4.4 Hyper-parameter Analysis. We perform an investigation of the hyper-parameters τ (for noise smoothing) and N (the number of generated relevance-improving hard negative samples) to demonstrate how they affect the good rate P_{good} . We conduct the evaluation by varying τ (or N) while fixing the other parameters.

As mentioned in Section 3.4.1, we can increase τ to smooth the noisy training data and thus alleviate the effect of insufficient relevance due to overfitting users’ click records. As shown in Figure 6, $\tau = 0.1$ decreases relevance, indicating that the training data does have noise. Also, every non-zero value of N gives better relevance than $N = 0$, showing that the generated hard negative samples can improve good rate P_{good} . Further, the good rate P_{good} reaches its maximum at $N = 684$ and then decreases, indicating that simply increasing the number of samples cannot bring more benefits.

5.5 Online A/B Test

We deploy MGDSR on Taobao Product Search and compare it with the strong baseline α -DNN. As aforementioned, to improve user experience, our relevance control module (introduced in Section 4.3)

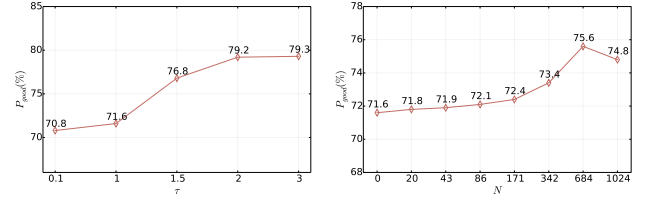


Figure 6: The influence of τ and N on the good rate P_{good} .

will filter out some products retrieved by the EBR system, resulting in low utilization of online computing resources. Therefore, apart from GMV, P_{good} , and P_{h_good} , we report the number of products that participate in the pre-ranking and ranking phases (denoted as Num_{prank} and Num_{rank}) to analyze the model’s effect on our search system.

Table 3: The improvements of MGDSR in Num_{prank} , Num_{rank} , P_{good} , and P_{h_good} compared with the previous model deployed on Taobao Product Search. The last two columns only report relative values that are calculated on the exposed item set.

Methods	Num_{prank}	Num_{rank}	P_{good}	P_{h_good}
Baseline	4070	1390	-	-
MGDSR	4987(+22.53%)	1901(+36.76%)	+1.0%	+0.35%

As shown in Table 3, after being filtered by the relevance control module, the number of products retrieved by MGDSR that enter the pre-ranking and ranking phases increases by 22.53% and 36.76%, respectively. Obviously, MGDSR retrieves more products with good relevance and effectively improves the utilization of computing resources. Besides, MGDSR achieves higher good rates P_{good} and P_{h_good} of exposure relevance, and thus can display more relevant products to users.

Table 4: Online A/B test of MGDSR. The improvements are averaged over 10 days in Jan 2021.

Launched Platform	GMV	#Transactions
Taobao Search on Mobile	+0.77%	+0.33%

Finally, we report the 10-day average of GMV improvements (by removing cheating traffic) achieved by MGDSR. We also include the corresponding number of transactions (denoted as #Transactions) to increase results confidence. As shown in Table 4, MGDSR improves GMV and #Transactions by 0.77% and 0.33%, respectively. Considering the billions of transaction amounts per day in Taobao Search, 0.77% improvement is already tens of millions of transaction amounts, indicating MGDSR can significantly better satisfy users.

6 CONCLUSION

This paper proposes a practical embedding-based product retrieval model, named Multi-Grained Deep Semantic Product Retrieval

(MGDSPR). It addresses model performance degradation and online computing resource waste due to the low retrieval relevance in the previous EBR system of Taobao Product Search. Meanwhile, we share the lessons learned from solving those problems, including model design and its effect on each stage of the search system, selection of offline metrics and test data, and relevance control of the EBR system. We verify the effectiveness of MGDSPR experimentally by offline and online A/B tests. Furthermore, we have deployed MGDSPR on Taobao Product Search to serve hundreds of millions of users in real time. Moreover, we also introduce the online architecture of our search system and the deployment scheme of the retrieval model to promote development of the community.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their helpful feedbacks. This research was supported by the Alibaba Innovative Research project P0034058 (ZGAL).

REFERENCES

- [1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.
- [2] Qingyao Ai, Daniel N Hill, SVN Vishwanathan, and W Bruce Croft. 2019. A zero attention model for personalized product search. In *Proceedings of the 28th ACM International Conference on Information & Knowledge Management*. 379–388.
- [3] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 645–654.
- [4] Yoshua Bengio and Jean-Sébastien Senécal. 2008. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Transactions on Neural Networks* 19, 4 (2008), 713–722.
- [5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 191–198.
- [6] Jeffrey Dean, Greg S Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V Le, Mark Z Mao, Marc Aurelio Ranzato, Andrew Senior, Paul Tucker, et al. 2012. Large scale distributed deep networks. (2012).
- [7] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, 7 (2011).
- [8] Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. MOBIUS: towards the next generation of query-ad matching in baidu's sponsored search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2509–2517.
- [9] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482* (2019).
- [10] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing search results using hierarchical RNN with query-aware attention. In *Proceedings of the 27th ACM International Conference on Information & Knowledge Management*. 347–356.
- [11] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International Conference on Information & Knowledge Management*. 55–64.
- [12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [13] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2015. Convolutional neural network architectures for matching natural language sentences. *arXiv preprint arXiv:1503.03244* (2015).
- [14] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.
- [15] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using click through data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. 2333–2338.
- [16] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007* (2014).
- [17] Shichen Liu, Fei Xiao, Wenwu Ou, and Luo Si. 2017. Cascade ranking for operational e-commerce search. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1557–1565.
- [18] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information & Knowledge Management*. 2635–2643.
- [19] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182* (2017).
- [20] Thanh V Nguyen, Nikhil Rao, and Karthik Subbian. 2020. Learning Robust Models for e-Commerce Product Search. *arXiv preprint arXiv:2005.03624* (2020).
- [21] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2876–2885.
- [22] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 4 (2016), 694–707.
- [23] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.
- [24] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.
- [25] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Information & Knowledge Management*. 101–110.
- [26] Daria Sorokina and Erick Cantu-Paz. 2016. Amazon search: The joy of ranking products. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 459–460.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).
- [28] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial rnn. *arXiv preprint arXiv:1604.04378* (2016).
- [29] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2020. "Click" Is Not Equal to "Like": Counterfactual Recommendation for Mitigating Clickbait Issue. *arXiv preprint arXiv:2009.09945* (2020).
- [30] Tao Wu, Ellie Ka-In Chio, Heng-Tze Cheng, Yu Du, Steffen Rendle, Dima Kuzmin, Ritesh Agarwal, Li Zhang, John Anderson, Sarvejit Singh, et al. 2020. Zero-Shot Heterogeneous Transfer Learning from Recommender Systems to Cold-Start Search Retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2821–2828.
- [31] Rong Xiao, Jianhui Ji, Baoliang Cui, Haihong Tang, Wenwu Ou, Yanghua Xiao, Jiwei Tan, and Xuan Ju. 2019. Weakly Supervised Co-Training of Query Rewriting and Semantic Matching for e-Commerce. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 402–410.
- [32] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 55–64.
- [33] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.
- [34] Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wen-Yun Yang. 2020. Towards Personalized and Semantic Retrieval: An End-to-End Solution for E-commerce Search via Embedding Learning. *arXiv preprint arXiv:2006.02282* (2020).
- [35] Jing Zhang and Dacheng Tao. 2020. Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet of Things Journal* (2020).
- [36] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weiwei Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5941–5948.
- [37] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.
- [38] Justin Zobel and Alistair Moffat. 2006. Inverted files for text search engines. *ACM Computing Surveys (CSUR)* 38, 2 (2006), 6–es.

Mining the Benefits of Two-stage and One-stage HOI Detection

Aixi Zhang^{1*} Yue Liao^{2*} Si Liu^{2†}
Miao Lu¹ Yongliang Wang¹ Chen Gao² Xiaobo Li¹
¹Alibaba Group ²Beihang University

Abstract

Two-stage methods have dominated Human-Object Interaction (HOI) detection for several years. Recently, one-stage HOI detection methods have become popular. In this paper, we aim to explore the essential pros and cons of two-stage and one-stage methods. With this as the goal, we find that conventional two-stage methods mainly suffer from positioning positive interactive human-object pairs, while one-stage methods are challenging to make an appropriate trade-off on multi-task learning, *i.e.*, object detection, and interaction classification. Therefore, a core problem is how to take the essence and discard the dregs from the conventional two types of methods. To this end, we propose a novel one-stage framework with disentangling human-object detection and interaction classification in a cascade manner. In detail, we first design a human-object pair generator based on a state-of-the-art one-stage HOI detector by removing the interaction classification module or head and then design a relatively isolated interaction classifier to classify each human-object pair. Two cascade decoders in our proposed framework can focus on one specific task, detection or interaction classification. In terms of the specific implementation, we adopt a transformer-based HOI detector as our base model. The newly introduced disentangling paradigm outperforms existing methods by a large margin, with a significant relative mAP gain of 9.32% on HICO-Det. The source codes are available at <https://github.com/YueLiao/CDN>.

1 Introduction

The goal of Human-Object Interaction (HOI) detection [2, 20, 6, 18, 7, 8, 17, 3] is to make a machine detailedly understand human activities from a static image. Human activities in this task are abstracted as a set of <human, object, action> HOI triplets. Thus, an HOI detector is required to locate human-object pairs and classify their corresponding action simultaneously. Based on this definition, we can summarize conventional HOI detection methods into two paradigms, *i.e.*, two-stage methods, and one-stage methods. These two paradigms have made significant progress with the development of deep learning, but both paradigms still have their shortcomings due to their structural design. This paper aims to present a detailed analysis of methods under these two paradigms and propose a solution to mine the benefits of two-stage and one-stage methods.

We first take a closer look at the conventional two-stage and one-stage HOI detectors. Conventional two-stage methods [6, 2, 18, 5] are mostly with a serial architecture. As shown in Figure 1 (a), two-stage methods detect humans and objects first and then feeds the human-object pairs, which are generated by matching humans and objects one by one, into an interaction classifier. The serial architecture suffers from locating the interactive human-object pairs under the interference of a large number of negative pairs only based on local region features. Otherwise, the efficiency of

*Equal contribution

†Corresponding author (liusi@buaa.edu.cn)

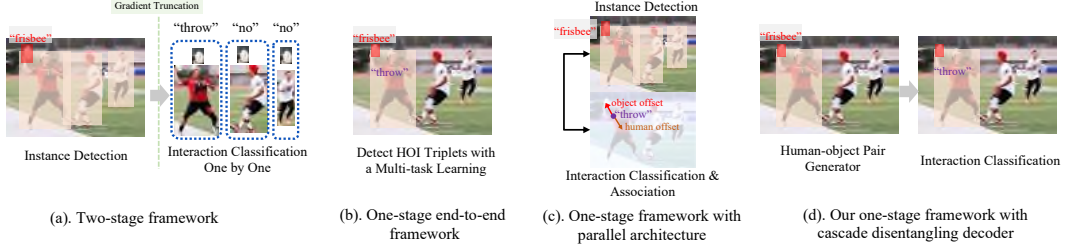


Figure 1: (a) Two-stage framework, (b) one-stage end-to-end framework, (c) one-stage framework with parallel architecture, and (d) our one-stage framework with a cascade disentangling head.

two-stage methods is also limited by the serial architecture. To alleviate these problems, one-stage methods [20, 13, 39, 3, 28, 14] are proposed to detect the HOI triplets directly and break HOI detection as multi-task learning, *i.e.*, human-object detection and interaction classification, which is shown in Figure 1(b). Therefore, one-stage methods can easily focus on the interactive human-object pairs and effectively extract corresponding features in an end-to-end manner. However, it is difficult for a single model to make a good trade-off on multi-task learning since human-object detection and interaction classification are two very different tasks, which requires the model to focus on different visual features. As shown in Figure 1(c), though some previous methods [20, 3] design two parallel branches to detect instances and predict interaction respectively, the interaction classification branch still needs to regress additional offsets to associate humans and objects. Thus the interaction branch is also required to make a trade-off between interaction classification and human and object positioning.

Therefore, the intuitive idea is to take the essence and discard the dregs from the two paradigms. To attain this, we propose a novel end-to-end one-stage framework with disentangling human-object detection and interaction classification in a cascade manner, namely Cascade Disentangling Network (CDN). The original intention of our framework is to keep the advantages of conventional one-stage methods, **directly and accurately locating the interactive human-object pairs**, and bring the advantages of two-stage methods into our one-stage framework, **disentangling human-object detection and interaction classification**. As shown in Figure 1(d), in our proposed framework, we design a human-object pair decoder based on the one-stage paradigm by removing the interaction classification function, namely HO-PD, and following an isolated interaction classifier. To instantiate our idea with an end-to-end manner, we design the HO-PD based on the previous state-of-the-art one-stage transformer-based HOI detector, HOI-Trans [39] and QPIC [28], where we remove the interaction classification head for each query and make it focus on human-object pairs detection. Otherwise, we design an independent HOI decoder for interaction classification to make the interaction classification unaffected by human-object detection. Therefore, there exists a core problem, *i.e.*, how to link the human-object pair and the corresponding action class. To address this problem, we initialize the query embedding of the HOI decoder with the output of the last layer of the HO-PD. In this case, the HOI decoder is able to learn the corresponding action category under the guidance of the query embedding and free out from the human-object detection task. Moreover, we design a decoupling dynamic re-weighting manner to handle the long-tailed problems in HOI detection.

Our contributions can be summarized threefold: (1) We conduct a detailed analysis of two conventional HOI detection paradigms, *i.e.*, two-stage and one-stage. (2) We propose a novel one-stage framework with a cascade disentangling decoder to combine the advantages of two-stage and one-stage methods. (3) Our method outperforms previous state-of-the-art methods by a large margin on the HOI detection task, especially achieves a 25.35% performance gain on rare classes of HICO-Det.

2 Analysis of Two-stage and One-stage HOI detectors

We first introduce a unified formulation for the HOI detection problem. Given a human-centric image I , the model $T(\cdot)$ is required to predict a set of HOI triplets $S = \{(b_i^h, b_i^o, a_i), i \in \{1, 2, \dots, K\}\}$, where b_i^h , b_i^o and a_i denotes a human bounding-box, an object bounding-box and their corresponding action category, respectively.

Two-stage HOI detector. Two-stage detectors can be regarded as an instance-driven manner, detecting instances first and predicting interaction based on the detected instances. The two-stage detector divides $T(\cdot)$ into two stages, *i.e.*, detection $T_d(\cdot)$ and interaction classification $T_c(\cdot)$. In the first stage,

we suppose that $T_d(\cdot)$ produces M human bounding-boxes and N object bounding-boxes. Here the ‘object’ is a universal object which includes human as one class. Therefore, $T_d(\cdot)$ generates $M \times N$ human-object pairs. In general, the number of true-positive interactive human-object pairs, denoted as K' , is much smaller than $M \times N$. However, in the second stage, $T_c(\cdot)$ needs to scan all $M \times N$ pairs one by one and predict an action category with its corresponding confidence score. In this case, $T_c(\cdot)$ is required to inference $M \times N$ times to find K' interactive pairs from $M \times N$ pairs. We argue that this manner causes three problems. Firstly, these models produce a more additional computational cost, whose time complexity is $\mathcal{O}(M \times N) \gg \mathcal{O}(K')$. Secondly, the imbalance between positive and negative samples makes the model easily overfit to negative samples. Thus the model tends to assign a ‘no-interaction’ class for human-object pairs with very high confidence, suppressing the true-positive samples. Thirdly, the accuracy of interaction classification is influenced by the non-end-to-end pipeline. Because the interaction classification is mostly based on the region features extracted by $T_d(\cdot)$, while the core of $T_d(\cdot)$ is to regress bounding-boxes and its extracted features pay more attention to the edge of regions, thereby such features are not good options for interaction classification, which needs more context. However, it is an excellent property for two-stage methods that disentangling detection and interaction classification makes each stage focus on its task and produce good results in each stage.

One-stage HOI detector. As for one-stage methods, they detect all HOI triplets S directly and simultaneously with an end-to-end framework. Such paradigm has greatly relieved the three problems of two-stage methods, especially for efficiency, where the time complexity is reduced to $\mathcal{O}(K')$. Most one-stage methods are interaction-driven, which directly locate the interaction point [20] or interactive human-object pairs [39], thereby reducing negative sample interference. However, coupling human-object detection and interaction classification limit their performance because it is hard to generate a unified feature representation for two very different tasks. Though the parallel one-stage methods break HOI detection into two parallel branches, their interaction branch still suffers from multi-task learning. Specifically, the optimization target of interaction branch is $\mathcal{P}(e_h, e_o, a|\mathbf{V})$, where e_h and e_o are associative embeddings, *e.g.*, offset, to match interaction with human and object respectively. Therefore, even if detection is organized as an independent branch, the interaction branch must position humans and objects for the association. The set-based detectors couple detection and interaction completely, whose optimization function is $\mathcal{P}(b^h, b^o, a|\mathbf{V})$.

Next, we introduce a simple one-stage framework with disentangling human-object detection and interaction classification, namely CDN, to mine the benefits of two-stage and one-stage HOI detectors. Our CDN disentangles the original set-based one-stage optimization function into two cascade decoders. Firstly, we predict human-object pair by $\mathcal{P}(b^h, b^o|\mathbf{V})$. Secondly, we apply an isolated decoder to predict the action category by $\mathcal{P}(a|\mathbf{V}, b^h, b^o)$. More details are in the following.

3 Method

In this section, we will present a detailed introduction to the pipeline of our proposed CDN. In section 3.1, we present an overview of our framework and briefly introduce the pipeline. In section 3.2, we introduce the visual feature extractor. The cascade disentangling HOI decoder is introduced in section 3.3. Section 3.4 introduces a novel dynamic re-weighting mechanism that mitigates the long-tailed problem. The detailed training process and post-processing are discussed in section 3.5.

3.1 Overview

The architecture of our proposed CDN is illustrated in Figure 2. Our CDN is organized in a cascade manner with a visual feature extractor. Given an image, we first follow transformer-based detection methods [1, 39] to apply a CNN followed by a transformer encoder architecture to extract visual features into a sequence. Then we detect HOI triplets in two cascade decoders. Firstly, we apply the Human-Object Pair Decoder (HO-PD) to predict a set of human-object bounding-boxes pairs based on a set of learnable queries. Next, taking the output of the last layer of HO-PD as queries, an isolated interaction decoder is utilized to predict the action category for each query. Finally, the HOI triplets are formed by the output of the above two cascade decoders.

3.2 Visual Feature Extractor

We define the visual feature extractor by combining a CNN and a transformer encoder. Fed with an input image I with shape (H, W, C) , the CNN generates a feature map of shape (H', W', D_b) .

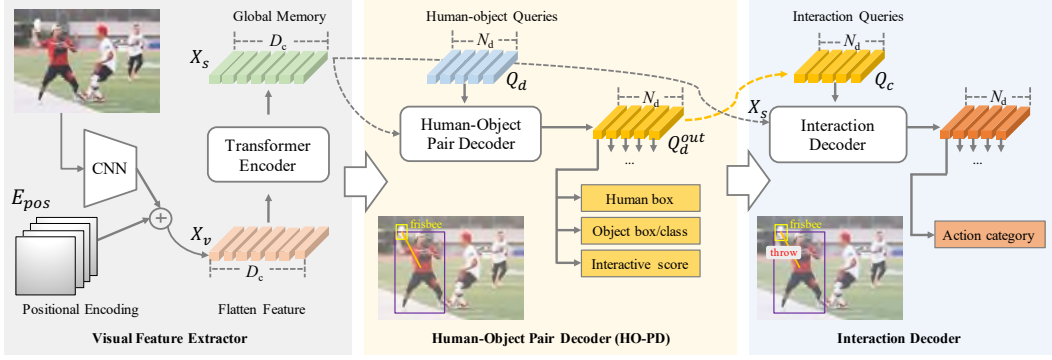


Figure 2: **The framework of our CDN.** It is comprised of three components: *Visual Feature Extractor*, *Human-Object Pair Decoder (HO-PD)* and *Interaction Decoder*. We first apply a CNN-transformer combined architecture to extract sequenced visual features X_s . Then, we divide HOI detection into two cascade transformer-based decoders. Firstly, we regress the human-object bounding-box pairs based on X_s and a set of random-initialized queries Q_d by HO-PD. The interactive score is from a binary classification to determine whether the human-object pair is an interactive pair or not. Secondly, we predict one or many action categories for each predicted human-object pairs, where we take the output of HO-PD Q_d^{out} to initialize the interaction queries Q_c and aggregate information with X_s . Finally, the HOI triplets are formed by the output of the cascade decoders.

Then, D_b is reduced to D_c by a projection convolution layer with a kernel size 1×1 . Next, a flatten operator is used to generating the flatten feature $X_v \in \mathcal{R}^{(H' \times W') \times D_c}$ by collapsing the spatial dimensions into one dimension. This flatten feature is then fed into a transformer encoder and the position encoding $E_{pos} \in \mathcal{R}^{(H' \times W') \times D_c}$, which distinguishes the relative position in the sequence $X_s \in \mathcal{R}^{(H' \times W') \times D_c}$. Thanks to the multi-head self-attention mechanism, the transformer encoder produces a feature map with richer contextual information by summarizing global information. The output of the encoder is denoted as global memory with a dimension of D_c .

3.3 Cascade Disentangling HOI Decoder

The cascade disentangling HOI decoder consists of two decoders: Human-Object Pair Decoder (HO-PD) and interaction decoder. Both decoders share the same architecture, a transformer-based decoder, with independent weights. In this subsection, we first introduce the general architecture of the decoder and then elaborate on the two decoders in detail, respectively.

Transformer-based decoder. We follow the transformer-based object detector DETR [1] to design the basic architecture in our cascade disentangling HOI decoder. We apply N transformer decoder layers for each decoder and equip each decoder layer with several FFN heads for intermediate supervision. Specifically, each decoder layer is comprised of a self-attention module and a multi-head co-attention module. During feed-forward, fed into a set of learnable queries $Q \in \mathcal{R}^{N_q \times C_q}$, each decoder layer first applies a self-attention module on all queries and then conducts a multi-head co-attention operation between queries and the sequenced visual features, and outputs a set of updated queries. For the FFN heads, each head is comprised of one or several MLP branches, and each branch is for a specific task, *e.g.*, regression, or classification. All queries share the same FFN heads. Therefore, each decoder can be simply represented as:

$$P = f(Q, X_s, E_{pos}). \quad (1)$$

Besides, the number of queries N_q is determined by the number of positive samples of an image.

HO-PD. Firstly, we design the HO-PD to predict a set of human-object pairs from the sequenced visual features. To this end, we first randomly initialize a set of learnable queries $Q_d \in \mathcal{R}^{N_d \times C_q}$ as HO queries. Then we apply a transformer-based decoder, which takes HO queries Q_d and sequenced visual features as input and applies three FFN heads for each query to predict human bounding-box, object bounding-box, and object class, which form a human-object pair. We also utilize an additional interactive score head to simply determine whether the human-object pair is an interactive pair or not by a binary classification. In this case, P is instantiated as P_{ho} , which is consist of a set of

human-object pairs $\{(b_i^h, b_i^o), i \in \{1, 2, \dots, N_d\}\}$. Thus, HO-PD can be denoted as:

$$P_{ho} = f_d(Q_d, X_s, E_{pos}). \quad (2)$$

In addition, we keep the output queries of the last layer of HO-PD as Q_d^{out} for the following step.

Interaction Decoder. Secondly, we propose the interaction decoder to classify the human-object queries and assign one or several action categories for each human-object query. To classify each human-object query one-to-one, we initialize Q_c with the output of HO-PD Q_d^{out} . In this way, Q_d^{out} can provide prior knowledge to guide Q_c to learn the corresponding action categories for each human-object query. The other components and inputs are the same as HO-PD, which conducts self-attention among queries and co-attention with X_s and E_{pos} . The final output is a set of action categories $P_{cls} : \{a_i, i \in \{1, 2, \dots, N_d\}\}$. Therefore, this process can be formulated as:

$$P_{cls} = f_{cls}(Q_d^{out}, X_s, E_{pos}). \quad (3)$$

In our proposed cascade disentangling HOI decoder, the task of HOI detection is disentangled into two relatively independent steps: human-object pairs detection and interaction classification. Therefore, each step can aggregate more related features to concentrate on its corresponding task.

3.4 Decoupling Dynamic Re-weighting

The HOI datasets usually have long-tail class distribution for both object class and action class. To alleviate the long-tail problem, we design a dynamic re-weighting mechanism for further improvements with a decoupling training strategy. In detail, we first train the whole model with regular losses. Then, we freeze the parameters of the visual feature extractor and only train the cascade disentangling decoders with a relatively small learning rate and the designed dynamic re-weighted losses.

During decoupling training, at each iteration, we apply two similar queues to accumulate number of each object class or action class. The queues are used as memory banks to accumulate training samples and truncate the accumulation with length L_Q as sliding windows. In detail, Q_o with length L_Q^o to accumulate object number N_i^o for each object class $i \in \{1, 2, \dots, C_o\}$, and Q_a with length L_Q^a to accumulate interaction number N_i^a for each action category $i \in \{1, 2, \dots, C_a\}$. The dynamic re-weighting coefficients $w_{dynamic}$ are presented as follow:

$$w_i^a|_{i \in \{1, 2, \dots, C_a\}} = \left(\frac{\sum_{i=1}^{C_a} N_i}{N_i} \right)^{p_a}, \quad w_{bg}^a = \left(\frac{\sum_{i=1}^{C_a} N_i}{N_{bg}^a} \right)^{p_a}, \quad (4)$$

$$w_i^o|_{i \in \{1, 2, \dots, C_o\}} = \left(\frac{\sum_{i=1}^{C_o} N_i}{N_i} \right)^{p_o}, \quad w_{bg}^o = \left(\frac{\sum_{i=1}^{C_o} N_i}{N_{bg}^o} \right)^{p_o}, \quad (5)$$

where N_i is the number of accumulated positive samples of category i by the queues Q_o and Q_a , N_{bg} is the number of accumulated background samples, C is the number of categories, and exponent p is a hyper-parameter that adapts the magnitude of mitigation. Specifically, the weight of background class, w_{bg} , is designed to balance the positives and negatives. For the stability of the dynamic re-weighted training, the weight coefficients are initialized as w_{static} with those calculated by 4 and 5 using the static number of object and action categories. The final dynamic weights are given as $w = \gamma w_{static} + (1 - \gamma) w_{dynamic}$, where γ is a smooth factor, given as $\min(0.999^{L_Q}, 0.9)$. The factor γ transits w from w_{static} to $w_{dynamic}$ with the increasing of L_Q . Finally, the weights are used to the classification losses in a traditional way by multiplying each coefficient to each class and then calculating the summation.

3.5 Training and Post-processing

In this section, we introduce the training and inference processes in detail. Especially, we will introduce a novel Pair-wise Non-Maximal Suppression (PNMS) strategy in the inference process.

Training. Following the set-based training process of HOI-Trans [39] and QPIC [28], we first match each ground-truth with its best-matching prediction by the bipartite matching with the Hungarian algorithm. Then the loss is produced between the matched predictions and the corresponding ground truths for the final back-propagation. During matching, we consider the predictions of two cascade decoders together. The loss of CDN follows QPIC which is composed by five parts: the box regression

loss L_b , the intersection-over-union loss L_{GIoU} [26], the interactive score loss L_p , the object class loss L_c^o , and the action category loss L_c^a . The target loss is the weighted sum of these parts as:

$$L = \sum_{k \in (h,o)} (\lambda_b L_b^k + \lambda_{GIoU} L_{GIoU}^k) + \lambda_p L_p + \lambda_o L_c^o + \lambda_a L_c^a, \quad (6)$$

where λ_b , λ_{GIoU} , λ_p , λ_o and λ_a are the hyper-parameters for adjusting the weights of each loss.

Inference. The inference process is to composite the output of instance-related FFNs and the interaction-related FFN to form HOI triplets. By our cascade disentangling decoder architecture, the instance queries and the interaction queries are one-to-one corresponding, therefore, the five components $\langle \text{human bounding box}, \text{object bounding box}, \text{object class}, \text{interactive score}, \text{action class} \rangle$ can be homologous in each of the N_d dimensions per FFN head. Formally, we generate the i -th output prediction as $\langle b_i^h, b_i^o, \arg\max_k c_i^{hoi}(k) \rangle$. The HOI triplet score c_i^{hoi} is given by $c_i^{hoi} = c_i^a c_i^o c_i^p$, where c_i^a and c_i^o are the scores of interaction and object classification, respectively, and c_i^p is the interactive score from the interactive FFN head for the query vector being an human-object pair.

PNMS. After sorting c_i^{hoi} in descending order and generating the top K HOI triplets, we design a pair-wise non-maximal suppression (PNMS) method to further filter out human-object pairs from pair-wise bounding boxes overlapping perspective. For two HOI triplets m and n , the pair-wise overlap $PIoU$ is calculated as:

$$PIoU(m, n) = \left(\frac{I(b_m^h, b_n^h)}{U(b_m^h, b_n^h)} \right)^\alpha \left(\frac{I(b_m^o, b_n^o)}{U(b_m^o, b_n^o)} \right)^\beta, \quad (7)$$

where the operators I and U compute the intersection and union areas between the two boxes of m and n , respectively. α and β are the balancing parameters between humans and objects.

4 Experiments

In this section, we conduct comprehensive experiments to demonstrate the superiority of our designed CDN. In section 4.1, we briefly introduce the experimental benchmarks. Section 4.2 presents implementation details. Next, It is a detailed experimental comparison and analysis of two-stage and one-stage methods in section 4.3. In section 4.4, we compare our methods with the previous state-of-the-art methods. The ablation studies and components analysis are included in 4.5.

4.1 Datasets and Evaluation Metrics

Datasets. We carry out experiments on two widely-used HOI detection benchmarks: HICO-Det [2] and V-COCO [8]. We follow the standard evaluation scheme. HICO-Det consists of 47,776 Creative Common images from Flickr (38,118 for training and 9,658 for test) with more than 150K human-object pairs. It contains the same 80 object categories as MS-COCO [21] and 117 action categories. The objects and actions form 600 classes of HOI triplets. V-COCO is derived from MS-COCO dataset, which consists of 5,400 images in the trainval subset and 4,946 images in the test subset. It has 29 action categories (25 HOIs and 4 body motions) and 80 object categories. For both datasets, one person can interact with multiple objects in different ways at the same time.

Evaluation Metrics. Following the standard evaluation [2], we use the mean average precision (mAP) as the evaluation metric. For one positively predicted HOI triplet $\langle \text{human}, \text{object}, \text{action} \rangle$, it needs to contain accurate human and object locations (box IoU with reference to GT box is greater than 0.5) and correct object and action categories. Specifically, for HICO-Det, besides the full set of 600 HOI classes, we also report the mAP over a rare set of 138 HOI classes that have less than 10 training instances and a non-rare set of the other 462 HOI classes. For V-COCO, we report the role mAP for two scenarios: scenario 1 includes the cases even without any objects (for the four action categories of body motions), and scenario 2 ignores these cases.

4.2 Implementation Details

We implement three variant architectures of CDN: CDN-S, CDN-B, and CDN-L, where ‘S’, ‘B’, and ‘L’ denote small, base, and large, respectively. For CDN-S and CDN-B, we adopt ResNet-50 with a 6-layer transformer encoder as the visual feature extractor. For the cascade decoders, CDN-S is equipped with both 3-layer transformers, while CDN-B has a 6-layer transformer for each decoder. CDN-L only replaces the ResNet-50 with ResNet-101 in CDN-B. The reduced dimension size D_c

is set to 256. The number of queries N_d is set to 64 for HICO-Det and 100 for V-COCO since the average number of positives for variant human-object pairs per image of HICO-Det is smaller than V-COCO. The human and object box FFNs have 3 linear layers with ReLU, while the object and action category FFNs have one linear layer. The code is provided in supplemental material.

During training, we initialize the network with the parameters of DETR [1] trained with the MS-COCO dataset. We set the weight coefficients λ_b , λ_{GIoU} , λ_p , λ_o and λ_a to 2.5, 1, 1, 1 and 1, respectively, which are exactly same with QPIC [28]. We optimize the network by AdamW [23] with the weight decay 10^{-4} . We first train the whole model for 90 epochs with a learning rate of 10^{-4} decreased by 10 times at the 60th epoch. Then, during the decoupling training process, we fine-tune the cascade disentangling decoders together with the box, object, and action FFNs for 10 epochs with a learning rate of 10^{-5} . We use both object and action dynamic re-weighting for HICO-Det and only action dynamic re-weighting for V-COCO. The re-weighting parameter p is set to 0.7 for both object and action. The length L_Q of training sample queue Q for both object and action is set to $2 \times N_s$, where N_s is the sample number of the training set. All experiments are conducted on the 8 Tesla V100 GPUs and CUDA10.2, with a batch size of 16.

For validation, we select 100 detection results with the highest scores and then adopt PNMS to further filter results. The threshold, α , and β of PNMS are set to 0.7, 1, and 0.5, respectively.

4.3 Experiment Analysis of Two-stage and One-stage Methods

In this part, we introduce a detailed experimental analysis of conventional two-stage and one-stage methods and our proposed CDN from the following three aspects.

Human-object Pair Generation. We first explore the quality of the human-object pairs generation between two-stage and one-stage methods. To attain this, we conduct a detailed experiment based on a representative two-stage method iCAN [6]. We first implement a PyTorch version iCAN as the baseline model, denoted as iCAN*, which only applies human and object appearance with a COCO-pretrained Faster-RCNN detector [25]. For a fair comparison, we first fine-tune DETR on HICO-Det for 100 epochs only with the instance detection annotation based on COCO-pretrained weights. Then we combine the detected human and object bounding-boxes, whose confidences are greater than a threshold, one by one to generate human-object pairs denoted as iCAN[†] in Table 1. We train our CDN only with HO-PD for 100 epochs and get the human-object pairs from the output directly. Then, we graft the human-object pairs to the baseline model to extract box features and utilize the same interaction classifier in the second stage of iCAN*. In this way, we degrade the number of pairs from $M \times N$ to K' , which means time complexity is reduced from $\mathcal{O}(M \times N)$ to $\mathcal{O}(K')$. Primarily, HO-PD significantly promotes mAP from 15.37 to 24.05, as shown in Table 1. This indicates that one-stage methods are much superior in human-object pair generation.

Strategy	Full	Rare	Non-Rare
iCAN*	14.16	12.26	14.73
iCAN [†]	15.37	13.23	16.01
HO-PD+iCAN*	24.05	18.32	25.76
QPIC [28]	29.07	21.85	31.23
CDN-S base	30.96	27.02	32.14

Table 1: **Analysis of Two-stage and One-stage Methods.** * denotes our implemented PyTorch version iCAN [6] baseline model. [†] denotes replacing instance detection boxes given by a HICO-Det fine-tuned DETR detector to extract box features. ‘HO-PD+iCAN*’ denotes replacing original one-by-one generated human-object pairs with our HO-PD generated. ‘CDN-S base’ denotes CDN-S w/o re-weighting and PNMS strategies.

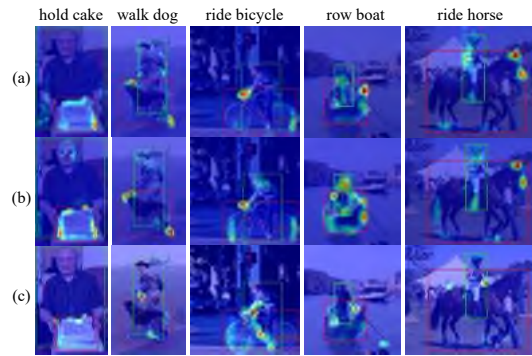


Figure 3: **Visualization of Feature Maps for Queries.** Visual attended features for query with top-1 score extracted from the last layer of the decoder of (a) QPIC, (b) HO-PD in CDN, and (c) interaction decoder in CDN. Zoom in for details.

Interaction Classification. We aim to study the interaction classification between conventional multi-task one-stage methods and our disentangled one-stage detector. We can regard QPIC [28] as a

Method	Detector	Backbone	Extra	Default			Know Object		
				Full	Rare	Non-Rare	Full	Rare	Non-Rare
Two-stage Method:									
InteractNet [7]	COCO	ResNet-50-FPN	$\times$	9.94	7.16	10.77	-	-	-
GPNN [24]	COCO	Res-DCN-152	$\times$	13.11	9.34	14.23	-	-	-
iCAN [6]	COCO	ResNet-50	$\times$	14.84	10.45	16.15	16.26	11.33	17.73
No-Frills [9]	COCO	ResNet-152	P	17.18	12.17	18.68	-	-	-
PMFNet [30]	COCO	ResNet-50-FPN	P	17.46	15.65	18.00	20.34	17.47	21.20
CHGNet [31]	COCO	ResNet-50	$\times$	17.57	16.85	17.78	21.00	20.74	21.08
DRG [5]	COCO	ResNet-50-FPN	T	19.26	17.74	19.71	23.40	21.75	23.89
VCL [12]	COCO	ResNet-50	$\times$	19.43	16.55	20.29	22.00	19.09	22.87
IP-Net [32]	COCO	Hourglass-104	$\times$	19.56	12.79	21.58	22.05	15.77	23.92
VSGNet [29]	COCO	ResNet-152	$\times$	19.80	16.05	20.91	-	-	-
FCMNet [22]	COCO	ResNet-50	$\times$	20.41	17.34	21.56	22.04	18.97	23.12
ACP [15]	COCO	ResNet-152	T	20.59	15.92	21.98	-	-	-
PD-Net [35]	COCO	ResNet-152-FPN	T	20.81	15.90	22.28	24.78	18.88	26.54
DJ-RN [16]	COCO	ResNet-50	P	21.34	18.53	22.18	23.69	20.64	24.60
IDN [17]	COCO	ResNet-50	$\times$	23.36	22.47	23.63	26.43	25.01	26.85
One-stage Method:									
UnionDet [13]	COCO	ResNet-50-FPN	$\times$	17.58	11.72	19.33	19.76	14.68	21.27
DIRV [4]	COCO	EfficientDet-d3	$\times$	21.78	16.38	23.39	25.52	20.84	26.92
PPDM-Hourglass [20]	HICO-DET	Hourglass-104	$\times$	21.94	13.97	24.32	24.81	17.09	27.12
HOI-Trans [39]	HICO-DET	ResNet-50	$\times$	23.46	16.91	25.41	26.15	19.24	28.22
GG-Net [37]	HICO-DET	Hourglass-104	$\times$	23.47	16.48	25.60	27.36	20.23	29.48
ATL [11]	HICO-DET	ResNet-50	$\times$	23.81	17.43	25.72	27.38	22.09	28.96
HOTR [14]	HICO-DET	ResNet-50	$\times$	25.10	17.34	27.42	-	-	-
AS-Net [3]	HICO-DET	ResNet-50	$\times$	28.87	24.25	30.25	31.74	27.07	33.14
QPIC [28]	HICO-DET	ResNet-50	$\times$	29.07	21.85	31.23	31.68	24.14	33.93
CDN-S	HICO-DET	ResNet-50	$\times$	31.44	27.39	32.64	34.09	29.63	35.42
CDN-B	HICO-DET	ResNet-50	$\times$	31.78	27.55	33.05	34.53	29.73	35.96
CDN-L	HICO-DET	ResNet-101	$\times$	32.07	27.19	33.53	34.79	29.48	36.38

Table 2: **Performance comparison on the HICO-Det test set.** The ‘P’, ‘T’ represent human pose information and the language feature, respectively.

multi-task version of our CDN. Table 1 shows that our ‘CDN-S base’ (w/o re-weighting and PNMS strategies) has achieved mAP 30.96 with 6.50% relative mAP gain compared to QPIC. Especially, our ‘CDN-S base’ significantly outperforms QPIC for rare classes with a 23.66% improvement. The performance of rare classes can partly reflect the accuracy of interaction classification.

Feature Learning. This part discusses the differences in feature learning between the conventional one-stage method, QPIC, and our CDN from a qualitative view. As shown in Figure 3, we visualized the feature maps extracted from the last layer of the decoder of QPIC, the HO-PD, and the interaction decoder in our CDN. We can see that HO-PD and QPIC attend very similar regions, *e.g.*, the boundaries of humans and objects and the human-object contact areas, which are beneficial for locating the interactive human-object pairs. However, the interaction decoder concentrates on human-pose and the regions that contribute to understanding human actions. As for the specific case, for example, for ‘hold cake’, HO-PD in CDN attends to the boundaries of the cake while the interaction decoder in CDN concentrates on the interaction context, *i.e.*, the human’s hands holding the cake. Thus, it shows that CDN disentangles the human-object detection and interaction classification. For ‘ride horse’, HO-PD in CDN emphasizes the overall feature of the human and the horse, and the highlight parts are the edges of the human and horse. For the interaction decoder in CDN, the highlighted part emphasizes the interaction context, *i.e.*, the human carries the rope when riding a horse. Finally, QPIC somehow combines the two highlights, but both are not obvious.

4.4 Comparison to State-of-the-Art

We conduct experiments on HICO-Det and V-COCO benchmarks to verify the effectiveness of our proposed methods. For HICO-Det dataset as shown in Table 2, comparing to the previous state-of-the-art two-stage method FCMNet [22] with ResNet-50 as backbone, our CDN-B significantly promotes mAP from 20.41 to 31.78, with a relative gain of 55.71%. Even compared with PD-Net [36] which adopts extra language feature and DJ-RN [16] which utilizes extra human pose features, CDN-B achieves 52.71% and 48.92% relative mAP gains, respectively. When comparing to the one-stage method AS-Net [3] and QPIC [28] which also adopt transformer-based detector architecture, CDN-B attains 10.08% and 9.32% point relative mAP gains, respectively. Table 3 shows the comparisons on V-COCO dataset. CDN-B achieves 62.29 AP_{role} on Scenario 1 and 64.42 AP_{role} on Scenario 2, which significantly outperform previous state-of-the-art method QPIC with relative 5.94% and 5.61%

Method	Extra	AP _{role} ^{S1}	AP _{role} ^{S2}
Two-stage Method:			
InteractNet [7]	✗	40.0	-
GPNN [24]	✗	44.0	-
iCAN [6]	✗	45.3	52.4
TIN [18]	✗	47.8	54.2
VCL [12]	✗	48.3	-
DRG [5]	T	51.0	-
IP-Net [32]	✗	51.0	-
VSGNet [29]	✗	51.8	57.0
PMFNet [30]	P	52.0	-
PD-Net [35]	T	52.6	-
CHGNet [31]	✗	52.7	-
FCMNet [22]	✗	53.1	-
ACP [15]	T	53.23	-
IDN [17]	✗	53.3	60.3
One-stage Method:			
UnionDet [13]	✗	47.5	56.2
HOI-Trans [39]	✗	52.9	-
AS-Net [3]	✗	53.9	-
GG-Net [37]	✗	54.7	-
HOTR [14]	✗	55.2	64.4
DIRV [4]	✗	56.1	-
QPIC [28]	✗	58.8	61.0
CDN-S	✗	61.68	63.77
CDN-B	✗	62.29	64.42
CDN-L	✗	63.91	65.89

Table 3: **Performance comparison on the V-COCO test set.** The ‘P’, ‘T’ represent the human pose information and the language feature, respectively.

Strategy	Full	Rare	Non-Rare
QPIC [28]	29.07	21.85	31.23
base	31.06	26.68	32.36
+ re-weighting	31.38	27.36	32.58
+ PNMS	31.78	27.55	33.05

(a) **Strategies:** Analysis of improvements by various training strategies.

Strategy	L_Q	p	Full	Rare	Non-Rare
base	-	-	31.06	26.68	32.36
decouple	-	-	30.90	26.09	32.33
static	-	0.7	31.25	27.12	32.49
dynamic	$2 \times N_s$	0.8	31.33	27.45	32.49
dynamic	$1 \times N_s$	0.7	31.34	27.48	32.49
dynamic	$2 \times N_s$	0.7	31.38	27.36	32.58

(b) **Dynamic re-weighting:** Analysis of decouple training with dynamic re-weighted losses, *i.e.*, different queue length L_Q , coefficient p and dynamic or static.

α	β	thres	Full	Rare	Non-Rare
-	-	-	31.38	27.36	32.58
1	1	0.8	31.66	27.46	32.91
1	1	0.7	31.75	27.50	33.03
1	0.7	0.7	31.77	27.54	33.03
1	0.5	0.8	31.75	27.51	33.02
1	0.5	0.7	31.78	27.55	33.05

(c) **PNMS:** The effects of different settings of PNMS coefficients, *i.e.*, α , β , and *thres* denotes threshold.

Table 4: **Ablation studies of our proposed method on the HICO-Det test set.** We carry out all experiments based on the base model (CDN-B).

gains, respectively. As for efficiency analysis, CDN-S has almost the same number of parameters and flops compared to QPIC, but CDN-S achieves mAP 31.44 on HICO-Det, 8.15% higher than QPIC.

4.5 Ablation Study

In this subsection, we analyse the effectiveness of the proposed strategies and components in detail. All experiments are eventuated on the HICO-Det dataset. The performance of each strategy is evaluated in Table 4a. The five hyper-parameters of the training loss in 7 follow QPIC [28]. The ablation study of the two hyper-parameters in the re-weighting is shown in Table 4b, and that of the three hyper-parameters in the PNMS is shown in Table 4c. We carry out all experiments based on the model CDN-B with ResNet-50 as backbone.

Strategies. As shown Table 4a, our pure model without any additional post-processing operation, namely base model, achieves mAP 31.06, promoting 1.99 compared with QPIC [28]. Especially, the base model significantly promotes mAP for rare classes from 21.85 to 26.68 compared to QPIC. It indicates the superiority of the architecture of disentangling human-object detection and interaction classification. The re-weighted training further promotes mAP to 31.38, with a gain of 0.32, and the gain mainly lies in rare classes. Finally, the PNMS further improves mAP to 31.78.

Dynamic re-weighting. In this part, we conduct experiments to evaluate the components in the dynamic re-weighted training strategy based on the base model as shown in Table 4b. If we only decouple training without re-weighting, the model achieves mAP 30.90, which is lower than the base model. Therefore, it shows that the performance gain does not come from a longer training process. Adding static weights w_{static} to losses promotes mAP to 31.25. The dynamic re-weighting method improves the re-weighting effect since it captures the real-time weight of each class for each real-time sample during training. Thus it can sufficiently dig information from every single sample to achieve the best overall performance. Our method obtains best result mAP 31.38 when $L_Q = 2 \times N_s$ and $p = 0.7$.

PNMS. On the basis of the model after re-weighted training, we compare the variance by different parameter settings of the PNMS strategy, which is shown in Table 4c. We fix the human box balance

factor α to 1. Then we tune the object box balance factor β and the threshold of the $PIoU$ to filter pair-wise boxes. We achieve best performance mAP 31.78 when $\beta = 0.5$ and $thres = 0.7$. The fact that β is smaller than α , indicates that the overall performance is more sensitive to human boxes compared with object boxes in our framework.

5 Related Work

Two-stage Methods. Most previous HOI detectors are with a two-stage paradigm [6, 2, 18, 5]. Firstly, a fine-tuned detector [25, 10] is applied to detect the instances. Secondly, generating the human-object pairs by matching detected human and object one by one, and then feeding them into an interaction classifier. To improve the interaction classification, some extra features were applied, such as human pose [27, 19, 9], human parts [38, 30, 16], and language features [33, 5, 22, 15]. Besides, some two-stage methods [24, 29, 31, 34, 38] applied graph neural networks to model the interactions.

One-stage Methods. One-stage methods detect HOI triplets directly [20, 32, 13, 39, 3, 28, 14]. In detail, [20, 32] proposed a point-based interaction detection method which performs inference at each interaction key point. [13] proposed an anchor-based method to predict the interactions for each human-object union box. Recently, set-based detection approach has been proposed to handle HOI detection as a set prediction problem [39, 3, 28]. Specifically, [39, 28] designed a transformer encoder-decoder architecture to directly predict HOI detection results in an end-to-end manner, while [3] utilized parallel instance and interaction decoder branches to adaptively aggregate the HOI triplets.

6 Conclusion

In this paper, we explore the essential pros and cons of two-stage and one-stage HOI detection in detail. We propose a novel one-stage framework with disentangling human-object detection and interaction classification in a cascade manner. Our CDN can keep the advantage of one-stage methods, directly and accurately locating the interactive human-object pairs, and bring the benefit of two-stage methods, disentangling detection and interaction classification. Our novel paradigm has outperformed previous methods by margins. However, we only implement a specific version to mine the benefits of two-stage and one-stage methods. In the future, we plan to apply our idea with more general one-stage methods and introduce more advantages of two-stage methods into the one-stage framework.

Potential Negative Societal Impacts

Similar to many other AI technologies, our proposed CDN itself is harmless. However, someone might utilize it for military purposes or apply it to any other malicious human activities detection, which might negatively impact society. Therefore, we encourage well-intentioned consideration before adopting our technique.

Acknowledgments and Disclosure of Funding

This research is partly supported by National Natural Science Foundation of China (Grant 61876177), Beijing Natural Science Foundation (4202034), Fundamental Research Funds for the Central Universities, Zhejiang Lab (No. 2019KD0AB04).

References

- [1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3, 4, 7
- [2] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *WACV*, 2018. 1, 6, 10
- [3] Mingfei Chen, Yue Liao, Si Liu, Zhiyuan Chen, Fei Wang, and Chen Qian. Reformulating hoi detection as adaptive set prediction. In *CVPR*, 2021. 1, 2, 8, 9, 10
- [4] Hao-Shu Fang, Yichen Xie, Dian Shao, and Cewu Lu. Dirv: Dense interaction region voting for end-to-end human-object interaction detection. In *AAAI*, 2021. 8, 9
- [5] Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. Drg: Dual relation graph for human-object interaction detection. In *ECCV*, 2020. 1, 8, 9, 10

- [6] Chen Gao, Yuliang Zou, and Jia-Bin Huang. ican: Instance-centric attention network for human-object interaction detection. In *BMVC*, 2018. 1, 7, 8, 9, 10
- [7] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *CVPR*, 2018. 1, 8, 9
- [8] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015. 1, 6
- [9] Tanmay Gupta, Alexander Schwing, and Derek Hoiem. No-frills human-object interaction detection: Factorization, layout encodings, and training techniques. In *ICCV*, 2019. 8, 10
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 10
- [11] Zhi Hou, Yu Baosheng, Yu Qiao, Xiaojiang Peng, and Dacheng Tao. Affordance transfer learning for human-object interaction detection. In *CVPR*, 2021. 8
- [12] Zhi Hou, Xiaojiang Peng, Yu Qiao, and Dacheng Tao. Visual compositional learning for human-object interaction detection. In *ECCV*, 2020. 8, 9
- [13] Bumsoo Kim, Taeho Choi, Jaewoo Kang, and Hyunwoo J. Kim. Uniondet: Union-level detector towards real-time human-object interaction detection. In *ECCV*, 2020. 2, 8, 9, 10
- [14] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J. Kim. Hotr: End-to-end human-object interaction detection with transformers. In *CVPR*, 2021. 2, 8, 9, 10
- [15] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *ECCV*, 2020. 8, 9, 10
- [16] Yong-Lu Li, Xinpeng Liu, Han Lu, Shiyi Wang, Junqi Liu, Jiefeng Li, and Cewu Lu. Detailed 2d-3d joint representation for human-object interaction. In *CVPR*, 2020. 8, 10
- [17] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Yizhuo Li, and Cewu Lu. Hoi analysis: Integrating and decomposing human-object interaction. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 8, 9
- [18] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yan-Feng Wang, and Cewu Lu. Transferable interactiveness prior for human-object interaction detection. In *CVPR*, 2019. 1, 9, 10
- [19] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. In *CVPR*, 2019. 10
- [20] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In *CVPR*, 2020. 1, 2, 3, 8, 10
- [21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6
- [22] Yang Liu, Qingchao Chen, and Andrew Zisserman. Amplifying key cues for human-object-interaction detection. In *ECCV*, 2020. 8, 9, 10
- [23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 7
- [24] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018. 8, 9, 10
- [25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 7, 10
- [26] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 6
- [27] Liyue Shen, Serena Yeung, Judy Hoffman, Greg Mori, and Li Fei-Fei. Scaling human-object interaction recognition through zero-shot learning. In *WACV*, 2018. 10
- [28] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In *CVPR*, 2021. 2, 5, 7, 8, 9, 10

- [29] Oytun Ulutan, A S M Iftekhar, and B. S. Manjunath. Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions. In *CVPR*, 2020. 8, 9, 10
- [30] Bo Wan, Desen Zhou, Yongfei Liu, Rongjie Li, and Xuming He. Pose-aware multi-level feature network for human object interaction detection. In *ICCV*, 2019. 8, 9, 10
- [31] Hai Wang, Wei-shi Zheng, and Ling Yingbiao. Contextual heterogeneous graph network for human-object interaction detection. In *ECCV*, 2020. 8, 9, 10
- [32] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *CVPR*, 2020. 8, 9, 10
- [33] Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and Mohan S. Kankanhalli. Learning to detect human-object interactions with knowledge. In *CVPR*, 2019. 10
- [34] Dongming Yang and Yuexian Zou. A graph-based interactive reasoning for human-object interaction detection. In *IJCAI*, 2020. 10
- [35] Xubin Zhong, Changxing Ding, Xian Qu, and Dacheng Tao. Polysemy deciphering network for human-object interaction detection. In *ECCV*, 2020. 8, 9
- [36] Xubin Zhong, Changxing Ding, Xian Qu, and Dacheng Tao. Polysemy deciphering network for robust human-object interaction detection. *IJCV*, 2021. 8
- [37] Xubin Zhong, Xian Qu, Changxing Ding, and Dacheng Tao. Glance and gaze: Inferring action-aware points for one-stage human-object interaction detection. In *CVPR*, 2021. 8, 9
- [38] Penghao Zhou and Mingmin Chi. Relation parsing neural network for human-object interaction detection. In *ICCV*, 2019. 10
- [39] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, and Jian Sun. End-to-end human object interaction detection with hoi transformer. In *CVPR*, 2021. 2, 3, 5, 8, 9, 10

Understanding Chinese Video and Language via Contrastive Multimodal Pre-Training

Chenyi Lei^{1,2}, Shixian Luo², Yong Liu³, Wanggui He², Jiamang Wang², Guoxin Wang²,
Haihong Tang², Chunyan Miao⁴, Houqiang Li¹

¹University of Science and Technology of China, Hefei, Anhui, China

²Alibaba Group, Hangzhou, Zhejiang, China

³Alibaba-NTU Singapore Joint Research Institute, Singapore

⁴School of Computer Science and Engineering, Nanyang Technological University, Singapore

leichy@mail.ustc.edu.cn, {huanyi.lsx, wanggui.hwg, jiamang.wang}@alibaba-inc.com

{stephenliu, ascymiao}@ntu.edu.sg, {xiaogong.wgx, piaoxue}@taobao.com, lihq@ustc.edu.cn

ABSTRACT

The pre-trained neural models have recently achieved impressive performance in understanding multimodal content. However, it is still very challenging to pre-train neural models for video and language understanding, especially for Chinese video-language data, due to the following reasons. Firstly, existing video-language pre-training algorithms mainly focus on the co-occurrence of words and video frames, but ignore other valuable semantic and structure information of video-language content, *e.g.*, sequential order and spatiotemporal relationships. Secondly, there exist conflicts between video sentence alignment and other proxy tasks. Thirdly, there is a lack of large-scale and high-quality Chinese video-language datasets (*e.g.*, including 10 million unique videos), which are the fundamental success conditions for pre-training techniques.

In this work, we propose a novel video-language understanding framework named **VICTOR**, which stands for **Video-language understanding via Contrastive mulTimOdal pRe-training**. Besides general proxy tasks such as masked language modeling, VICTOR constructs several novel proxy tasks under the contrastive learning paradigm, making the model be more robust and able to capture more complex multimodal semantic and structural relationships from different perspectives. VICTOR is trained on a large-scale Chinese video-language dataset, including over 10 million complete videos with corresponding high-quality textual descriptions. We apply the pre-trained VICTOR model to a series of downstream applications and demonstrate its superior performance, comparing against the state-of-the-art pre-training methods such as VideoBERT and UniVL.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Computer vision**.

Corresponding authors: Yong Liu, Houqiang Li.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '21, October 20–24, 2021, Virtual Event, China

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8651-7/21/10...\$15.00

<https://doi.org/10.1145/3474085.3475431>

KEYWORDS

Multimodal Pre-training; Video and Language Analysis; Contrastive Learning

ACM Reference Format:

Chenyi Lei^{1,2}, Shixian Luo², Yong Liu³, Wanggui He², Jiamang Wang², Guoxin Wang², Haihong Tang², Chunyan Miao⁴, Houqiang Li¹. 2021. Understanding Chinese Video and Language via Contrastive Multimodal Pre-Training. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3474085.3475431>

1 INTRODUCTION

Inspired by the revolutionary successes of pre-training strategies in natural language processing [8, 28, 35, 43, 51], there has been a steady momentum of breakthroughs in vision-language pre-training techniques [5, 12, 13, 21, 23, 24, 29, 34, 36, 39, 45]. However, most of existing vision-language pre-training methods are focusing on image-language pre-training tasks. The pre-training techniques developed for video-language pre-training are still relatively limited [20, 22, 30, 40, 41]. VideoBERT [41] and CBT [40] are the pioneering video-language pre-training methods that are developed based on the Transformer [47] structure and three widely used proxy tasks, *i.e.*, masked language modeling, masked frame modeling, and video and sentence alignment. To support both discriminant and generation tasks, UniVL [30] further introduces an encoder-decoder pre-training framework with masked sentence generation task. Figure 1 shows an example of these proxy tasks.

Compared with image-language pre-training, video-language pre-training has the following main challenges. First, videos are dynamic and have more complex spatiotemporal and sequential relationships than static 2D images. The spatiotemporal relationship reflects the correlation of video frames, and the sequential relationship verifies the semantic coherence, including visual frames and textual segmentations. Most existing video-language pre-training models are directly extended from image-language pre-training methods without exploiting these unique video and language relationships. Second, there are conflicts between the typical implementation of Video Sentence Alignment (VSA) [41] and other proxy tasks. The reason is that mismatched video-text pairs would force the model to fuse irrelevant multimodal signals to understand video and language with other proxy tasks [24, 39]. Third, existing video-language datasets are mainly crawled from websites, leading to

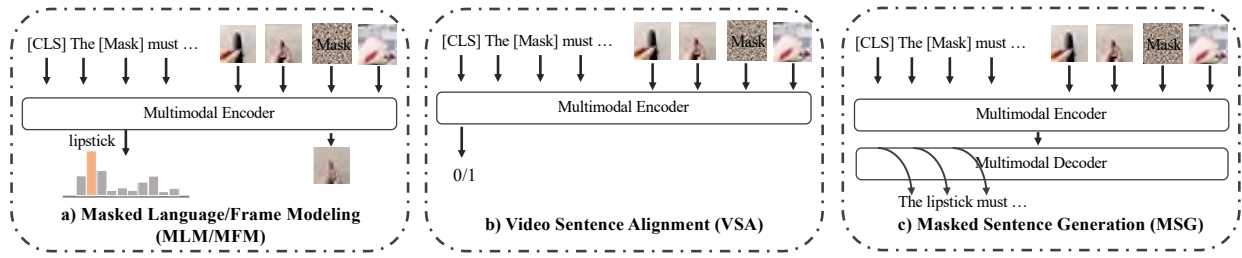


Figure 1: General proxy tasks that are widely used in existing video-language pre-training methods.

lots of semantic irrelevant video-text pairs in the dataset. There is a lack of large-scale and high-quality datasets for video-language pre-training, especially Chinese video-language datasets.

To tackle these aforementioned challenges, we firstly propose a novel video-language pre-training framework, namely **VICTOR** (i.e., Video-language understanding via Contrastive mulTimOdal pRe-training). Specifically, VICTOR is based on the encoder-decoder framework, whose backbone is a stacking of Transformers [47]. The encoder part can be used to learn robust multimodal representations of video and text for downstream discriminant tasks such as classifications and cross-modal retrieval. The decoder part emphasizes the ability of generations, which can be applied to downstream generation tasks such as video caption and summarization. The proxy tasks of VICTOR fall into two categories: 1) *reconstructive proxy tasks*, and 2) *contrastive proxy tasks*.

- The reconstructive proxy tasks of VICTOR include two general tasks, i.e., Masked Language Modeling (MLM) and Masked Sentence Generation (MSG), which have been used in previous work [30]. In addition, we also propose the following two novel reconstructive proxy tasks from the sequential structure perspective, i.e., Masked Frame Order Modeling (MFOM) and Masked Sentence Order Modeling (MSOM). The objective is to learn the semantic sequential order information from both video frames and sentence segmentations.
- The contrastive proxy tasks include three novel tasks, i.e., dual Video Sentence Alignment (dual-VSA), intra-Masked Frame Modeling (intra-MFM), and inter-Masked Frame Modeling (inter-MFM). We extend typical VSA [41] to dual-VSA to avoid forcing the model to fuse inaccurate multimodal signals. The intra-MFM and inter-MFM proxy tasks aim to exploit the spatiotemporal supervision signals in the video. Particularly, intra-MFM focuses on the spatial standpoint within a video, and inter-MFM emphasizes the spatiotemporal standpoint among different videos. In this work, we formulate all the contrastive proxy tasks under the contrastive learning paradigm [2, 14]. The basic principle is to make positive/negative query-key pairs similar/dissimilar by the ranking loss or Noise Contrastive Estimation (NCE) loss. Inspired by MoCo [14], we additionally build a dynamic memory queue to track keys across mini-batches for better training.

Note that the data pre-processing for MLM, MFOM, and MSOM can be regarded as data augmentations in this work, which makes comparative learning tasks more robust.

To empower the model with rich Chinese knowledge, it is essential to perform pre-training on large-scale and high-quality

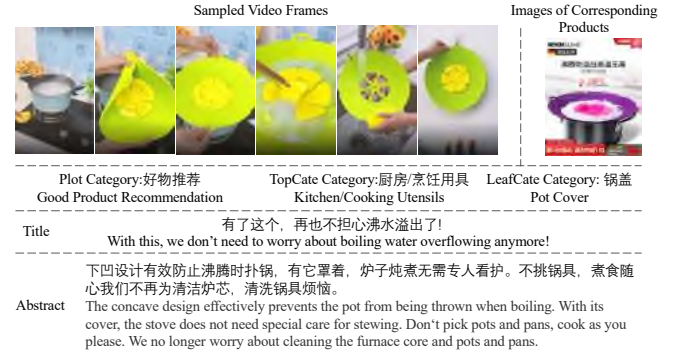


Figure 2: An example of the videos in ALIVOL-10M dataset. Besides the high-resolution video frames and human-created title, the video also contains a long-text abstract and a related e-commerce product with images. Each video has three types of categories: plot category, coarse-grained product category, and fine-grained product category (respectively denoted by Plot, TopCate, and LeafCate).

Chinese video-language datasets. In this work, we also collected A Large-scale ChInese VideO-Language dataset, named **ALIVOL-10M**, from Taobao. In ALIVOL-10M, there are more than 10 million short-videos, instead of video clips cut from complete videos as in existing datasets such as HowTo100M [31]. The total duration of videos in ALIVOL-10M is about 99 thousand hours. To the best of our knowledge, ALIVOL-10M is the largest Chinese video-language dataset. Moreover, all the videos in ALIVOL-10M are carefully produced by professional content creators, and each video is accompanied by a manually written high-quality text description. Furthermore, each video in ALIVOL-10M also contains various manually annotated information, e.g., plot categories, e-commerce categories, and semantic tags. These additional knowledge provides several new credible downstream applications for evaluation. Figure 2 illustrates an example of the videos in ALIVOL-10M.

The main contributions made in this work are as follows:

- We propose a novel encoder-decoder framework, named VICTOR, for video-language pre-training with novel proxy tasks, including MFOM, MSOM, dual-VSA, intra-MFM, and inter-MFM. VICTOR is able to capture the sequential structure and spatiotemporal information from the video and language data.
- We formulate the proposed contrastive proxy tasks, i.e., dual-VSA, intra-MFM, and inter-MFM, under the contrastive learning paradigm with an advanced momentum updating strategy.

- We build to-date the largest Chinese video-language dataset for video-language pre-training, *i.e.*, ALIVOL-10M, which includes over 10 million high-quality e-commerce videos and corresponding human-created text descriptions.
- Experimental results on ALIVOL-10M demonstrate that VICTOR achieves significant improvements over State-Of-The-Art (SOTA) video-language pre-training methods, *i.e.*, VideoBERT and UniVL, on four different downstream applications. We also conduct an in-depth analysis about the impacts of different proxy tasks on the pre-training performance.

2 RELATED WORK

The large scale pre-training techniques have been widely studied in recent years. In the language domain, Transformer [47] based approaches pre-trained with the masked language modeling objective have revolutionized Natural Language Processing (NLP). BERT [8], RoBERTa [28], XLNET [51], T5 [35], and ERNIE [43] are the representative methods which have been used across different NLP tasks. Riding on the successes of BERT-style pre-training models in NLP, numerous visual-linguistic pre-training models have been developed. For example, ViLBERT [29] and LXMERT [45] are the pioneering image-language pre-training methods, which adopt two separate Transformers for encoding image and text independently and a third Transformer for later multimodal fusion. Compared with these two-stream frameworks, VL-BERT [39], Unicoder-VL [21], OSCAR [23], and UNITER [5] use one shared BERT model to build a single-stream architecture. These models mainly employ MLM and image-text matching as pre-training tasks. Moreover, the recent work DALL-E [36] and TDEN [24] also make impressive progress on generation tasks, *e.g.*, text-to-image generation and image captions.

For video-language pre-training, VideoBERT [41] and CBT [40] are the pioneers to explore the capability of pre-training models, which are single-stream and two-stream architectures respectively. For better pre-training on generation tasks, UniVL [30] proposes an encoder-decoder framework based on the two-stream architecture. HERO [22] further encodes multimodal inputs in hierarchical transformers with two new proxy tasks, *i.e.*, video-subtitle matching and frame order modeling. ClipBERT [20] proposes to utilize a sparse sampling strategy to model videos from raw pixels. CUPID [53] studies the domain gap between video-language pre-training and fine-tuning. [33] leverages a generative model to push video and text together. These pre-trained models have been applied in various downstream applications, *e.g.*, action recognition [15], video caption [6, 32, 55], video retrieval [9, 10, 50], and video question answering [19, 25]. The video-language data has more complex semantic structures and relationships than the image-language data, leading video-language pre-training to be a much more challenging research problem. However, existing video-language pre-training works lack detail exploring these unique challenges, such as the spatiotemporal and sequential relationships.

Large-scale and high-quality datasets are the fundamental success condition for large-scale pre-training techniques. For image-language pre-training, there exist numerous large-scale and high-quality datasets with over 100 thousand unique images, including English datasets [17, 27] and Chinese datasets [13, 26] (10 millions level). However, there are only three datasets including over

100 thousand unique videos for video-language pre-training, *i.e.*, TGIF [25], AutoGIF [32], and HowTo100M [31]. All TGIF, AutoGIF, and HowTo100M are English datasets and crawled from open websites. There exist lots of videos with irrelevant textual descriptions. Moreover, to the best of our knowledge, there are no large-scale Chinese video-language datasets. Thus, there is an urgent need to pre-train, evaluate, and release models on a large-scale and high-quality Chinese video-language dataset.

In contrast to the boom in the image-language pre-training area, pre-training for video and language is still in infancy. Additionally, there is no existing work performing large-scale video-language pre-training on Chinese data. In this work, we aim to propel video-language pre-training in two dimensions: 1) developing a better video-language pre-training model, especially for capturing spatiotemporal and sequential relationships of the video and language data; 2) evaluating and releasing models pre-trained on the large-scale and high-quality Chinese video-language dataset.

3 METHOD

In this section, we first introduce an overview of the VICTOR model and then describe the designed proxy tasks from two perspectives. After that, we will introduce the model optimization strategy.

3.1 Model Overview

The video-language pre-training problem can be defined as: *given a video and the corresponding textual descriptions as a pair data $\{V, T\}$, pre-training model is to learn joint video and text representation network $f(V, T)$ with the self-supervision approach.* The representation network $f(V, T)$ is expected to maximize the mutual information and preserve the complementary information of multi-modalities. Thus, it can be further fine-tuned for various downstream applications, *e.g.*, video classifications and captions. As discussed in recent work in NLP [35] and multimodal pre-training [30], the encoder-decoder framework usually achieves the best performance for both discriminant and generation tasks. Motivated by these observations, we also adopt the encoder-decoder architecture as pre-training framework to learn $f(V, T)$ in this work. The detailed architecture of VICTOR is illustrated as follows.

Input Embedder. For each textual description T , we follow BERT [8] and tokenize the input sentence into WordPieces [49], *i.e.*, $T = \{w_i | i \in [1, n]\}$, where n is the maximum number of tokens. Two special tokens [CLS] and [SEP] are used to indicate the beginning and ending of the input sentence, respectively. The final representation for each token is obtained via summing up its token embedding and position embedding, followed by a layer normalization (LN) layer. For each video V , we first sample a frame sequence, *i.e.*, $V = \{f_i | i \in [1, m]\}$, where m is the maximum number of frames. Then, we utilize Inception-V4 [44] pre-trained on ImageNet [7] to extract visual features for each frame (denoted as $F_{ori} \in \mathcal{R}^{m \times 1536}$). Next, visual features are fed through a fully connected (FC) layer to be projected into the same lower-dimensional space as token embeddings. The final embedding of a frame is then obtained by summing up the outputs of FC layer and the position embeddings, followed by an LN layer. Finally, we have token embeddings $W \in \mathcal{R}^{n \times d}$ for the given text T , and frame embeddings $F \in \mathcal{R}^{m \times d}$ for the given video V , where d is the hidden size.

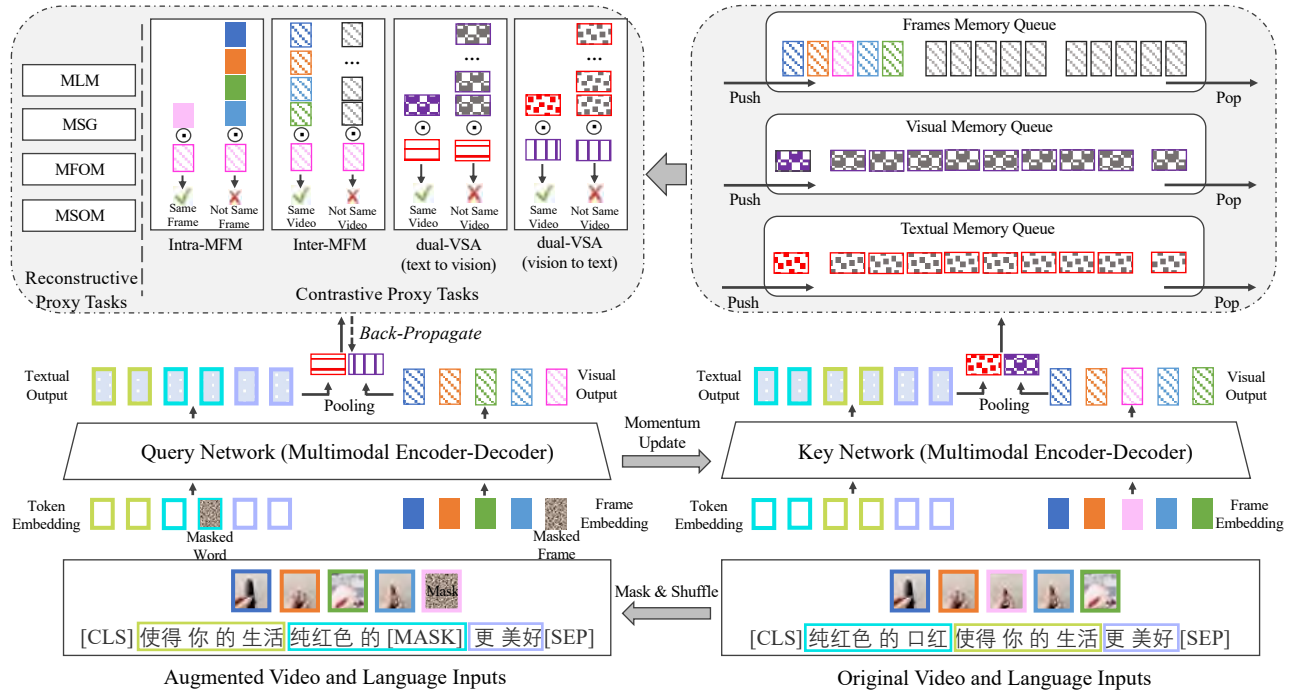


Figure 3: The overall framework of the proposed VICTOR model. The squares of the same color in the figure represent the associated representations. Please view this figure in color.

Multimodal Transformer Encoder and Decoder. The multimodal encoder is devised to fully capture the intra-modal relationships and inter-modal interaction. Given the token embedding $W \in \mathcal{R}^{n \times d}$ and frame embeddings $F \in \mathcal{R}^{m \times d}$, we first concatenate them as $[W, F] \in \mathcal{R}^{(n+m) \times d}$. Then, we further feed $[W, F]$ into a stack of transformer-based encoder blocks [47], leading to the contextual multimodal representations of each word/frame token $[W_E, F_E] \in \mathcal{R}^{(n+m) \times d}$. To empower the pre-training model to have generation capability, we also involve a multimodal decoder that learns to auto-regressively reconstruct the input sentence word-by-word conditioned on the input video and masked text with unidirectional attention mechanism [24, 30]. Here, we implement the multimodal decoder by stacking transformer-based decoder blocks. In particular, each transformer-based decoder block first collects contextual information from all the “past” word tokens via unidirectional attention, and then performs cross-attention over all frame tokens for next word prediction.

Figure 3 illustrates the overall framework of the VICTOR model. Specifically, VICTOR consists of a Query Network and a Key Network. Both of them are based on the multimodal encoder-decoder framework mentioned above. The Query Network produces multimodal embeddings with masking and shuffling strategies on both textual and visual inputs, while the Key Network generates multimodal embeddings with the original inputs. After the Query/Key Network, we have multimodal frame embeddings $F_E^q/F_E^k \in \mathcal{R}^{m \times d}$ and multimodal text embeddings $W_E^q/W_E^k \in \mathcal{R}^{n \times d}$. Moreover, we perform max pooling for embeddings to get the general representation for visual and text, i.e., $R_v \in \mathcal{R}^d$ and $R_t \in \mathcal{R}^d$. Motivated

by MoCo [14], the output embeddings of Key Network (i.e., F_E^k , R_v , and R_t) are pushed into three dynamic memory queues, which continuously generate negative samples for calculating losses for different contrastive proxy tasks. The Query Network is updated by all pre-training losses, and the Key Network is updated conditioned on Query Network weight similar to MoCo. Note all masking and shuffling strategies for Query Network inputs can be regarded as data augmentations for our proposed momentum contrastive pre-training strategy.

3.2 Reconstructive Proxy Tasks

Masked Language Modeling (MLM). Motivated by the breakthrough success of the MLM task in NLP, we leverage a multimodal version of MLM for pre-training. Following BERT, we randomly mask 15% text tokens with the special token [MASK] in the sentence, denoted by W_m . The goal is to predict these masked tokens based on the video frames F and other tokens W_{-m} , by minimizing the following negative log-likelihood:

$$\mathcal{L}_{MLM}(\theta) = -\mathbb{E}_{(W, F) \sim D} \log P_{\theta}(W_m | W_{-m}, F), \quad (1)$$

where θ denotes the trainable parameters, and each pair (W, F) is sampled from the whole training set D .

Masked Sentence Order Modeling (MSOF). The objective of MSOF is to learn the relationships of text tokens from sequential perspective. During the pre-training stage, 15% randomly selected sentences are randomly split into 3 segments, and then all the segments are shuffled by a random permuted order. We let the pre-trained model reorganize these permuted segments, modeled as a

k -class classification problem, where $k = 3!$. Empirically, the sentence reordering task can enable the model to learn the sequential semantic structure of language. Specifically, after the multimodal encoder, the embedding of [CLS] is transformed through an FC layer, followed by a softmax layer, to produce a probability. The final objective is to minimize the following negative log-likelihood:

$$\mathcal{L}_{MSOM}(\theta) = -\mathbb{E}_{(W,F) \sim D} \log P_{\theta}(y|W, F), \quad (2)$$

where y is the ground truth of shuffled segments.

Masked Frame Order Modeling (MFOM). We also construct a reordering task to learn the sequential semantic structure of the video. For each frame sequence, we randomly select to shuffle 15% frames that are denoted by $F_s = \{f_{s,i}, i \in [1, N_s]\}$, where N_s is the number of shuffled frames. Our objective is to reconstruct the original order of these shuffled frames. The embedding of each shuffled frame is transformed through an FC layer, followed by a softmax layer, to predict the original order index. This is a m -class classification problem, where m is the maximum length of frame sequence. We minimize the following negative log-likelihood:

$$\mathcal{L}_{MFOM}(\theta) = -\mathbb{E}_{(W,F) \sim D} \sum_{i=1}^{N_s} \log P_{\theta}(y_{s,i}|f_{s,i}, W, F), \quad (3)$$

where $y_{s,i}$ is the original order index of the shuffled frame $f_{s,i}$. There exist two main differences between this MFOM proxy task and the frame order modeling task used in the HERO model [22]. One is that HERO only shuffles the unmasked frames. The other is that HERO fuses the frames and corresponding subtitles before frame shuffling. These two differences make the frame order modeling task of HERO much easier. Thus, it is more difficult for HERO to learn the sequential semantic structure of the video.

Masked Sentence Generation (MSG). Inspired by previous work [30, 35], we introduce the MSG proxy task to enable the multimodal decoder to auto-regressively reconstruct the input sentence word-by-word. The loss function of the MSG proxy task is defined as follows:

$$\mathcal{L}_{MSG}(\theta) = -\mathbb{E}_{(W,F) \sim D} \log P_{\theta}(\hat{w}_i|\hat{w}_{<i}, W, F), \quad (4)$$

where W is the sentence with 15% masked tokens (c.f. MLM) when pre-training. To endow the model with better generation capacity, we further randomly select 15% sentences with full masked tokens.

3.3 Contrastive Proxy Tasks

Besides the input processing of reconstructive proxy tasks, such as masking tokens, we further randomly mask the embeddings of 15% input frames with zeros for the Query Network (Figure 3). We utilize InfoNCE [46] to define the following contrastive loss:

$$\mathcal{L}_{NCE}(q, k^+, \mathcal{K}^-) = -\log \frac{\exp(q^{\top} k^+ / \tau)}{\exp(q^{\top} k^+ / \tau) + \sum_{i=1}^K \exp(q^{\top} k_i^+ / \tau)}, \quad (5)$$

where q and k^+ are a positive pair for query and key, $\mathcal{K}^-$ denotes the negative key set, and τ is the temperature hyper-parameter. The motivation of such formulation is to train the model to maximize the mutual information of the query and positive key.

Intra-Masked Frame Modeling (intra-MFM). The objective of this proxy task is to guide the model to exploit the inherently spatial variation across frames within a video. In particular, we

first feed all encoded embeddings of masked frames from Query Network into an FC layer to project them into vectors with the same dimension as the pre-extracted input frame embeddings (e.g., Inception-V4 in our work). We denote the projected embeddings by $\{g_{\theta}(F_{E,i}), i \in [1, N_m]\}$, where $g_{\theta}(\cdot)$ denotes the FC layer, $F_{E,i}$ is the encoded embedding of the i -th masked frame, and N_m is the number of masked frames. Then, our goal is to classify the correct pre-extracted input frame embedding $F_{ori,i}$ comparing to the rest input frame embeddings $\mathcal{K}_{ori}^-$ for the given $g_{\theta}(F_{E,i})$. Accordingly, we define the following contrastive loss for this task:

$$\mathcal{L}_{intra-MFM}(\theta) = \mathbb{E}_{(W,F) \sim D} \frac{1}{N_m} \sum_{i=1}^{N_m} \mathcal{L}_{NCE}(g_{\theta}(F_{E,i}), F_{ori,i}, \mathcal{K}_{ori}^-). \quad (6)$$

Inter-Masked Frame Modeling (inter-MFM). In this proxy task, the pre-trained model is learnt to differentiate the augmented frames in the same video or from different videos. From the spatiotemporal perspective, the augmented query frame should be similar to all the other frames in the same video. For each encoded frame $\{F_{E,i}^q | i \in [1, m]\}$ from the Query Network, we randomly select one encoded frame $\{F_{E,j}^k | j \in [1, m]\}$ of the same video from the Key Network as the positive key. The frame set $\mathcal{K}_F^-$ from other videos in the dynamic frame memory queue are taken as the negative keys. We define the following contrastive loss to measure the objective of the inter-MFM proxy task:

$$\mathcal{L}_{inter-MFM}(\theta) = \mathbb{E}_{(W,F) \sim D} \frac{1}{m} \sum_{i=1}^m \mathcal{L}_{NCE}(F_{E,i}^q, F_{E,j}^k, \mathcal{K}_F^-), \quad (7)$$

where m is the maximum number of sampled frames of a video.

Such design goes beyond the traditional self-supervision in one video with data augmentation, and fetches more positive frame patches within the same video as supervision for contrastive learning. It sheds new light on objects with temporal evolution (e.g., new views/poses of objects). This task elegantly takes the advantage of spatiotemporal structure within videos, and thus strengthens the unsupervised visual and textual feature learning for multimodal video understanding.

Dual Video and Sentence Alignment (dual-VSA). We also involve VSA task to strengthen global representations for video and language understanding. Nevertheless, the typical implementation of VSA (e.g., VideoBERT [8]), i.e., taking the mismatched video and text as inputs and utilizing the output embedding of [CLS] to perform the binary classification, would hurt the performance of downstream discriminant applications. One possible reason is that the introduced mismatched vision-text pairs through shared multimodal encoder would hamper the pre-training of other proxy tasks [24, 39]. In this work, we only take matched video-text pairs as inputs, and utilize the representation of frames/text to retrieve the representation of corresponding text/frames.

In particular, taking the representation of frames R_v as query, the representation of text R_t from the same video is the positive key, and the text representation set $\mathcal{K}_t^-$ encoded by the Key Network from other videos in the dynamic textual memory queue are taken as the negative keys. Then, we can define the following loss function:

$$\mathcal{L}_{dual-VSA-v2t}(\theta) = \mathbb{E}_{(W,F) \sim D} \mathcal{L}_{NCE}(R_v, R_t, \mathcal{K}_t^-). \quad (8)$$

Similarly, we have the dual loss function as follows:

$$\mathcal{L}_{dual-VSA-t2v}(\theta) = \mathbb{E}_{(W,F) \sim D} \mathcal{L}_{NCE}(R_t, R_v, \mathcal{K}_v^-), \quad (9)$$

where $\mathcal{K}_v^-$ is the visual representation set encoded by the Key Network from other videos in the dynamic visual memory queue.

3.4 Model Optimization

The overall training objective of the proposed pre-training model combines the loss functions of all proxy tasks as follows:

$$\begin{aligned} \mathcal{L} = & \mathcal{L}_{MLM} + \mathcal{L}_{MSOM} + \mathcal{L}_{MFOM} + \mathcal{L}_{MSG} + \\ & \mathcal{L}_{intra-MFM} + \mathcal{L}_{inter-MFM} + \\ & \mathcal{L}_{dual-VSA-v2t} + \mathcal{L}_{dual-VSA-t2v}. \end{aligned} \quad (10)$$

In the proposed model, the Query Network is directly optimized by using SGD algorithm to minimize $\mathcal{L}$. The weights of Key Network are accordingly updated conditioned on the Query Network weights via a momentum updating strategy:

$$\theta_k = \alpha \cdot \theta_k + (1 - \alpha) \cdot \theta_q, \quad (11)$$

where α stands for momentum value, θ_k and θ_q are the parameters of Key Network and Query Network, respectively. This form of contrastive learning can achieve superior performance, due to the large memory queue and the small feature drift during the training process.

4 EXPERIMENTS

4.1 Pre-Training Dataset

All existing datasets for video-language pre-training are in English. To empower the pre-trained model with Chinese knowledge and evaluate the effectiveness of the pre-training mechanism on Chinese data, we build a video-language pre-training dataset, namely ALIVOL-10M, from Taobao, one of the world's largest e-commerce platforms. Existing video-language datasets are usually collected from open websites [6, 25, 25, 31, 50], where the data are freely created by users without restriction. This usually leads to the following problems in existing datasets. First, there exist many cases where videos are irrelevant to text descriptions. Second, the textual sentences may contain many meaningless tokens, thus may not be fluent in natural language. Third, video and text from open websites may also contain illegal content and user privacy information in some cases. Instead, all of the video and text data in ALIVOL-10M are created by professional creators, *e.g.*, merchants, advertising companies, opinion leaders, and professional reviewers, following the standards of Taobao. For example, the videos and texts must be semantically related and must not contain illegal content. As the videos are provided by Taobao, videos in ALIVOL-10M mainly describe merchandises in different display forms, *e.g.*, merchandise evaluation and situation comedy. This ensures that most videos and texts describe the observed visual content. In addition, we also group the videos of ALIVOL-10M using three different category systems, *i.e.*, 13 categories of plot, 153 categories of the coarse-grained product, and 11,529 categories of fine-grained product. We compare some characteristics of ALIVOL-10M with existing widely used datasets in Table 1. To the best of our knowledge, ALIVOL-10M is the first large-scale and high-quality dataset for Chinese video-language pre-training.

Table 1: Statistics of different video-language datasets.

Dataset	# Videos	# Sentences	Duration	Source	Language
YouCook2 [54]	2K	15.4K	176h	Youtube	English
Charades [38]	10K	27.8K	82h	Home	English
M-VAD [1]	92	55.9K	84h	Movies	English
MPII-MD [37]	94	68.3K	41h	Movies	English
MSR-VTT [50]	7.2K	200K	40h	Youtube	English
TGIF [25]	102.1K	126K	103h	Tumblr	English
AutoGIF [32]	163.2K	165K	-	Web	English
HowTo100M [31]	1.2M	136M	134.5Kh	Youtube	English
VATEX [48]	41.3K	825K	-	Web	Multilingual
ALIVOL-10M	10.3M	11M	98.8Kh	E-Commerce	Chinese

Table 2: Results of cross-modal video retrieval tasks. R@N denotes the recall of top-N predictions.

Methods	Text-based			Image-based		
	R@1	R@10	R@20	R@1	R@10	R@20
VideoBERT	0.1940	0.7132	0.8251	0.4700	0.8373	0.8672
UniVL	0.1677	0.4220	0.7221	0.4636	0.8351	0.8901
VICTOR	0.2425	0.7222	0.8656	0.4770	0.8954	0.9441

4.2 Implementation Details

We utilize 30,000 token vocabulary and pre-trained text token embeddings of BERT-base-Chinese model¹ to initialize the text token embeddings of VICTOR. Inception-V4 pre-trained on ImageNet is applied to extract frame features. The maximal input tokens of text are 128, and the maximal number of video frames is 32. For each video, we sample one frame every second. There are 12 and 2 Transformer blocks for multimodal encoder and decoder, respectively. Each Transformer block has 12 heads and 768 hidden size. The size of memory queue in contrastive learning is 65,586. The momentum value α is set to 0.999, and the temperature value τ is set to 0.7. We adopt Adam [16] as the optimizer with an initial learning rate 10^{-4} , and set the batch size to 128 for each GPU. We pre-train the proposed VICTOR model on 128 NVIDIA Tesla V100 GPUs for 30 epochs, which takes about three days. VideoBERT [41] and UniVL [30] are chosen as baseline methods. They are also pre-trained for 30 epochs on ALIVOL-10M. Moreover, we randomly select 200 thousand videos that are not used in the pre-training stage and use them as the data for downstream tasks. For all downstream tasks, we take 180 thousand videos to fine-tune the pre-trained model and 20 thousand videos to evaluate the performance of different baselines in downstream tasks.

4.3 Results on Downstream Tasks

4.3.1 Cross-Modal Video Retrieval. In this work, we focus on two cross-modal video retrieval tasks: text-based and image-based video retrieval. The text-based task aims to retrieve relevant videos from candidates based on the given text query. For the image-based task, we use an image as the given query to retrieve videos. In particular, for ALIVOL-10M, we utilize the video title and corresponding product image to retrieve relevant videos. For each video, we randomly select 100 videos as the negative ones. Therefore, we obtain 18 million text-video/image-video pairs for fine-tuning and 2 million

¹<https://github.com/google-research/bert>

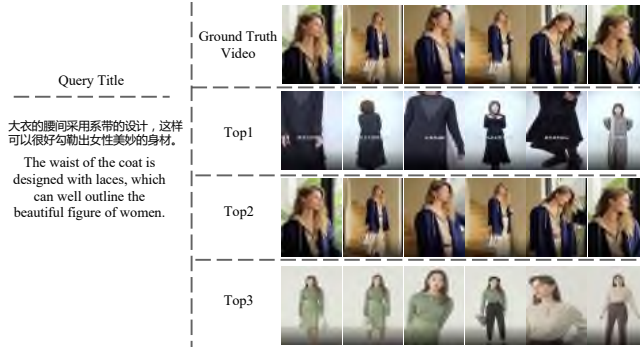


Figure 4: An example result of the text-based video retrieval task, which is fine-tuned based on the pre-trained VICTOR model.

pairs for evaluating. We introduce the following two fine-tuning strategies for these video retrieval tasks, respectively.

- **Text-based Retrieval.** The inputs of the text-based video retrieval task are the title and candidate video frames. Similar to the dual-VSA task in pre-training, we also utilize max pooling to obtain the overall representations of text and vision, which are utilized to calculate the matching score based on Cosine distance.
- **Image-based Retrieval.** Besides the query image, the inputs of the image-based video retrieval task include the title and frames of the candidate video. During fine-tuning, we replace the first frame of the candidate video as the query image. The matching score is calculated as the same as in text-based video retrieval.

From Table 2, we can note that VICTOR achieves the best performance comparing to baseline methods with a large extent for two cross-modal video retrieval tasks. In particular, for the image-based video retrieval task, VICTOR achieves the significant absolute improvement over the second-best method (*i.e.*, VideoBERT) by 5.81%, in terms of R@10. There are two potential reasons. Firstly, this may be caused by the advantage of the dual-VSA proxy task for learning cross-modal semantic representations. Secondly, the inter-MFM task captures the spatiotemporal information of the video to distinguish the query image and candidate frames. Figure 4 and Figure 5 show case studies for text-based and image-based video retrieval, respectively.

4.3.2 Multi-level Video Classification. For each video in ALIVOL-10M, it has three different types of category labels from different levels, *i.e.*, plot category (Plot), coarse-grained product category (TopCate), and fine-grained product category (LeafCate). There are significant differences in the classification systems of these three types of categories. The pre-trained model must understand videos from different aspects for making an accurate classification. In particular, we utilize an additional FC layer after the output embedding of [CLS] to fine-tune pre-trained models for the plot classifier. We add two different FC layers after the output embedding of [CLS] to fine-tune pre-trained models simultaneously for the other two multi-level product categories. The motivation is that these two multi-level product categories are interrelated. Moreover, we also test baseline methods under the Translate-Test setting [11, 42, 56], where baseline methods are pre-trained on English datasets and

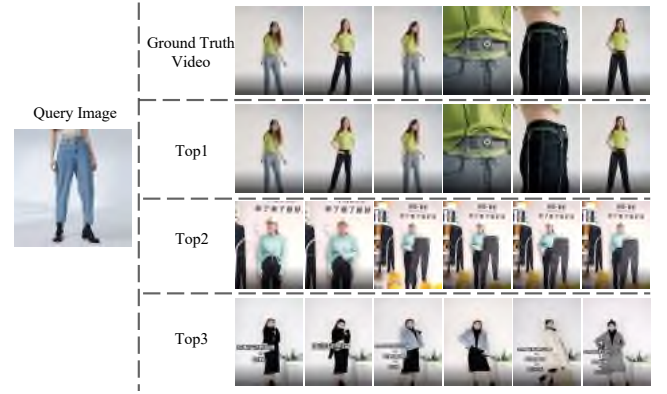


Figure 5: An example result of the image-based video retrieval task, which is fine-tuned based on the pre-trained VICTOR model.

Table 3: Results of three video classification tasks and the content-based video recommendation task. “-” denotes baseline methods under the Translate-Test setting.

Methods	Classification			Recommendation	
	Plot Acc.(%)	TopCate Acc.(%)	LeafCate Acc.(%)	AUC	HR@3
VideoBERT	79.46	70.14	55.43	0.7056	0.4577
UniVL	75.66	67.13	54.13	0.7027	0.4523
VideoBERT-T	76.79	68.11	54.77	-	-
UniVL-T	71.22	66.20	52.39	-	-
ClipBERT-T	61.87	65.83	51.15	-	-
VICTOR	81.09	70.50	56.71	0.7104	0.4655

Table 4: Results of the multimodal video caption task.

Methods	BLEU-1	BLEU-4	Meteor	Rouge-L	Cider
VideoBERT	23.98	9.90	11.36	20.01	0.66
UniVL	24.52	11.33	11.89	21.19	0.78
VICTOR	28.38	14.75	13.82	23.89	1.10

evaluated on translated ALIVOL-10M dataset. Specifically, UniVL-T² and ClipBERT-T³ employ the official pre-trained models on English datasets. VideoBERT-T is pre-trained on sampled HowTo100M by ourselves. The video classification results are summarized in Table 3. For all three category classification tasks, VICTOR significantly outperforms other baselines. These results demonstrate the effectiveness of the proposed pre-training strategy for different video classification tasks.

4.3.3 Content-Based Video Recommendation. The content-based video recommendation task aims to recommend videos to a user based on the multimodal information of videos viewed by her/him [3, 4, 18]. Thus, the multimodal representations of videos are crucial for the content-based video recommendation task. We utilize users’ video viewing logs on the e-commerce platform to evaluate the

²<https://github.com/microsoft/UniVL>

³<https://github.com/jayleicn/ClipBERT>

Table 5: Evaluation on proxy tasks and pre-training datasets. All methods are based on our proposed framework. 10M videos in the table denote ALIVOL-10M, and 100K videos denote a randomly sampled subset of ALIVOL-10M that contains 100 thousand videos. The best and second-best results are in bold and underlined, respectively.

Method Number	Proxy Tasks	#Pre-Training Videos	Video Retrieval		Video Classification		
			Text-based R@10	Image-Based R@10	Plot Acc.(%)	TopCate Acc.(%)	LeafCate Acc.(%)
M1	MLM + MSG	10M	0.6798	0.8632	78.21	70.02	55.15
M2	M1 + MSOM + MFOM	10M	0.6733	0.8689	80.42	70.04	56.61
M3	M2 + intra/inter-MFM	10M	0.7015	0.8713	<u>80.91</u>	70.55	57.15
M4	M3 + VSA	10M	<u>0.7108</u>	<u>0.8764</u>	79.80	70.2	55.15
M5	M3 + dual-VSA	100K	0.6916	0.8641	79.17	68.08	55.10
M6	M3 + dual-VSA	10M	0.7222	0.8954	81.09	<u>70.50</u>	<u>56.71</u>

performance of this task. Specifically, we collect seven days of user video viewing logs as training data, and the user video viewing logs in the following day are used as test data. To this end, we have about 200 thousand users with 1 million logs for training and about 30 thousand users with 200 thousand logs for evaluation. The DIN [52] model, which is a well-known recommendation method, is selected as the backbone. The video representations are extracted by pre-trained models, which are the output embeddings of [CLS]. Area Under the receiver operating characteristic Curve (AUC) and Hit Ratio at rank K (denoted by HR@K) are selected as evaluation metrics. Table 3 summarizes the video recommendation results. As shown in Table 3, VICTOR achieves the best performance, in terms of both AUC and HR. This again demonstrates the multimodal representing capacity of our pre-trained model.

4.3.4 Multimodal Video Caption. The objective of multimodal video caption is to generate a sequence of descriptive sentences based on a given video and some textual hints. In this work, we choose the video title and frames as inputs to generate the video abstract (c.f. Figure 2). We use the beam search with setting beam size to 5, and adopt the following corpus-level generation evaluation metrics using the open-source tool⁴, including BLEU-3, BLEU-4, Meteor, Rouge-L, and Cider. Table 4 summarizes the video caption performance achieved by all methods. We can note that our pre-training model VICTOR outperforms UniVL, which is the SOTA pre-training model for generation task. These results verify the generation capacity of our proposed model.

4.4 In-depth Analysis

We also perform experiments analyzing the effectiveness of our model design, especially focusing on different combinations of the proxy tasks. Table 5 summarizes the ablation study results across different pre-training data scales, under different proxy tasks. Particularly, we use 10M to denote ALIVOL-10M, and 100K to denote a randomly sampled subset of ALIVOL-10M that contains 100 thousand videos. From Table 5, we can make the following observations. Firstly, M6 consistently outperforms M5. This indicates that the pre-training method benefits a lot from the larger pre-training dataset. It also verifies the contribution of training and releasing the pre-trained model on a large-scale Chinese video-language dataset. Secondly, comparing the performance of M1 and M2, we can note

that MSOM and MFOM significantly improve the accuracy of the Plot classification task. This indicates that sequential characteristics of sentence and video frames can benefit the tasks relying on sequential reasoning. Thirdly, the method M3, which considers Intra- and Inter-MFM, not only achieves the best results in TopCate and LeafCate classification, but also achieves significantly better video retrieval accuracy than M1 and M2. This demonstrates the importance of modeling spatiotemporal information of videos. Fourthly, although M4 improves the R@10 of video retrieval tasks, the performance of video classification tasks drop a lot. Instead, the method M6 achieves better results than M4 and comparable results with M3, in video classification tasks. Moreover, M6 achieves the best video retrieval performance. These observations verify the motivation of introducing dual-VSA.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel encoder-decoder framework, namely VICTOR (*i.e.*, Video-language understanding via Contrastive mulTimOdal pRe-training), for video-language pre-training with novel reconstructive proxy tasks and contrastive proxy tasks. To support Chinese video-language pre-training, we also build to-date the largest Chinese video-language dataset ALIVOL-10M, which includes more than 10 million videos and corresponding text descriptions. Experimental results on ALIVOL-10M demonstrate that the proposed VICTOR model effectively exploits the multimodal information of the video-language data to achieve better performance than STOA video-language pre-training methods in downstream tasks, including cross-modal video retrieval, multi-level video classification, content-based video recommendation, and multimodal video caption. For future work, we would like to pre-train the proposed VICTOR on large-scale English datasets, *e.g.*, HowTo100M [31], and study its performance in different downstream tasks.

6 ACKNOWLEDGMENTS

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore. This research is also supported, in part, by the National Natural Science Foundation of China under Grants 61836011 and 62021001.

⁴<https://github.com/Maluuba/nlg-eval>

REFERENCES

- [1] Hugo Larochelle Aaron Courville Atousa Torabi, Christopher Pal. 2015. Using Descriptive Video Services to Create a Large Data Source for Video Annotation Research. *arXiv:1503.01070*
- [2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*. IEEE.
- [3] Xusong Chen, Chenyi Lei, Dong Liu, Guoxin Wang, Haihong Tang, Zheng-Jun Zha, and Houqiang Li. 2021. E-Commerce Storytelling Recommendation Using Attentional Domain-Transfer Network and Adversarial Pre-Training. In *Transactions on Multimedia*. IEEE.
- [4] Xusong Chen, Dong Liu, Chenyi Lei, Rui Li, Zheng-Jun Zha, and Zhiwei Xiong. 2019. BERT4SessRec: Content-Based Video Relevance Prediction with Bidirectional Encoder Representations from Transformer. In *MM*. ACM, 2597–2601.
- [5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-Text Representation Learning. In *ECCV*.
- [6] Pradipto Das, Chenliang Xu, Richard F. Doell, and Jason J. Corso. 2013. A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching. In *CVPR*. IEEE.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. IEEE.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805*
- [9] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal Transformer for Video Retrieval. In *ECCV*.
- [10] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. COOT: Cooperative Hierarchical Transformer for Video-Text Representation Learning. In *NIPS*.
- [11] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. In *ICML*.
- [12] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers. *arXiv:2004.00849*
- [13] Yuqi Huo, Manli Zhang, Guangzhen Liu, Haoyu Lu, Yizhao Gao, Guoxing Yang, Jingyuan Wen, Heng Zhang, Baogui Xu, Weihao Zheng, Zongzheng Xi, and et al. 2021. WenLan: Bridging Vision and Language by Large-Scale Multi-Modal Pre-Training. *arXiv:2103.06561*
- [14] Yuxin Wu Saining Xie Ross Girshick Kaiming He, Haoqi Fan. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*. IEEE.
- [15] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, and et al. 2017. The Kinetics Human Action Video Dataset. *arXiv:1705.06950*
- [16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *arXiv:1602.07332*
- [18] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. 2021. SEMI: A Sequential Multi-Modal Information Transfer Network for E-Commerce Micro-Video Recommendations. In *KDD*. ACM.
- [19] Chenyi Lei, Lei Wu, Dong Liu, Zhao Li, Guoxin Wang, Haihong Tang, and Houqiang Li. 2020. Multi-Question Learning for Visual Question Answering. In *AAAI*.
- [20] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is More: ClipBERT for Video-and-Language Learning via Sparse Sampling. In *CVPR*. IEEE.
- [21] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. 2020. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training. In *AAAI*.
- [22] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training. In *EMNLP*.
- [23] Xijun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *ECCV*.
- [24] Yehao Li, Yingwei Pan, Ting Yao, Jingwen Chen, and Tao Mei. 2021. Scheduled Sampling in Vision-Language Pretraining with Decoupled Encoder-Decoder Network. In *AAAI*.
- [25] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A New Dataset and Benchmark on Animated GIF Description. *arXiv:1604.02748*
- [26] Junyang Lin, Rui Men, An Yang, Chang Zhou, Ming Ding, Yichang Zhang, Peng Wang, and et al. 2021. M6: A Chinese Multimodal Pretrainer. *arXiv:2103.00823*
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312*
- [28] Yinhan Liu, Mylène Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*
- [29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NIPS*.
- [30] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation. *arXiv:1906.05743*
- [31] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*. IEEE.
- [32] Yingwei Pan, Yehao Li, Jianjie Luo, Jun Xu, Ting Yao, and Tao Mei. 2020. Auto-captions on GIF: A Large-scale Video-sentence Dataset for Vision-language Pre-training. *arXiv:2007.02375*
- [33] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metzger, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. 2021. Support-Set Bottlenecks for Video-Text Representation Learning. In *ICLR*.
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, and et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020*
- [35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv:1910.10683*
- [36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. *arXiv:2102.12092*
- [37] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A Dataset for Movie Description. *arXiv:1501.02530*
- [38] Gunnar A. Sigurdsson, Gu I Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in Homes Crowdsourcing Data Collection for Activity Understanding. In *ECCV*.
- [39] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *ICLR*.
- [40] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019. Learning Video Representations using Contrastive Bidirectional Transformer. *arXiv:2002.06353*
- [41] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *ICCV*. IEEE.
- [42] Siqi Sun, Yen-Chun Chen, Linjie Li, Shuohang Wang, Yuwei Fang, and Jingjing Liu. 2021. LightingDOT: Pre-training Visual-Semantic Embeddings for Real-Time Image-Text Retrieval. In *NAACL*.
- [43] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. In *AAAI*.
- [44] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261*
- [45] Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*.
- [46] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748*
- [47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NIPS*. ACM, 6000–6010.
- [48] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. VaTeX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research. In *ICCV*.
- [49] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, and et al. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144*
- [50] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*. IEEE.
- [51] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NIPS*.
- [52] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep Interest Evolution Network for Click-Through Rate Prediction. In *KDD*. ACM.
- [53] Luowei Zhou, Jingjing Liu, Yu Cheng, Zhe Gan, and Lei Zhang. 2021. CUPID: Adaptive Curation of Pre-training Data for Video-and-Language Representation

- Learning. arXiv:2104.00285
- [54] Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards Automatic Learning of Procedures from Web Instructional Videos. In *AAAI*.
- [55] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018. End-to-End Dense Video Captioning with Masked Transformer. In *CVPR*. IEEE.
- [56] Mingyang Zhou, Luowei Zhou, Shuohang Wang, Yu Cheng, Linjie Li, Zhou Yu, and Jingjing Liu. 2021. UC2: Universal Cross-lingual Cross-modal Vision-and-Language Pre-training. In *CVPR*.

Pre-training Graph Transformer with Multimodal Side Information for Recommendation

Yong Liu¹, Susen Yang³, Chenyi Lei³, Guoxin Wang³, Haihong Tang³, Juyong Zhang⁴
Aixin Sun², Chunyan Miao^{1,2}

¹Alibaba-NTU Singapore Joint Research Institute & Joint NTU-UBC LILY Research Centre, Nanyang Technological University, Singapore

²School of Computer Science and Engineering, Nanyang Technological University, Singapore

³Alibaba Group ⁴University of Science and Technology of China, China

{stephenliu, axsun, ascymiao}@ntu.edu.sg, susen.yss@alibaba-inc.com, leichy@mail.ustc.edu.cn
{xiaogong, wgx, piaoxue}@taobao.com, juyong@ustc.edu.cn

ABSTRACT

Side information of items, e.g., images and text description, has shown to be effective in contributing to accurate recommendations. Inspired by the recent success of pre-training models on natural language and images, we propose a pre-training strategy to learn item representations by considering both item side information and their relationships. We relate items by common user activities e.g., co-purchase, and construct a homogeneous item graph. This graph provides a unified view of item relations and their associated side information in multimodality. We develop a novel sampling algorithm named MCNSampling to select contextual neighbors for each item. The proposed Pre-trained Multimodal Graph Transformer (PMGT) learns item representations with two objectives: 1) graph structure reconstruction, and 2) masked node feature reconstruction. Experimental results on real datasets demonstrate that the proposed PMGT model effectively exploits the multimodality side information to achieve better accuracies in downstream tasks including item recommendation and click-through ratio prediction. In addition, we also report a case study of testing PMGT in an online setting with 600 thousand users.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Recommendation Systems, Pre-training Model, Graph Transformer

ACM Reference Format:

Yong Liu¹, Susen Yang³, Chenyi Lei³, Guoxin Wang³, Haihong Tang³, Juyong Zhang⁴, Aixin Sun², Chunyan Miao^{1,2}. 2021. Pre-training Graph Transformer with Multimodal Side Information for Recommendation. In *Proceedings of the 29th ACM Int'l Conference on Multimedia (MM '21)*, Oct.

Corresponding authors: Chenyi Lei, Chunyan Miao.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '21, October 20–24, 2021, Virtual Event, China.

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8651-7/21/10...\$15.00

<https://doi.org/10.1145/3474085.3475709>

20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 9 pages.
<https://doi.org/10.1145/3474085.3475709>

1 INTRODUCTION

In recent years, a good range of recommendation techniques have been proposed, from classic collaborative filtering techniques [30] to the recent deep learning models [39]. Besides the interactions between users and items, the multimodality side information of items has also been exploited and showed effectiveness in further improving recommendation accuracy [31]. Example side information of items include textual descriptions, images, and videos, as shown in Figure 1(a). Traditional methods exploit item side information by manual feature engineering [19], and then employ factorization machine [29] or gradient boosting machine [4] to predict users' preferences on items. These methods often require domain-specific knowledge, and are time-consuming. Deep learning-based methods leverage the strong representation learning ability of neural networks to exploit item side information, for learning the user and/or item representations. However, existing solutions only consider a specific type of side information of items for the dedicated recommendation applications [11, 20, 21]. The full multimodality side information of items are not fully exploited.

Inspired by the successes of unsupervised pre-training strategies designed for natural language processing [7] and graph data [14, 15, 22, 38], we propose to develop an unsupervised pre-training framework to fully exploit the multimodality side information for item representations learning. Illustrated in Figure 1, we construct an *item multimodal graph* to provide a unified view of items with their associated multimodality side information. In this item multimodal graph, each node is an item and the edges model their relationships (e.g., co-purchase or co-viewership, depending on the application domain). We then pre-train graph neural network (GNN) on this item multimodal graph to enable the GNN model to capture both item relationships and their multimodality side information.

The unified item multimodal graph well distinguishes our work from previous studies where side information of items and their relationships are studied separately. We argue that these two types of information complement each other in solving recommendation problems. We hence focus on effective pre-training on top of this item multimodal graph to benefit item recommendation. Moreover, we also show that our pre-training benefit other E-commerce applications like click-through ratio (CTR) prediction.

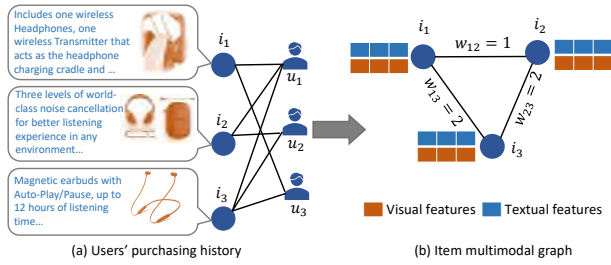


Figure 1: (a) Item side information and users’ purchasing history, and (b) Item multimodal graph built on co-purchase relationship. In this graph, each node denotes an item with its visual and textual features extracted from the image and text description respectively. An edge between two items is weighed by the number of co-purchases.

The contributions made in this paper are as follows. First, we propose a novel pre-training framework, namely Pre-trained Multimodal Graph Transformer (PMGT), to exploit items’ multimodality information through unsupervised learning. To the best of our knowledge, this is the first deep pre-training method developed to exploit the multimodality side information of items for recommender systems. Second, we decompose the learning objective of PMGT into two sub-objectives: (i) graph structure reconstruction, and (ii) masked node feature reconstruction. To handle large-scale graph data, we develop an algorithm, named Mini-batch Contextual Neighbors Sampling (*i.e.*, MCNSampling), for effective and scalable training. Moreover, we employ the attention mechanism to aggregate the item’s multimodality information, and a diversity-promoting Transformer framework to model the influences between an item and its contextual neighbors in the graph. Lastly, to demonstrate the effectiveness of the proposed PMGT model, we conduct extensive experiments on real datasets for different applications. The results on two downstream tasks, *i.e.*, item recommendation and CTR prediction, demonstrate that PMGT is more effective than existing graph-based pre-training methods in fully exploiting items’ multimodality information. To further show the effectiveness of PMGT, we report a case study of applying this model in an online E-commerce platform.

2 RELATED WORK

Pre-training methods have been widely applied in computer vision (CV) and natural language processing (NLP) tasks [7, 10]. It has been shown that pre-training is effective in boosting performances of various downstream applications. For example, a general pre-training paradigm for CV tasks is firstly training a model on the ImageNet dataset [6], and then fine-tuning the pre-trained model for a specific task. NLP is another domain where pre-training is usually adopted. The shallow pre-training methods, *e.g.*, word2vec [23] and GloVe [25], learn word representations based on the word co-occurrence patterns in a corpus of documents. Recently, significant progress has been made in developing deep pre-training models for contextual word representations. For example, ELMo [27] employs a bidirectional language model to learn high-quality deep context-dependent word representations. The BERT [7] and XLNET [37] models use attention mechanisms to learn the word representations.

Significant improvements are achieved when applying these pre-trained models on various NLP tasks.

On graph data, many embedding techniques have been developed in recent years [3]. Representative shallow graph embedding methods include TransE [2], DeepWalk [26], LINE [32], and Node2vec [8]. The recent popularity of GNNs motivates the development of pre-training strategies for GNN models. In general, these methods pre-train GNNs by solving the graph reconstruction problem. For example, Kipf *et al.* introduce the variational graph autoencoder (VGAE) framework for graph reconstruction [18]. Hamilton *et al.* propose a general inductive framework called GraphSAGE [9], which exploits node features to generate node embeddings by sampling and aggregating features from a node’s local neighborhood. Velickovic *et al.* propose the Deep Graph Infomax model [34] that aims to maximize the mutual information between the node representations and the representation of the graph. Recently, self-supervised learning [16] is employed to simultaneously pre-train GNNs at both node and graph levels, and an example is the self-supervised learning method for graph neural networks [14]. Similarly, the generative framework GPT-GNN [15] employs a self-supervised attributed graph generation task to pre-train GNNs, by effectively capturing both semantic and structural properties of the graph. In [28], a self-supervised graph neural network pre-training model is proposed to capture the universal network topological properties across multiple networks. In [38], the proposed Graph-BERT model employs attention mechanism to aggregate the neighborhood information of a target node in the graph.

3 THE PROPOSED PRE-TRAINING MODEL

Figure 2 shows the proposed PMGT framework. Observe that PMGT contains four main components: 1) contextual neighbors sampling, 2) node embedding initialization, 3) transformer-based encoder, and 4) graph reconstruction. Before we detail each component in this section, we provide the preliminary background.

In this work, we construct a homogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to provide a uniform view of items’ multimodality side information and their relationships. Here, $\mathcal{V}$ denotes the set of nodes (*i.e.*, items), and $\mathcal{E}$ denotes the set of edges between them.¹ Each node h has multiple types of side information. We denote the i -th modality feature of the node h by $\mathbf{x}_h^i$, and the number of modality by m .

In $\mathcal{G}$, we denote the one-hop neighbors of h by $\mathcal{N}_h$, and use ω_{ht} to denote the weight of the edge between two nodes h and t , where $\omega_{ht} > 0$. For a node h , we use C_h to denote its contextual neighbors selected by a sampling algorithm, *e.g.*, the MCNSampling algorithm. Given the item graph $\mathcal{G}$ and the contextual neighbors of each node, PMGT aims to obtain the node representations that can capture the multimodality information of nodes and the graph structure. Then, the learned node representations can be applied in downstream tasks directly or with adjustments such as fine-tuning.

3.1 Contextual Neighbors Sampling

For each node h , there exist some relevant nodes in the graph that may help enrich its representation. These relevant nodes are

¹In the context of recommendation, the relationships between items can be defined by their interactions with users, *e.g.*, co-purchase or co-click. More details about the construction of item graph are presented in Section 4.1.1.

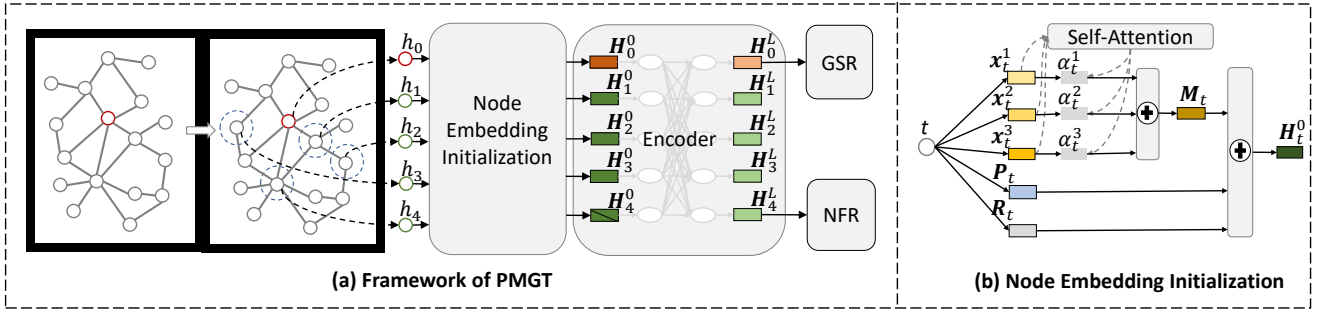


Figure 2: (a) An overview of the proposed PMGT framework. PMGT contains four components (illustrated from the left to the right): contextual neighbors sampling, node embedding initialization, transformer-based graph encoder, and graph reconstruction. GSR and NFR, in the last step, denote the graph structure reconstruction task and the masked node feature reconstruction task, respectively. (b) The node embedding is initialized by considering the node’s multimodal features, position-id embedding, and role-label embedding.

referred as the *contextual neighbors* of h . To efficiently select contextual neighbors for a batch of nodes during the training of PMGT, we develop a sampling algorithm named MCNSampling. MCNSampling iteratively samples a list of nodes for a target node h with a predefined sampling depth K . Let S_h^{k-1} denote the bag of nodes sampled at the $(k-1)$ -th step. For each node t in S_h^{k-1} , we randomly sample n_k nodes with replacement from t ’s one-hop neighbors N_t at the k -th step. The probability that a node $t' \in N_t$ being sampled is proportional to the weight $\omega_{tt'}$ of the edge between nodes t and t' . Note that a node may appear multiple times in S_h^{k-1} . In the MCNSampling algorithm, we treat all node instances in S_h^{k-1} as “different nodes” and perform the sampling procedure.

In our sampling algorithm, we select contextual neighbors by considering 1) the sampled frequency of a node, and 2) the number of sampling steps between the target node h and a sampled node in the sampling process. For every node $t \in \mathcal{V} \setminus h$, we empirically define its importance to the target node h at the k -th sampling step ($k \leq K$) as follows,

$$s_t^k = f_t^k \times (K - k + 1), \quad (1)$$

where f_t^k denotes the number of times t appearing in S_h^k . That is, a node t is considered more relevant to the target node h , if t is sampled more frequently and it has a smaller sampling steps to h . The final importance score of a node t to h is defined as follows,

$$s_t = \sum_{k=1}^K s_t^k. \quad (2)$$

Then, we sort all nodes in $\mathcal{V} \setminus h$ according to their importance scores in descending order, and choose the S top-ranked nodes as the sampled contextual neighbors of h . The details of the MCNSampling algorithm are summarized in Algorithm 1.

3.2 Node Embedding Initialization

After the neighborhood sampling, we concatenate the target node h and its ordered contextual neighbors C_h , denoted by $I_h = [h, h_1, h_2, \dots, h_S]$. h_j is the j -th node in C_h , and $1 \leq j \leq S$. For each node $t \in I_h$, we apply the attention mechanism to obtain its multimodal

representation M_t as follows,

$$\begin{aligned} X_t^i &= x_t^i W_M^i + b_M^i, 1 \leq i \leq m \\ X_t &= X_t^1 \oplus X_t^2 \oplus \dots \oplus X_t^m, \\ \alpha_t &= \text{softmax}[\tanh(X_t)W_s + b_s], M_t = \sum_i \alpha_t^i X_t^i, \end{aligned} \quad (3)$$

where $W_M^i \in \mathbb{R}^{d_i \times d_0}$ and $b_M^i \in \mathbb{R}^{1 \times d_0}$ denote weight matrix and bias term for the i -th modality, $W_s \in \mathbb{R}^{(md_0) \times m}$ and $b_s \in \mathbb{R}^{1 \times m}$ denote weight matrix and bias term for attention mechanism. $\oplus$ is the concatenation operation. That is, the multimodality side information of each item is concatenated to contribute to the comprehensive representation learning. α_t^i denotes the i -th element of α_t .

The position of a node in the input list I_h reflects its importance to the target node h . Thus, we argue that the order of nodes in I_h is important in learning node representations. The position-id embedding is used to identify the node order information of an input list,

$$P_t = \mathbf{P}\text{-Embedding}[p(t)], \quad (4)$$

where $p(t)$ denotes the position id of node t in I_h , $P_t \in \mathbb{R}^{1 \times d_0}$ denotes the position-based embedding for t . Our main objective is to obtain the representation of the target node h . Intuitively, the target node and its contextual neighbors should play different roles in the pre-training. To identify the role differences, we add the following role-based embedding to each node $t \in I_h$,

$$R_t = \mathbf{R}\text{-Embedding}[r(t)], \quad (5)$$

where $r(t)$ and $R_t \in \mathbb{R}^{1 \times d_0}$ denote the role label and role-based embedding of the node t , respectively. In practice, we set the role label of the target node as “Target” and the role labels of the contextual neighbors as “Context”. Based on the embeddings stated above, we aggregate them together to define the initial input embedding for a node $t \in I_h$ as,

$$H_t^0 = \text{Aggregate}(M_t, P_t, R_t). \quad (6)$$

In this work, we simply define the $\text{Aggregate}(\cdot)$ function as vector summation. The initial input embeddings for the nodes in the input list I_h can be stacked into a matrix $H^0 = [H_0^0, H_1^0; \dots; H_S^0] \in \mathbb{R}^{(S+1) \times d_0}$, where H_0^0 corresponds to the target node h .

Algorithm 1 MCNSampling Algorithm

Input: Graph $\mathcal{G}$, batch of nodes $\mathcal{B}$, sampling depth K , sampling size $\{n_k\}_{k=1}^K$, number of contextual neighbors S ;

Output: Sampled contextual neighbors C_B ;

```
1:  $C_B \leftarrow []$ ;
2: for  $h \in \mathcal{B}$  do
3:    $S_0 \leftarrow [h], S_1, S_2, \dots, S_K \leftarrow [], s_t \leftarrow 0 \forall t \in \mathcal{V} \setminus h$ ;
4:   for  $k = 1, 2, \dots, K$  do
5:     for  $t \in S_{k-1}$  do
6:       Sample  $n_k$  nodes with replacement from  $\mathcal{N}_t$  and append them to  $S_k$ ;
7:   end for
8:   Count the frequency  $f_t^k$  of each distinct node  $t$  in  $S_k$  and update its score  $s_t \leftarrow s_t + f_t^k * (K - k + 1)$ ;
9: end for
10: Sort the nodes in  $\mathcal{V} \setminus h$  according to the importance scores in descending order;
11: Choose  $S$  top-ranked nodes as the contextual neighbors  $C_h$  of  $h$  and append it to  $C_B$ ;
12: end for
13: return  $C_B$ 
```

3.3 Transformer-based Graph Encoder

We use the Transformer framework [33] to model the mutual influences between a node and its contextual neighbors. Given the node representations $\mathbf{H}^{\ell-1}$ at the $(\ell - 1)$ -th layer, the output at the ℓ -th layer of original Transformer model is defined as follows,

$$\mathbf{H}^\ell = \text{FFN} \left[\text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} \right) \mathbf{V} \right],$$
$$\mathbf{Q} = \mathbf{H}^{\ell-1} \mathbf{W}_Q^\ell, \mathbf{K} = \mathbf{H}^{\ell-1} \mathbf{W}_K^\ell, \mathbf{V} = \mathbf{H}^{\ell-1} \mathbf{W}_V^\ell, \quad (7)$$

where $\mathbf{W}_Q^\ell, \mathbf{W}_K^\ell, \mathbf{W}_V^\ell \in \mathbb{R}^{d_0 \times d_0}$ denote the weight matrices, $\text{FFN}(\cdot)$ is the feed forward network. Here, we omit the residual network in the formula for convenience.

For the target node h , there may exist some sampled nodes in C_h , whose representations are similar to the representation of h . Assume that all the sampled contextual neighbors are relevant to h . We hope the proposed model can capture the diversity of the sampled contextual neighbors, by concentrating on the nodes that are relevant but not very similar to the target node. To achieve this objective, we design a diversity-promoting attention mechanism and include it into the attention network of Transformer,

$$\mathbf{S} = \mathbf{H}^{\ell-1} \mathbf{W}_S^\ell,$$
$$\mathbf{U}_1 = \text{softmax} \left(\mathbf{E} - \frac{\mathbf{S}\mathbf{S}^\top}{\|\mathbf{S}\|_2 \|\mathbf{S}\|_2^\top} + \mathbf{I} \right), \quad (8)$$
$$\mathbf{U}_2 = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} \right),$$
$$\mathbf{H}^\ell = \text{FFN} \left[(\beta \mathbf{U}_1 + (1 - \beta) \mathbf{U}_2) \mathbf{V} \right], \quad (9)$$

where $\mathbf{W}_S^\ell \in \mathbb{R}^{d_0 \times d_0}$ is the weight matrix, $\mathbf{E} \in \mathbb{R}^{(S+1) \times (S+1)}$ is a matrix where all its elements are 1, $\|\mathbf{S}\|_2 \in \mathbb{R}^{(S+1) \times 1}$ denotes the ℓ_2 row norm of $\mathbf{S}$, and $\mathbf{I} \in \mathbb{R}^{(S+1) \times (S+1)}$ denotes the identity matrix.

In Eq. (8), “ $-$ ” denotes the element-wise division of two matrices. Note that the larger the similarity between two different nodes, the smaller the attention weight between them in $\mathbf{U}_1$. The objective of adding $\mathbf{I}$ in the definition of $\mathbf{U}_1$ is to include the node’s self information. β is a constant ($0 \leq \beta \leq 1$) balancing the contributions of the two attention weights. After obtaining the output $\mathbf{H}^L$ at the last layer of the encoder, we obtain $\mathbf{H}_0^L$ as the representation of target node h , denoted as $\mathbf{h}$ for simplicity. Then, $\mathbf{H}^L$ will be used in the following pre-training tasks.

3.4 Model Optimization

PMGT is pre-trained with the following two objectives: 1) graph structure reconstruction, and 2) masked node feature reconstruction. To ensure the learned node representations can capture the graph structure, we define the following loss function [9],

$$\mathcal{L}_{edge} = \frac{1}{|\mathcal{V}|} \sum_{h \in \mathcal{V}} \frac{1}{|\mathcal{N}_h|} \sum_{t \in \mathcal{N}_h} \left[-\log \left(\sigma \left(\frac{\mathbf{h}^\top \mathbf{t}}{\|\mathbf{h}\|_2 \|\mathbf{t}\|_2} \right) \right) \right. \\ \left. - Q \cdot \mathbb{E}_{t_n \sim P_n(t)} \log \left(\sigma \left(-\frac{\mathbf{h}^\top \mathbf{t}_n}{\|\mathbf{h}\|_2 \|\mathbf{t}_n\|_2} \right) \right) \right], \quad (10)$$

where $\sigma(\cdot)$ is the sigmoid function, P_n and Q denote the negative sampling distribution and the number of negative samples.

The node feature reconstruction task focuses on capturing the multimodal features in the learned node representations. Previous methods, e.g., GRAPH-BERT [38], design an attribute reconstruction task without masking operations. Thus, the models’ abilities in aggregating the features of different nodes may be limited. In this work, we design a masked node feature reconstruction task, which aims to reconstruct the features of masked nodes by other non-masked nodes in $\mathcal{I}_h$. As the representation of the target node h is needed to reconstruct the graph structure in Eq. (10), we do not apply the masking operation to the target node h . Following [7], we randomly choose 20% of nodes in the list $\mathcal{I}_h \setminus h$ for masking. If the node t is chosen, we replace t with: 1) the [Mask] node 80% of the time, (2) a random node 10% of the time, and (3) the unchanged node t 10% of the time. Then, the masked item list will be input to the model, and the output $\mathbf{H}^L$ will be used to reconstruct the multimodal features of the masked nodes. We set the input features of the [Mask] node to $\mathbf{0}$, and define the feature reconstruction loss as follows,

$$\mathcal{L}_{feature} = \frac{1}{|\mathcal{V}|} \sum_{h \in \mathcal{V}} \frac{1}{|\mathcal{M}_h|} \sum_{t \in \mathcal{M}_h} \sum_i^m \|\mathbf{H}_t^L \mathbf{W}_r^i - \mathbf{x}_t^i\|_2^2, \quad (11)$$

where $\mathcal{M}_h$ denotes the set of masked nodes in $\mathcal{I}_h$, $\mathbf{H}_t^L$ denotes the representation of t in $\mathbf{H}^L$, and $\mathbf{W}_r^i$ is the weight matrix for the i -th modality information reconstruction.

The model parameters of PMGT can be learned by minimizing the combined objective function,

$$\mathcal{L}_{edge} + \lambda \mathcal{L}_{feature}. \quad (12)$$

The entire framework can be effectively trained by the end-to-end backpropagation algorithm. To make the training of the model more stable, a mini-batch of nodes are randomly sampled to update the model. When applying the pre-trained PMGT model in downstream tasks, the learned node representations can be either fed into the new tasks directly or with necessary adjustment (e.g., fine-tuning).

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Experimental Datasets. The experiments are performed on the Amazon review dataset [24] and Movielens-20M dataset².

Amazon Datasets. We choose the following 5-score review subsets of the Amazon review dataset for experiments, *i.e.*, “Video Games”, “Toys and Games”, and “Tools and Home Improvement” (respectively denoted by VG, TG, and THI). The metadata of a product includes its text description and the URL of its image³, which are used to extract the textual and visual features of the product respectively. In the experiments, we use the rating data generated before 2015-01-01 for building the product graph, and the rating data generated since 2015-01-01 for studying the performance of the item recommendation and CTR prediction tasks. In these two downstream tasks, we convert all the observed review ratings to be positive interactions and filter out the products that are not included in the product graph.

Moreover, we build the product graph based on users’ review behaviors. Let r_{ht} denotes the number of users who have commonly reviewed the two products h and t . If $r_{ht} \geq 3$, we connect the two products h and t in the graph. In the graph, there inevitably exist some popular nodes with large degrees. This usually causes that the popular nodes with large degrees are more likely to be sampled in the contextual neighbors sampling procedure. To alleviate this problem, we define the edge weight based on the vertex degrees of an edge. Empirically, we define the weight ω_{ht} of the edge e_{ht} between the nodes h and t as follows,

$$\omega_{ht} = \frac{\log(r_{ht}) + 1}{\log(\sqrt{d_h * d_t}) + 1}, \quad (13)$$

where d_h and d_t denote the degrees of the nodes (*i.e.*, products) h and t in the graph, respectively. The operation of $\log(\cdot)$ alleviates the problem of large variance of edge weights in the product graph. By inversely proportional to the degrees of nodes h and t , we reduce the weights of popular nodes.

Movielens-20M Dataset. For the Movielens-20M dataset (denoted by ML), we construct the movie graph based on the tags of movies. The tags of a movie are obtained from the MovieLens Tag Genome Dataset⁴, which includes 11 million computed tag-movie relevance scores from a pool of 1,100 tags applied to 10,000 movies. For each movie, we only keep the tags with relevance scores larger than 0.9. Then, we use r_{ht} to denote the number of common tags two movies h and t have. If $r_{ht} \geq 3$, we construct an edge between the two movies h and t , and the weight ω_{ht} of the edge e_{ht} between two nodes h and t is defined following Eq. (13). We collect the movie trailers from Youtube and extract keyframes for each movie trailer. These keyframes are used as the movie images for extracting the visual modality of a movie. The movie descriptions are collected from TMDb⁵. In addition, the rating data generated since 2008-01-01 are used to evaluate the performance of item recommendation and CTR prediction tasks, where we keep ratings larger than 3 as positive interactions.

²<https://grouplens.org/datasets/movielens/20m/>

³<https://nijianmo.github.io/amazon/index.html>

⁴<https://grouplens.org/datasets/movielens/tag-genome/>

⁵<https://www.themoviedb.org/>

Table 1: Statistics of the experimental datasets.

Datasets	Data for Downstream Tasks			Item Graph	
	# Users	# Items	# Interact.	# Nodes	# Edges
VG	4,525	3,921	27,780	5,032	83,981
TG	31,109	13,870	182,744	17,388	232,720
THI	20,082	11,170	109,717	15,619	178,834
ML	27,715	4,253	2,179,386	4,271	249,498

Multimodal Feature Extraction. In the experiments, we use the pre-trained Inception-v4 network to extract the visual features of each image. Then, we average the visual features of all the images of an item (*i.e.*, product or movie) to obtain its visual modality. For the text description of an item, we utilize the pre-trained BERT model to extract the features of each sentence. Then, we average the features of all the sentences in the description of an item to obtain its textual modality. Empirically, we set the length of the sentence to 128. For the ML dataset, we separate the audio track from the movie trailer with FFmpeg⁶ and adopt VGGish [13] to obtain the acoustic modality of the movie. The dimensionality of the visual, textual, and acoustic modalities are 1,536, 768, and 128, respectively. Table 1 summarizes the statistics of these experimental datasets used for item recommendation and CTR prediction tasks.

4.1.2 Setup and Metrics. After the data pre-processing in Section 4.1.1, an item multimodal graph $\mathcal{G}$ is built, and a set of user-item interactions $\mathcal{D}_{down}$ is prepared for studying downstream tasks.

In the pre-training task, we randomly keep 90% of nodes and their relationships in $\mathcal{G}$ to pre-train the item representations. The remaining 10% of nodes (denoted by $\mathcal{V}_{val}$) are used to construct the validation data for choosing the hyper-parameters of different pre-training models. For each node t in $\mathcal{V}_{val}$, we randomly sample a node t_+ from its one-hop neighbors in $\mathcal{G}$, and use the pair (t, t_+) as a positive example in validation data. And we also randomly sample a node t_- that is not connected with t in $\mathcal{G}$, and use the pair (t, t_-) as a negative example in validation data. The model parameters are chosen based on the AUC (*i.e.*, Area under the ROC Curve) computed on all the validation data.

For downstream tasks, we choose Neural Collaborative Filtering (NCF) [12] as the base model for item recommendation task, and Deep & Cross Network (DCN) [35] as the base model for CTR prediction task. For item recommendation task, we randomly select 80% of the interactions in $\mathcal{D}_{down}$ as training data to update the NCF model, and the remaining 20% of interactions in $\mathcal{D}_{down}$ are used for testing. Moreover, we also randomly hold out 10% of the training data for tuning the hyper-parameters of NCF. In the training of NCF, for each positive interaction pair, we randomly sample one item that has no interactions with the user as negative feedback to update the model. The item recommendation performances achieved by NCF are evaluated by Recall@10, Recall@20, NDCG@10, and NDCG@20 (respectively denote by REC-R@10, REC-R@20, REC-N@10, and REC-N@20). To improve the evaluation efficiency, we randomly sample 1000 items that the testing user has not interacted with to compute the recall and NDCG. For CTR prediction task, we use $\mathcal{D}_{down}$ to simulate the CTR data. For each possible user-item pair in $\mathcal{D}_{down}$, we randomly sample 5 items that have no interactions

⁶<http://ffmpeg.org/>

Table 2: Performances of item recommendation (REC) and CTR prediction by using different pre-training methods.

Datasets	Metrics	Random	DeepWalk	LINE	TransAE	GraphSAGE	GRAPH-BERT	GPT-GNN	PMGT
VG	REC-R@10	0.1994	0.2236	0.2234	0.1989	0.2052	<u>0.2380</u>	0.2251	0.2480
	REC-N@10	0.1278	0.1441	0.1420	0.1217	0.1301	<u>0.1491</u>	0.1343	0.1625
	REC-R@20	0.2742	0.3179	0.3169	0.2903	0.2821	<u>0.3330</u>	0.3269	0.3405
	REC-N@20	0.1494	0.1711	0.1690	0.1480	0.1522	<u>0.1767</u>	0.1636	0.1890
	CTR-AUC	0.7311	0.768	<u>0.7762</u>	0.7675	0.7674	0.7746	<u>0.7839</u>	0.7990
TG	REC-R@10	0.2147	0.2787	0.2805	0.2137	0.2391	<u>0.2858</u>	0.2598	0.3032
	REC-N@10	0.138	0.1847	0.1857	0.1358	0.1514	<u>0.1942</u>	0.1671	0.2056
	REC-R@20	0.3068	0.3807	<u>0.3873</u>	0.3051	0.3352	0.3808	0.3608	0.4030
	REC-N@20	0.1644	0.2141	0.2162	0.1620	0.1790	<u>0.2215</u>	0.1962	0.2342
	CTR-AUC	0.8047	0.8289	0.8326	0.8214	0.8266	<u>0.8322</u>	<u>0.8328</u>	0.8370
THI	REC-R@10	0.1957	<u>0.2043</u>	0.1626	0.1425	0.1921	0.2023	0.1680	0.2358
	REC-N@10	0.136	0.1399	0.0996	0.0861	0.1320	<u>0.1462</u>	0.1047	0.1707
	REC-R@20	0.2555	<u>0.2756</u>	0.2403	0.2191	0.2577	0.2627	0.2451	0.3025
	REC-N@20	0.1529	0.1598	0.1214	0.1077	0.1505	<u>0.1632</u>	0.1264	0.1895
	CTR-AUC	0.7652	0.7850	<u>0.7896</u>	0.7815	0.7643	0.7878	0.7817	0.7933
ML	REC-R@10	0.3104	0.3144	0.3112	0.3096	0.3139	0.3136	0.3162	<u>0.3161</u>
	REC-N@10	0.4714	0.4793	0.4763	0.4703	0.4769	0.4781	0.4823	<u>0.4814</u>
	REC-R@20	0.4556	0.4618	0.4589	0.4549	0.4606	0.4581	<u>0.4628</u>	0.4640
	REC-N@20	0.4829	<u>0.4911</u>	0.4881	0.4822	0.4888	0.4883	0.4928	0.4928
	CTR-AUC	0.9109	0.9218	0.9200	0.9194	0.9145	0.9184	0.9191	<u>0.9205</u>

with the user to construct the negative samples of the CTR data. All the interaction pairs in $\mathcal{D}_{down}$ are used as the positive examples of the CTR data. Then, we randomly sample 80% of the CTR data to train the DCN model, and use the remaining 20% of CTR data to testing the CTR prediction performances. Moreover, 10% of the training data are also held out for tuning the hyper-parameters of DCN. The CTR prediction performances are evaluated by AUC (denoted by CTR-AUC).

4.1.3 Baseline Methods. We compare the proposed PMGT model with the following pre-training methods: 1) **Random**: The item embeddings in the downstream tasks are randomly initialized; 2) **DeepWalk** [26]: This method learns node representations by sampling a large number of paths in the graph and maximizing the average logarithmic probability of all vertex context pairs in sampled paths; 3) **LINE** [32]: This graph embedding method is trained to preserve the first- and second-order proximities of nodes in the graph; 4) **GraphSAGE** [9]: This GNN model forces connected nodes to have similar embeddings by aggregating the information of neighboring nodes; 5) **TransAE** [36]: This method combines a multimodal encoder and the TransE [2] model to learn the node representations; 6) **GRAPH-BERT** [38]: This method applies Transformer to aggregate neighbors’ information without masking operations on the nodes; 7) **GPT-GNN** [15]: This method employs the attribute generation and edge generation tasks to pre-train the GNN model.

For a fair comparison, we use the same multimodal representation in Eq. (3) as the inputs for all pre-training methods.

4.1.4 Implementation Details. For the pre-training and downstream tasks, we set the dimensionality of latent space d_0 to 128. In the experiments, we empirically set the sampling depth K to 3, and the sampling sizes n_1, n_2, n_3 to 16, 8, 4 respectively. The number of contextual neighbors S is selected from {5, 10, 20, 30, 40}. The number of transformer layers L is chosen from {1, 2, 3, 4, 5}. The weight of diversity-promoting attention β is selected from {0, 0.2,

0.5, 0.8, 1.0}. λ is empirically set to 1. We implement PMGT based on TensorFlow [1] and AliGraph [40] frameworks. Adam [17] is used as the optimizer for learning model parameters, and the learning rate is chosen from $\{10^{-4}, 10^{-3}, 10^{-2}\}$.

4.2 Performance Comparison

After pre-training on the item graph, we use the pre-trained item representations to initialize the item embeddings in the downstream tasks. Then, we train the NCF and DCN models and fine-tune the item embeddings based on the user-item interaction data. Table 2 summarizes the performances of NCF and DCN initialized with item representations pre-trained by different methods. We make the following observations. Compared with the random initialization, initializing the base models with pre-trained item representations usually achieves better item recommendation and CTR prediction performances. This demonstrates the pre-training strategies can benefit downstream tasks in recommendation scenarios. The deep pre-training methods GRAPH-BERT, GPT-GNN, and PMGT usually outperform other shallow pre-training methods, by employing GNN to aggregate the neighbor information, and using node feature reconstruction and graph structure reconstruction tasks to pre-train the model. Moreover, PMGT usually achieves the best item recommendation and CTR prediction performances on all datasets. This demonstrates the effectiveness of PMGT in exploiting the item graph structure and item features. In addition, we can also note that PMGT achieves smaller improvements on the ML dataset. One potential reason is that the interaction data of the ML dataset are denser. Thus, with the random initialization, the base models can learn sufficiently good item representations for downstream tasks.

4.3 Ablation Study

We also study the effectiveness of PMGT in exploiting different modality information. As shown in Figure 3, we can note that the

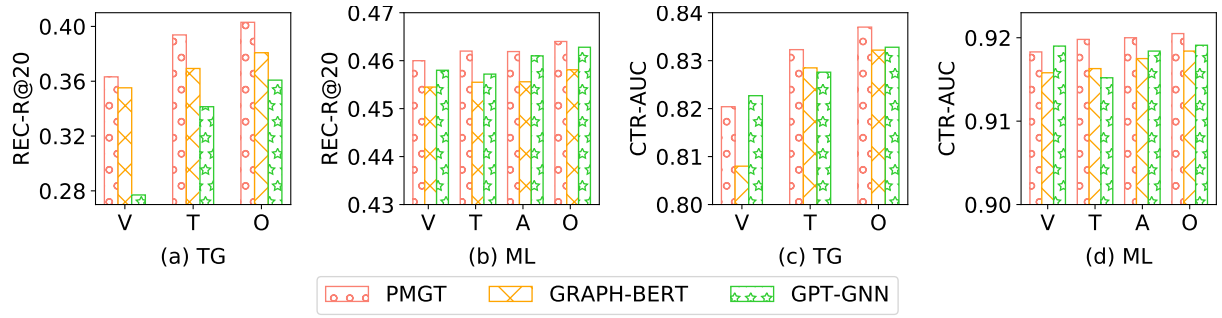


Figure 3: Performances of PMGT, GRAPH-BERT and GPT-GNN considering different modality information on TG and ML datasets. V, T, A denotes the visual, textual, and acoustic modality information respectively. O denotes original models considering all the modality information.

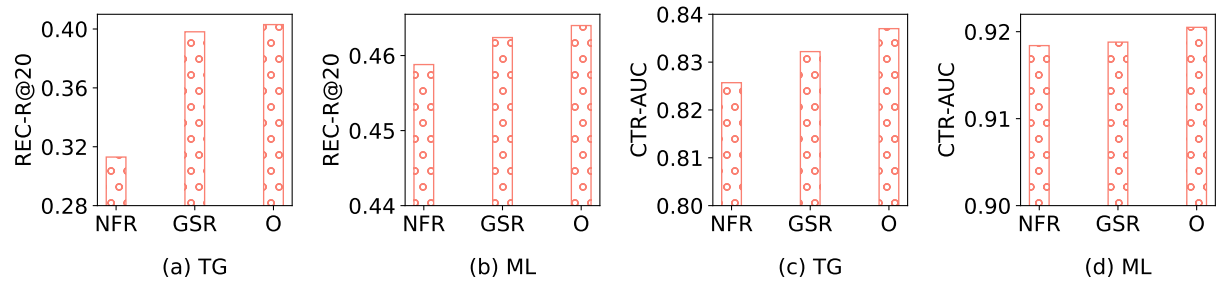


Figure 4: Performances of PMGT considering different graph reconstruction tasks on TG and ML datasets. NFR and GSR denote PMGT only considering the node feature reconstruction task and graph structure reconstruction task, respectively. O denotes the original PMGT considering both tasks.

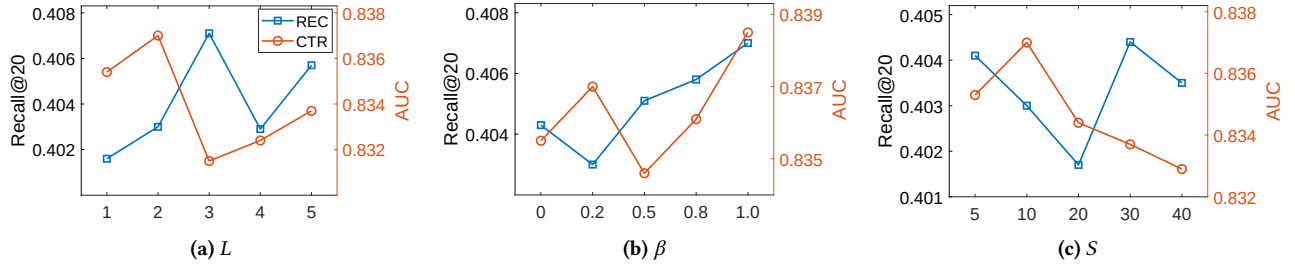


Figure 5: The performance trend of PMGT with respect to different settings of L , β , and S on TG dataset.

original methods considering multimodality information usually outperform the variants that only consider single modality information. This observation is as expected. It indicates that representing items with multimodality information can achieve better performance. PMGT is superior to GRAPH-BERT and GPT-GNN with considering single modality information in most scenarios. This observation again demonstrates that PMGT is more effective in capturing different types of modality information than baseline methods. Moreover, we also study the effectiveness of the two graph reconstruction tasks in learning node representations. Figure 4 summarizes the performances of PMGT variants on the TG and ML datasets. We can note that the original PMGT model with two tasks consistently outperforms the variants using a single task as the pre-training objective. This indicates that both the graph structure reconstruction task and the masked node feature reconstruction task are essential for learning useful node representations to benefit downstream tasks.

4.4 Parameter Sensitivity Study

In this section, we study the performances of PMGT with respect to (w.r.t.) different settings of three important hyper-parameters. Firstly, we vary the number of Transformer layers L from 1 to 5. As shown in Figure 5a, the best item recommendation and CTR prediction performances are achieved by setting L to 3 and 2, respectively. Further stacking more layers does not help improve the performances of downstream tasks. Moreover, we vary the weight of the diversity-promoting attention score β in $\{0, 0.2, 0.5, 0.8, 1.0\}$. As shown in Figure 5b, the recommendation accuracy can be improved by considering diversity-promoting attention in the Transformer-based encoder, when β is set to 0.5, 0.8, and 1.0. This indicates it is important to consider the diversity of contextual neighbors when learning the node representations. Moreover, we also note that the best performances of both downstream tasks are achieved by setting β to 1.0. However, more experiments are needed

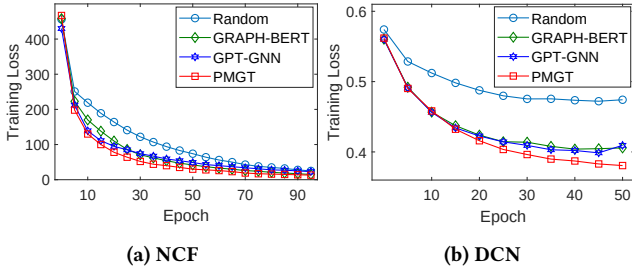


Figure 6: The convergence speed of the training loss of (a) NCF model and (b) DCN model on the TG dataset.

to study whether this is a general observation on different datasets. In the current implementation, we keep β to enable the flexibility of PMGT. Figure 5c summarizes the performances of PMGT w.r.t. different settings of the number of contextual neighbors S . We observe that PMGT usually achieves good performances by setting S to a small value (*e.g.*, 5 and 10). This indicates that a small number of contextual neighbors can capture the important neighborhood information of a node. Further increase of S tends to include noise information, thus may not help improve the model performances.

4.5 Convergence Speed Study

Figure 6 shows the convergence speed of the training losses of both NCF and DCN models on the TG dataset. We initialize the item embeddings in the NCF and DCN models by the following strategies: 1) random initialization, 2) initializing using the representations pre-trained by GRAPH-BERT, 3) initializing using the representations pre-trained by GPT-GNN, and 4) initializing using the representations pre-trained by PMGT. We can make the following observations. Compared with the random initialization, initializing item embeddings with the pre-trained representations achieves faster convergence speed. This once again demonstrates the effectiveness of the pre-training strategies. Moreover, both NCF and DCN models achieve the fastest convergence speed by using pre-trained representations by PMGT. The pre-trained PMGT model gives the downstream model a good initial state, thus can help the downstream model achieves faster and earlier convergence.

5 CASE STUDY IN ONLINE PLATFORM

A case study is conducted in the video recommendation scenario of one of the world’s largest E-commerce platforms. The video graph is built based on users’ watching behaviors. Let r_{ht} denote the number of users who have watched the videos h and t within one hour. If $r_{ht} \geq 10$, we build an edge between the nodes h and t in the video graph. The weight of the edge e_{ht} between h and t is defined following Eq. (13). Finally, there are about 4 million nodes and 500 millions of edges in the video graph used for this case study. Given the pre-trained video representations by PMGT, for a user, we retrieve 50 most similar videos for each video she has watched, based on the Cosine similarity between the video representations. Then, ItemKNN is used to rank and recommend the retrieved videos to the user. After three days of online testing, for 600 thousand users, the number of new video plays increases by 6.80%, compared with the online baseline method using the video representation learned by UNITER [5]. Figure 7 shows two video retrieval examples. We

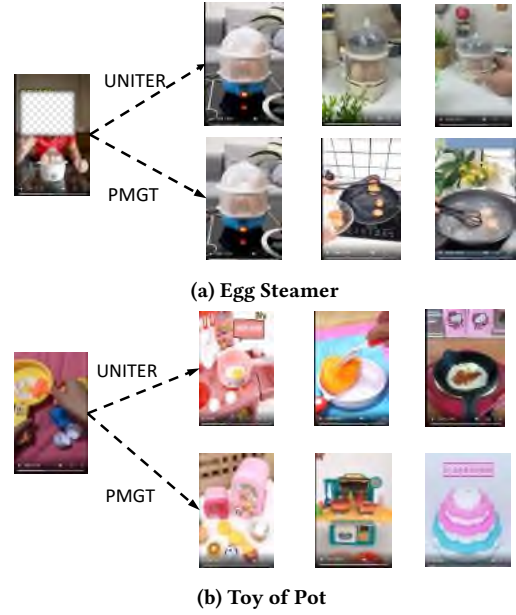


Figure 7: Examples of video retrieval based on the video representations pre-trained by PMGT and UNITER.

can note that the videos retrieved based on the representations pre-trained by PMGT are more diverse than those retrieved based on the representations pre-trained by UNITER.

6 CONCLUSION AND FUTURE WORK

This paper proposes a novel pre-training GNN framework, named PMGT (*i.e.*, Pre-trained Multimodal Graph Transformer), which exploits items’ multimodal information guided by the unsupervised learning tasks on graph. Two graph reconstruction tasks, *i.e.*, graph structure reconstruction and masked node feature reconstruction, are used as learning objectives to pre-train the model. The learned representation of an item not only integrates the multimodal information of the item itself but also aggregates the information of its contextual neighbors in the graph. The superiority of PMGT has been validated by two downstream tasks (*i.e.*, item recommendation and CTR prediction) on real datasets. In this work, we focus on the homogeneous graph of items. For future work, we would like to investigate how to extend the proposed model to process the heterogeneous item graph.

ACKNOWLEDGMENTS

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore. This research is also supported, in part, by the National Research Foundation, Prime Minister’s Office, Singapore under its AI Singapore Programme (AISG Award No: AISG-GC-2019-003) and under its NRF Investigatorship Programme (NRFI Award No. NRF-NRFI05-2019-0002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

- [1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*. 265–283.
- [2] Antoine Bordes, Nicolas Usunier, A. Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data.. In *Neural Information Processing Systems*.
- [3] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1616–1637.
- [4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.
- [5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740* (2019).
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [8] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.
- [9] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.
- [10] Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*. 4918–4927.
- [11] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.
- [12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
- [13] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. CNN architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing (icassp)*. IEEE, 131–135.
- [14] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *ICLR'20*.
- [15] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. GPT-GNN: Generative Pre-Training of Graph Neural Networks. In *KDD'20*.
- [16] Longlong Jing and Yingli Tian. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).
- [17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [18] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
- [19] Guimei Liu, Tam T Nguyen, Gang Zhao, Wei Zha, Jianbo Yang, Jianneng Cao, Min Wu, Peilin Zhao, and Wei Chen. 2016. Repeat buyer prediction for e-commerce. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 155–164.
- [20] Yong Liu, Susen Yang, Yonghui Xu, Chunyan Miao, Min Wu, and Juyong Zhang. 2021. Contextualized Graph Attention Network for Recommendation with Item Knowledge Graph. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [21] Yong Liu, Susen Yang, Yinan Zhang, Chunyan Miao, Zaiqing Nie, and Juyong Zhang. 2021. Learning Hierarchical Review Graph Representations for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [22] Yuanfu Lu, Xunqiang Jiang, Yuan Fang, and Chuan Shi. 2021. Learning to pre-train graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4276–4284.
- [23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
- [24] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 188–197.
- [25] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*. 1532–1543.
- [26] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
- [27] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).
- [28] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *KDD'20*.
- [29] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.
- [30] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 1–45.
- [31] Zhu Sun, Qing Guo, Jie Yang, Hui Fang, Guibing Guo, Jie Zhang, and Robin Burke. 2019. Research commentary on recommendations with side information: A survey and research directions. *Electronic Commerce Research and Applications* 37 (2019), 100879.
- [32] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [34] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax.. In *ICLR'19*.
- [35] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
- [36] Z. Wang, L. Li, Q. Li, and D. Zeng. 2019. Multimodal Data Enhanced Representation Learning for Knowledge Graphs. In *2019 International Joint Conference on Neural Networks (IJCNN)*. 1–8.
- [37] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*. 5753–5763.
- [38] Jiawei Zhang, Haopeng Zhang, Li Sun, and Congying Xia. 2020. Graph-Bert: Only Attention is Needed for Learning Graph Representations. *arXiv preprint arXiv:2001.05140* (2020).
- [39] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.
- [40] Rong Zhu, Kun Zhao, Hongxia Yang, Wei Lin, Chang Zhou, Baole Ai, Yong Li, and Jingren Zhou. 2019. AliGraph: a comprehensive graph neural network platform. *Proceedings of the VLDB Endowment* 12, 12 (2019), 2094–2105.

Shape Controllable Virtual Try-on for Underwear Models

Xin Gao¹, Zhenjiang Liu¹, Zunlei Feng², Chengji Shen², Kairi Ou¹, Haihong Tang¹, Mingli Song^{2*}

¹Alibaba Group, ²Zhejiang University

Hangzhou, China

{zimu.gx, stan.lzj, piaoxue}@alibaba-inc.com, {zunleifeng, chengji.shen, brooksong}@zju.edu.cn, {suzhe.okr}@taobao.com

ABSTRACT

Image virtual try-on task has abundant applications and has become a hot research topic recently. Existing 2D image-based virtual try-on methods aim to transfer a target clothing image onto a reference person, which has two main disadvantages: cannot control the size and length precisely; unable to accurately estimate the user's figure in the case of users wearing thick clothing, resulting in inaccurate dressing effect. In this paper, we put forward an akin task that aims to dress clothing for underwear models. To solve the above drawbacks, we propose a Shape Controllable Virtual Try-On Network (SC-VTON), where a graph attention network integrates the information of model and clothing to generate the warped clothing image. In addition, the control points are incorporated into SC-VTON for the desired clothing shape. Furthermore, by adding a Splitting Network and a Synthesis Network, we can use in-shop clothing/model pair data to help optimize the deformation module and generalize the task to the typical virtual try-on task. Extensive experiments show that the proposed method can achieve accurate shape control. Meanwhile, compared with other methods, our method can generate high-resolution results with detailed textures, which can be applied in real applications.

CCS CONCEPTS

• Applied computing → Online shopping; • Computing methodologies → Image-based rendering.

KEYWORDS

Virtual try-on; Graph attention networks; Image warping

ACM Reference Format:

Xin Gao¹, Zhenjiang Liu¹, Zunlei Feng², Chengji Shen², Kairi Ou¹, Haihong Tang¹, Mingli Song². 2021. Shape Controllable Virtual Try-on for Underwear Models. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21)*, October 20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3474085.3475210>

1 INTRODUCTION

The online virtual try-on system [14, 15, 17, 18, 20, 28, 31, 34, 35, 37, 47, 51, 53, 55] has become a research hot-spot in recent years to

*Corresponding authors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '21, October 20–24, 2021, Virtual Event, China

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8651-7/21/10...\$15.00
<https://doi.org/10.1145/3474085.3475210>



Figure 1: Shape controllable visual results. The shape can be continuously controlled in length and tightness.

accommodate the vast market demand. Virtual try-on applications provide users with convenient what-you-see-is-what-you-get function: users can pick a clothing and put it on themselves virtually, and matching their favorite. It is still a big challenge for virtual try-on algorithms to make the results both realistic and attractive.

Existing virtual try-on techniques can be divided into 3D virtual try-on methods and 2D image-based methods. 3D virtual try-on methods need to model characters and clothing, then render the digitalized clothing on the character, achieving controllable and detailed results. However, the process of modeling for characters and clothing is costly and time-consuming. Rendering process also requires high performance of mobile devices.

Han et al. [15] proposed a 2D image-based virtual try-on task that adopts model and clothing pairs to train a clothing deformation sub-network and then composites the deformed clothing and coarse model with a generator. Following [15], many 2D image-based methods [18, 47, 53, 55] have been proposed. Most of them focus on finding a better deformation module or a more realistic generator. However, few of those methods could control clothing shape, which is a significant capacity for the virtual try-on task. What's more, existing 2D virtual try-on methods use parsing and key points information of the clothed model image as input, which cannot accurately obtain the model's figure and leads to inaccurate dressing effects. Limited by low-resolution datasets and methodological limitations, most methods can only generate low-resolution results and have obvious blurry visual results on high-resolution results, which is not applicable in the online e-commerce scene.

This paper proposes the task that dressing clothing for underwear models, which is also an urgent demand for online clothing shops to exhibit new clothing efficiently. The model's figure can be more easily got when they wear underwear than thick clothing.

Considering the real application requirement and the drawbacks of existing 2D image-based methods, we propose a Shape Controllable Virtual Try-On Network (SC-VTON) for the underwear model dressing task. As shown in Fig. 1, SC-VTON can achieve continuous shape control in tightness and length. SC-VTON uses GAT (Graph Attention Network) [45] to learn the deformation offset of the meshed clothing. A shape information extraction module is used to extract the features of the underwear model and clothing. The extracted feature maps are concatenated as shape vectors in channel-dimension. With the guidance of shape vectors and control points (the control points are some discrete points near the shape of the model), GAT predicts coordinate offsets of all mesh vertices. The warped clothing image is obtained by using the differentiable rendering module. The underwear model and clothing pairs for training SC-VTON are generated with an annotation tool based on As Rigid As Possible (ARAP) [42] with clothing key points as control points.

Furthermore, in-shop clothing/model pairs are adopted to optimize SC-VTON. To generate self-loop supervised information, we devise a splitting network to predict the underwear model, and a synthesis network to synthesize the underwear model with the clothing image. The self-loop supervised information will enhance the robustness of SC-VTON in the real application by combining SC-VTON with the pre-trained splitting network and synthesis network. What’s more, with the splitting and synthesis networks, the proposed SC-VTON can be easily extended to normal virtual try-on for a model in clothing.

Therefore, our approach is the first method for the underwear model dressing task, which can adaptively generate high-quality results with different resolutions. The proposed SC-VTON is the first method that achieves continuous shape control in length and tightness. What’s more, the devised self-loop optimization improves the robustness of SC-VTON and extends the framework to the typical virtual try-on task for a model in clothing. Extensive experiments demonstrate that our method achieves more pleasing visual results than the state-of-art methods.

2 RELATED WORKS

2.1 Virtual Try-on

Virtual try-on systems have been an attractive topic even before the renaissance of deep learning.

3D virtual try-on systems based on human reconstruction in computer graphics have been widely researched, such as [1, 3, 27, 30, 44]. While Physics-Based Simulation (PBS) can accurately drape a 3D garment on a 3D body [12, 33, 40, 46], it remains too costly for real-time applications. 3D human estimation methods try to digitalize real-world 2D character photos to 3D models, which have made significant progress after the popularity of deep learning, such as [10, 22, 25, 52], while [13, 21, 38, 39] also digitizing with garments, these methods still need to be optimized in texture authenticity.

Recently 2D image-based virtual try-on tasks have gained much attention from academia and industry. It is highly challenging because it requires to warp the clothing on the target person while preserving its patterns and characteristics.

Han et al. [15] proposed a 2D image-based virtual try-on task that adopts model and clothing pairs to train a clothing deformation sub-network and then composites the deformed clothing and coarse model with a generation network. Based on [15], Wang et al. [47] refined the architecture with a geometric matching module to learn parameters for Thin-Plate Splines (TPS) deformation [50]. Considering the complexity of pose and self-occlusion cases, Yu et al. [55] and Raffee et al. [34] took parsing information into account and used a generator to predict the target model’s parsing result, which could generate better results in self-occlusion cases. What’s more, Wu et al. [51] adopted DensePose [2] to get more precise information of the model. Minar et al. [31] solved the problem by first mapping 2D clothing texture to 3D character model and deforming the clothing in 3D space, and then back to 2D space. Chaudhuri et al. [4] proposed a method to generate diverse high fidelity texture maps for 3D human meshes in a semi-supervised setup. On the other hand, some researchers focus on the improvement of the deformation module. In [17, 18, 20, 53], the spatial transformer networks [19] is adopted to get more accurate deformation result. Roy et al. [37] adopted human and fashion landmarks to help optimize thin-plate splines deformation model. Han et al. [14] and Ren et al. [35] adopted optical flow to get a more natural texture deformation effect. Li et al. [28] tried to get better results by using multiple specialized warps. Ge et al. [9] proposed a knowledge distillation method to produce images without human parsing.

Based on the generated high-resolution images of StyleGAN [23], Yildirim et al. [54] trained a conditional StyleGAN for generating virtual try-on images with clothing image as conditional input on around 380K entries. Cheng et al. [6] embedded human geometries into the latent space as independent codes and achieved flexible and continuous control of geometries via mixing and interpolation operations in explicit style representations. However, acceptable visual results of StyleGAN usually requires hundreds of thousands of samples, which is very difficult to obtain.

2.2 GCN based Deformation

Graph Convolutional Network (GCN) [24] extends convolution operation from euclidean data (such as images) to a more extensive topological structure and is widely used in natural language processing, recommendation system, computer vision, and computer graphics domains. Wang et al. [48] represented 3D meshes in a graph-based CNN and produced correct geometry by progressively deforming an ellipsoid. Based on [48], Wen et al. [49] used GCN to solve the problem of shape generation in 3D mesh representation from a few color images with known camera poses. For the virtual try-on task, Vidaurre [46] used GCN for parametric predefined 2D panels with arbitrary mesh topology. Kolotouros et al. [25] adopted a Graph-CNN to estimate 3D human pose and shape from a single image. Zhu et al. [56] used GCN to predict 3D clothing’s feature line and use it to warp clothing.

Graph Attention Networks (GATs) [45] enable specifying different weights to different nodes in a neighborhood, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions. In this work, we consider using GAT to get more precise warping results.

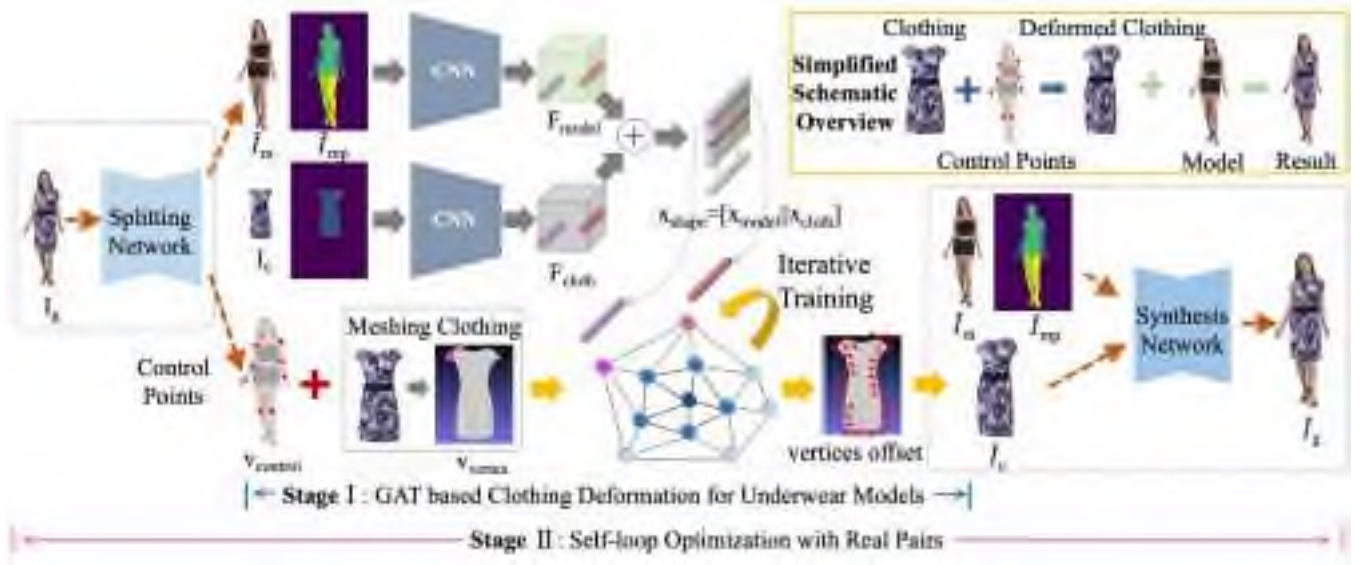


Figure 2: An overview of the whole framework. Stage I : The shape information extraction model extract feature maps from the underwear model and clothing image with CNN extractors. The clothing is meshed and presented as a graph. GAT based deformation module predicts the coordinate offsets with the clothing graph as input under the guidance of control points and shape information extracted from the model and clothing feature maps. The warped clothing image is obtained by using the differentiable rendering module. Stage II : The splitting network and the synthesis network are added into SC-VTON (Stage I). The whole framework is extended to the typical virtual try-on task for clothed model image. Then, the whole framework can be trained with real pairs (clothing image and clothed model image).

3 METHODS

Different from the typical virtual try-on task, the primary goal of the paper is to wear clothing for an underwear model: given an underwear model image I_m and a clothing image I_c , try to generate model image I_g wearing the clothing with the same pose as I_m . It is hard to get the pair $\{I_m, I_c, I_g\}$ that the underwear models and the model wearing clothing with the same pose at the same time. We adopt an annotation tool based on ARAP with manually fine-tuning to get the pseudo-labeled pairs $\{I_m, I_c, I_g\}$, which are used as ground truth pairs for SC-VTON on stage I (Section 3.1). Furthermore, real pairs $\{I_c, I_g\}$ are used to improve the robustness of SC-VTON in application scenarios. To adapt to real pairs $\{I_c, I_g\}$, a splitting network and a synthesis network are added into SC-VTON, which is extended to handle virtual try-on task for model in clothing (Section 3.2). The whole framework is illustrated in Fig. 2.

3.1 GAT based Clothing Deformation for Underwear Models

The GAT based clothing deformation contains three parts: the shape information extraction module, the control points guided deformation module and the differentiable rendering module.

The shape information extraction module is devised for extracting features from the clothing image and the underwear model image. The extracted feature maps will be concatenated as shape vectors and sent to the deformation module as part of the guiding information. For the control points guided deformation module, the clothing firstly meshes with the Delaunay Triangulation

Algorithm[26], then the meshed coordinate points are set as vertices V of a graph $G = (V, E)$, where E denotes the edge set (See *supplementary materials* for more details). For each vertex in V , the vertex’s information is expressed as a vertex vector. Next, the shape control points and the shape vectors are integrated into the corresponding vertex vector as the deformation’s guiding information. GAT predicts coordinate offsets of all mesh vertices. The warped clothing image is obtained by using the differentiable rendering module. The whole deformation process is trained in an iterative manner to obtain higher accuracy. The following subsections give detailed descriptions.

3.1.1 Shape Information Extraction Module. The shape information extraction module contains two branches. The first branch extracts model’s shape feature maps F_{model} from the original model image and model’s parsing image with a CNN extractor. The second branch extracts clothing feature maps F_{cloth} from the original clothing image and clothing’s parsing image with the same architecture (not sharing weights). It is worth noting that the clothing image has been coarsely aligned to the model image with some key points as references. For each vertex in the clothing graph, the Perceptual Feature Pooling [48] is adopted to find the mapping relationship between each vertex’s coordinate and the feature vector’s coordinate in the extracted feature maps F_{model} and F_{cloth} . Next, feature vectors $x_{model} \in F_{model}$ and $x_{cloth} \in F_{cloth}$ with the same coordinate are concatenated as shape vectors $x_{shape} = [x_{model} || x_{cloth}]$, which will be appended to the vertex vector x (here $||$ denotes concatenation).



Figure 3: Control points definitions. The control points are discrete points near the silhouette of the model. The red points show the control points with different clothing types.

3.1.2 Control Points Guided Deformation Module. Here we first introduce how to get the meshed clothing. A Deeplabv3+ model [5] is trained to get clothing’s parsing results. Then, we uniformly sample points along the contour of clothing according to the parsing results. Next, the Delaunay Triangulation [26] is adopted to mesh the inner part of the clothing. Contour points and inner points form the vertex set V (See *supplementary materials* for more details). Each vertex’s information is expressed as a vertex vector $x = [v_{vertex} || v_{control} || x_{shape}]$, which is composed of three parts: the vertex’s coordinate v_{vertex} , the control point’s coordinate $v_{control}$ and the corresponding shape feature vector x_{shape} .

Control points are some discrete points near the silhouette of the model. The red points in Fig. 3 show control points with different clothing types. Control points are obtained indirectly through model key points and model parsing results. SC-VTON can control both the tightness and length of the clothing, points near the model’s limbs and torso are used. By slightly adjusting the control points’ position (For length, adjusting the points position along the extension of the limbs. For tightness, a dilate operation is used for the model’s parsing result and finding the corresponding position according to the dilated parsing), we can achieve a shape controllable virtual try-on result.

GAT introduces the attention mechanism as a substitute for the statically normalized convolution operation. Below are the equations to compute the node embedding h_i^{l+1} of layer $l + 1$ from the embeddings of layer l .

$$z_i^l = W^l h_i^l,$$

$$\alpha_{ij}^l = \frac{\exp(\text{LeakyReLU}(\vec{a}^{lT} (z_i^l || z_j^l)))}{\sum_{k \in N(i)} \exp(\text{LeakyReLU}(\vec{a}^{lT} (z_i^l || z_k^l)))},$$

$$h_i^{l+1} = \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^l (z_j^l) \right),$$

where h_i^l is the lower layer embeddings and W^l is its learnable weight matrix. The second equation computes a pair-wise normalized attention score between neighbors, j is the neighbor of node i . Here, it first concatenates the z embeddings of the two nodes, then takes a dot product of it and a learnable weight vector $\vec{a}$, and applies a LeakyReLU function. Each node applies a softmax to normalize the attention scores on each node’s incoming edges.



Figure 4: Details of the GAT based deformation module. The input contains clothing meshes and control points, as well as clothing and model features extracted by CNN extractors. The output is the vertex’s coordinate offset. An iterative training strategy is adopted to optimize the result. For each iteration, model features will be recalculated according to the updated vertices coordinates.

Finally the embeddings from neighbors are aggregated together, scaled by the attention scores and applied a nonlinearity σ .

The network structure of adopted GAT is shown in Fig. 4. Since the task is a deformation task using low-level features, shallow GAT with two graph convolution blocks are used. Each block is composed of three graph attention layers. The ground truth of GAT deformation is the offset between the vertex’s coordinate in clothing image I_c and generated pseudo image I_g . In the training stage, the vertex number K of each clothing image sample could be different and can be batched using diagonal alignment.

3.1.3 Differentiable Rendering Module. GAT predicts coordinate offsets of all vertices, the differentiable rendering module is then used to get the warped clothing image. We first get an indicator matrix $\mathbb{I} \in \mathbb{R}^{H \times W \times N}$ during the preprocessing step, the last dimension is one-hot vector indicates which mesh each pixel belongs to (N is the number of meshes). An affine transformation matrix with all meshes $M \in \mathbb{R}^{N \times 2 \times 3}$ can be get by multiply the coordinates of warped mesh’s vertices and the inverse of the coordinates of the original mesh’s vertices. The warped clothing image $\tilde{I}_c$ can be get by the following equation:

$$\tilde{I}_c = \text{GridSample}(I_c, G_c \odot (\mathbb{I} \times M)),$$

where I_c is the original clothing image, G_c is the normalized grid of the clothing image, after getting the flow-grid of all pixels, Grid-Sample [19] is used to get the warped clothing image $\tilde{I}_c$. This step is differentiable and can be trained in an end-to-end manner.

3.1.4 Iterative Training Policy. In the training stage, the iterative training policy is adopted to optimize the results continuously. After each iterative training, the Perceptual Feature Pooling is reused to extract new feature vector x_{model} from the feature map F_{model} according to the updated mesh coordinates, see Fig. 4 for more detailed information. In the experiment, SC-VTON achieves satisfying results with 2-3 iterations.

To achieve a controllable and precise result, SC-VTON should satisfy two principles under the constrain of control points. Firstly, the deformed clothing should fit and cover the model’s body as much as possible; Secondly, the deformed clothing should maintain the clothing silhouette. The training loss $\mathcal{L}_{gat}$ of SC-VTON is

composed of three parts: $\mathcal{L}_{vertex}$, $\mathcal{L}_{pixel}$ and $\mathcal{L}_{smooth}$. The first term is Huber loss [16] for constraining all vertices of the meshed clothing, the second term is the L2 pixel loss, the last term $\mathcal{L}_{smooth}$ is the smoothing constraint term for adjacent vertices:

$$\begin{aligned}\mathcal{L}_{gat} &= \mathcal{L}_{vertex} + \lambda_1 \mathcal{L}_{pixel} + \lambda_2 \mathcal{L}_{smooth}, \\ \mathcal{L}_{vertex} &= \sum_{k=1}^K \text{Huber}(\Delta v_k, \Delta \bar{v}_k), v_k \in V, \\ \mathcal{L}_{pixel} &= \|I_c - \tilde{I}_c\|_2^2, \\ \mathcal{L}_{smooth} &= \frac{1}{K} \sum_{k=1}^K \sum_{t=1}^{|Neighbor(v_k)|} \|\Delta(v_k, v_t) - \Delta(\hat{v}_k, \hat{v}_t)\|_2^2,\end{aligned}$$

where K denote the vertex number of all vertices V , Δv_k denote the predicted vertex coordinate offset, $\Delta \bar{v}_k$ denote the ground truth of vertex coordinate offset, $|Neighbor(v_k)|$ denotes the number of the adjacent vertex set of the vertex v_k , $\Delta(v_k, v_t)$ denotes the vertex offset between the vertex v_k and its adjacent vertex v_t in original clothing image, $\Delta(\hat{v}_k, \hat{v}_t)$ denotes the vertex offset between the vertex $\hat{v}_k$ and its adjacent vertex $\hat{v}_t$ in the deformed clothing image, λ_1, λ_2 are the balance parameters.

3.2 Self-loop Optimization with Real Pairs

In Section 3.1, large number of pseudo-labeled pairs $\{I_m, I_c, I_{g'}\}$ are adopted to train SC-VTON, the pseudo-labeled pairs are generated with an ARAP tool, which limits the performance on real data.

To solve the above drawbacks, real pairs $\{I_c, I_g\}$ of the clothing image I_c and the model I_g in the same clothing are adopted to optimize SC-VTON. We devise two sub-networks: a splitting network and a synthesis network, and add them into SC-VTON, which can generate self-loop supervision information for promoting robustness and performance of SC-VTON. The splitting network is devised for predicting underwear model I'_m and control points from the clothed model image I_g . The synthesis network is devised for synthesizing the clothed model image I'_g from the generated underwear model image I'_m and the warped clothing image $\tilde{I}_c$ as input. The difference between real clothed model image I_g and synthesis clothed image I'_g will generate the supervision information for training the whole framework.

3.2.1 Splitting Network. As described above, the inputs of the splitting network are the real clothed model image I_g , the output contain the predicted underwear model image $\hat{I}_m$, underwear model's parsing result I_{mp} and control points. Before training the whole framework together, the splitting network is pre-trained by pseudo-labeled pairs $\{I_m, I_c, I_{g'}\}$ with the following loss:

$$\begin{aligned}\mathcal{L}_{split} &= \|\hat{I}_m - I_m\|_2^2 + \alpha_1 \sum_{t=1}^T \|\phi_i(\hat{I}_m) - \phi_i(I_m)\|_1 \\ &\quad - \alpha_2 I_{mp} \log(\hat{I}_{mp}) + \alpha_3 \sum_{k=1}^K \|\hat{v}_{control}^k - v_{control}^k\|_2^2,\end{aligned}$$

where I_{mp} and $\hat{I}_{mp}$ are the origin and predicted parsing of the model, $\hat{v}_{control}^k$ and $v_{control}^k$ are the coordinates of the predicted control

points and labeled control points, $\phi_i(I_m)$ and $\phi_i(\hat{I}_m)$ denote the feature maps of image I_m and generated image $\hat{I}_m$ of the t -th layer ($T = 4$) in the visual perception network ϕ , which is a VGG19 [41] pre-trained on ImageNet [7]. $\alpha_1, \alpha_2, \alpha_3$ are the balance parameters.

3.2.2 Synthesis Network. As depicted in Fig. 2, the synthesis network contains three input parts: the warped clothing image, the underwear model image, and the corresponding parsing. The output of the synthesis network is the synthesis clothed model image. Similarly, the synthesis network are also pre-trained on the pseudo-labeled pairs $\{I_m, I_c, I_{g'}\}$ with the following loss:

$$\mathcal{L}_{synth} = \|\hat{I}_{g'} - I_{g'}\|_2^2 + \beta \sum_{t=1}^T \|\phi_i(\hat{I}_{g'}) - \phi_i(I_{g'})\|_1,$$

where $\hat{I}_{g'}$ is the synthetic clothed model image with the pseudo-labeled pair as input. β is the balance parameter.

3.2.3 Whole Framework Optimization. With the pre-trained splitting network and the synthesis network, the whole framework can be trained on real pairs $\{I_c, I_g\}$ with the following loss:

$$\mathcal{L}_{real} = \|\hat{I}_g - I_g\|_2^2 + \gamma \sum_{t=1}^T \|\phi_i(\hat{I}_g) - \phi_i(I_g)\|_1,$$

where $\hat{I}_g$ is the synthetic clothed model image of whole framework, I_g is the input with the real clothed model image, γ is the balance parameter.

In the training stage of the whole framework, SC-VTON is also trained on the pseudo-labeled pairs simultaneously. The total loss $\mathcal{L}$ of the whole framework is defined as follows:

$$\mathcal{L} = \mathcal{L}_{synth} + \eta \mathcal{L}_{real},$$

where η is the balance parameter.

3.2.4 Extension to Clothed Model Virtual Try-on Task. With the splitting network and the synthesis network, the whole framework can be easily extended to the virtual try-on task for the clothed model image. For a clothed model image, the splitting network first predicts the underwear model image, parsing, and control points. Then GAT uses the meshed clothing, predicted underwear model, and control points as input to get the warped clothing image. The synthesis network synthesizes the final result.

4 EXPERIMENTS

4.1 Implementation Details

Underwear Model Virtual Try-On (UMV) Dataset. The UVM dataset consists of 143,428 pseudo labeled pairs in total. To get the pseudo labeled pairs, we photographed 2648 underwear models with different poses and collected tens of thousands of clothing images from the e-commerce platform. Key points and parsing labels for both models and clothing are defined. We manually label the model images to get accurate results. For clothing images, we use pre-trained models [29, 43] to predict key points and parsing results, and remap to our own definitions. An ARAP based annotation tool is used with manually fine-tune to get the pseudo label $I_{g'}$. *supplementary materials* provide more detailed descriptions of the UVM dataset.

Multi-Pose Virtual Try-On (MPV) Dataset. MPV dataset [8] consists of 37,723/14,360 person/clothing images, the dataset used for experiments is one part of the MPV dataset, consisting of 14,754 pairs of top clothing images and positive perspective images of female models. The resolution of images in the dataset is 256×192 .

Network Architecture. The GAT module contains two graph attention blocks, each with three graph attention layers followed by a ReLU activation layer. The output dimension of each layer is 256, and the final output dimension is 2. The feature maps of the former three layers with 64, 128, 256 channels of VGG network are used in the shape information extraction module. For each vertex, Perceptual Feature Pooling extract the feature from four near pixels using bilinear interpolation according to its coordinate position and get a 1×448 vector. Two 1×448 vectors extracted from the model and the clothing are concatenated into a 1×896 vector, and then reduce the dimension by a fully connection layer to get the final 1×256 vector.

The Splitting Network is an U-Net [36] like encoder-decoder network. Three contracting blocks are used in Encoder, each with two convolution layers, the output stride is 8. The decoder uses three expansive blocks, each with two convolution layers and a transposed convolution to recover feature map size and concatenate with the corresponding contracting block’s output. The Synthesis Network’s has the similar network architecture as the Splitting Network. More details about network architecture are given in the *supplementary materials*.

Parameters Setting. As for training parameters, we set batch size 4, the epoch is 30, and 2 iterations are used. ADAM optimizer is used with $\alpha = 0.5$ and $\beta = 0.999$, the learning rate is 10^{-3} . We set $\lambda_1 = 1$, $\lambda_2 = 10$, $\alpha_1 = 0.1$, $\alpha_2 = 1$, $\alpha_3 = 1$, $\beta = 0.1$, $\gamma = 0.1$, $\eta = 1$. Hyperparameters are set in terms of the importance of each subterm, and the balance of magnitudes between each loss.

4.2 Experiment on Controllable Shape

To verify the effectiveness of shape control, we set different control points as guidance for the same clothing and underwear model. Fig. 5 shows the results of various models and clothing, where we can see that the tightness of sweater and pants, the length of skirt and sleeves are successfully controlled with different control points as inputs. What’s more, for the pants in the middle column of Fig. 5, the tightness and length are controlled simultaneously.

4.3 Qualitative Results

Fig. 6 illustrates our results compared with other methods on test samples selected from the MPV dataset. CP-VTON [47], CP-VTON+ [32] and ACGPN [53] are adopted for comparison. The textures of clothing range from simple to complex. Compared with CP-VTON and CP-VTON+, the texture of our approach is much clearer, please see the results of the last two rows in Fig. 6. Meanwhile, compared with ACGPN, our approach can better maintain the clothing’s silhouette. For example, for the two cases in the last row in Fig. 6, ACGPN has turned short sleeves into sleeveless, which is an obvious mistake. Overall, our method is better at keeping the clothing’s texture and can better maintain the clothing’s silhouette.

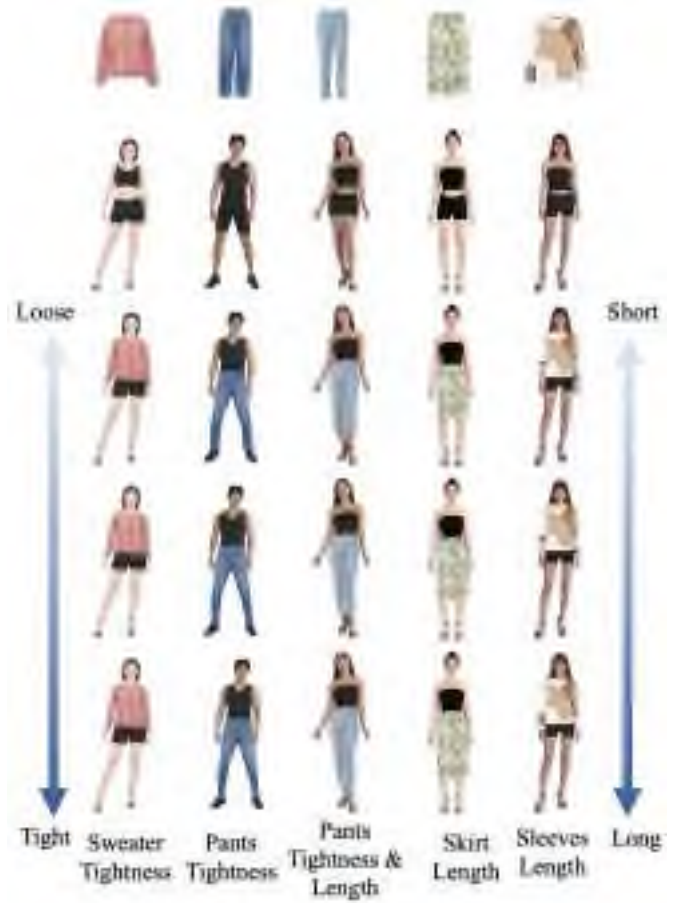


Figure 5: Controllable shape results with different control point setting. The shape can be continuously controlled in length and tightness.

4.4 Quantitative Results

We conduct a user study on the Qince Platform (A crowdsourcing platform developed by Alibaba Inc.) Given the clothing image, and two synthesized clothed model images by two different methods (with resolution 256×192 and 512×384 , respectively), the worker is asked to choose the one is more realistic and accurate in a virtual try-on situation. We choose 500 testing samples from the MPV dataset. All synthesized results are presented five times to different workers to avoid individual preferences. Quantitative comparisons are summarized in Table 1. As can see from the table, our method gives better results than the other methods at a resolution of 256×192 . When the resolution is enlarged to 512×384 , our evaluation results are significantly better than the other methods, demonstrating that our method can better maintain texture details for large resolution images.

4.5 Comparing the Warping Results

As shown in Fig. 7, we further analyze and show the visual comparison of warping modules with CP-VTON, CP-VTON+ and ACGPN



Figure 6: Visual comparison with CP-VTON [47], CP-VTON+ [32] and ACGPN [53]. The textures of clothing range from simple to complex, our method is better at keeping the clothing’s texture and maintaining the clothing’s silhouette.

Table 1: Perceptual user study results with other methods on the MPV dataset. The score represents the proportion of samples that workers considered more realistic.

Method	Resolution 256	Resolution 512
CP-VTON [47]	0.37	0.27
SC-VTON (Ours)	0.63	0.73
CP-VTON+ [32]	0.43	0.33
SC-VTON (Ours)	0.57	0.67
ACGPN [53]	0.42	0.38
SC-VTON (Ours)	0.58	0.62

Table 2: Precision within a certain threshold on upper clothing, pants and their average. Larger is better. SC-VTON[†] is the optimal model trained with UMV. SC-VTON* is the optimal model trained with both UMV and MPV.

Method	Uppers	Pants	Average
SC-VTON (w/o cloth info)	0.67	0.80	0.74
SC-VTON (w/o $\mathcal{L}_{smooth}$)	0.84	0.87	0.86
SC-VTON (w/o iterative training)	0.71	0.83	0.77
SC-VTON [†] (UMV)	0.76	0.83	0.80
SC-VTON* (UMV & MPV)	0.78	0.85	0.82

on MPV dataset. Both CP-VTON and CP-VTON+ use the Spatial Transformation Network (STN) to learn parameters of Thin-Plate



Figure 7: Warping results with CP-VTON, CP-VTON+ and ACGPN. Compared with other methods, our warping results are flatter and rigid, and also better maintain the original silhouette of the garment.

Spline (TPS) to warp the clothing image. ACGPN uses TPS with a second-order difference constraint to get less distortion results. As shown in Fig. 7, the results of CP-VTON and CP-VTON+ are often distorted in the sleeve and torso parts, and ACGPN is prone to



Figure 8: Visual results of ablation study experiments. From left to right: model image, result without using clothing information in GAT, result without using $\mathcal{L}_{smooth}$, result without using iterative training in GAT. SC-VTON[†] is the optimal model trained with UMV. SC-VTON* is the optimal model trained with both UMV and MPV.

skewing. This is because the deformation between different clothing parts can affect each other using TPS.

We take this into account in the pre-processing stage. For the parsing definitions of person and clothing, we distinguish between different parts. For example, the left arm, torso and right arm are marked as different parsing categories (See *supplementary materials*), thus we know which part each mesh belongs to when meshing the clothing. GAT allows overlap between different meshes, and different parts do not affect each other. During the rendering step, the overlapping regions can be handled orderly. Therefore, compared to other methods, our warping results are flatter and rigid, and also better maintain the original silhouette of the garment.

4.6 Ablation Experiments

Here we conducted a series of ablation experiments to verify each module and loss’s effectiveness of the proposed methods. For evaluation metric, we calculate the precision by checking the percentage of vertices between prediction and ground truth within a certain threshold $\tau = 10^{-2}$.

Table 2 and Fig. 8 show the quantitative and qualitative results of the ablation study. SC-VTON[†] is the optimal model trained with UMV dataset. SC-VTON* is the optimal model trained with both UMV and MPV datasets. From Table 2 and Fig. 8, we can see that removing clothing’s image and parsing information made the results worse. Removing $\mathcal{L}_{smooth}$ increases the accuracy, but the silhouette of the clothing becomes jagged. There is also a slight decrease with no iteration during training, demonstrating that iterative training can improve the deformation result. Training with UMV and MPV simultaneously also slightly boosts the performance.

4.7 Extend to Typical Virtual Try-on Task

As described in Section 3.2, with the splitting network and the synthesis network, the whole framework can be extended to typical virtual try-on task for the clothed model. At inference time, both the clothed model image I_m and a different clothing image I_c are sent to the network. We can directly put the warped clothing image to the predicted underwear model to get the final result, or use the Synthesis Network to generate a refined result. More results are given in the *supplementary materials*. The task of predicting the



Figure 9: Failure cases. Our method fails on side posture with large angles. For the obscured part (left arm), the result of deformation is wrong.

underwear model is very difficult, hence the visual result of II is not quite good. Even so, adding II can optimize the warping result, while increasing the robustness on new samples.

4.8 Failure Cases

All warping methods based on 2D clothing images face a common problem: cannot handle the side postures. Most works infer the side textures of the clothing by generators, the results are often blurry. Our method also has this problem and this paper only deal with the front view models. For side posture cases, the results are not good, please see Fig. 9 for visual results. We came up with some solutions to solve the problem as our future works: for person modeling, 3D information need to be considered, DensePose [11] or SMPL [30] model may be two options. A more detailed clothing dataset need to be collected, which should contain at least two clothing images of front and back views. For the warping algorithm, depth information need to be considered during the warping process.

5 CONCLUSION

In this paper, we propose a Shape Controllable Virtual Try-On Network (SC-VTON) to wear clothing for underwear models, which is an urgent demand for an online clothing shop to exhibit new clothing efficiently. The proposed method comprises two parts: GAT based clothing deformation for the underwear model and self-loop optimization with real pairs. The former is devised for deforming clothing for the underwear model under the constraint of shape control points. The latter is introduced for improving the robustness and performance of SC-VTON, which also extends the whole framework to the typical virtual try-on task. The advantage of the proposed method is that SC-VTON can achieve continuous shape control in length and tightness. As far as we know, SC-VTON is the first shape controllable method for the 2D image-based virtual try-on task. What’s more, based on the inherent advantage of the underwear model image, we can extract more accurate model’s figure information than the clothed model, which brings in more pleasing virtual try-on visual results. Our method can maintain detailed texture and clothing silhouette for high-resolution image, which has practical advantages in real applications.

ACKNOWLEDGMENTS

This work is supported by Key Research and Development Program of Zhejiang Province (2018C01004), National Natural Science Foundation of China (61976186, U20B2066), Fundamental Research Funds for the Central Universities (2021FZZX001-23), Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies.

REFERENCES

- [1] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. 2019. Learning to reconstruct people in clothing from a single RGB camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1175–1186.
- [2] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7297–7306.
- [3] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. 2019. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5420–5430.
- [4] Bindita Chaudhuri, Nikolaos Sarafianos, Linda Shapiro, and Tony Tung. 2021. Semi-supervised Synthesis of High-Resolution Editable Textures for 3D Humans. *arXiv preprint arXiv:2103.17266* (2021).
- [5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 801–818.
- [6] Haoqing Cheng, Heng Liu, Fei Gao, and Zhuo Chen. 2020. ADGAN: A Scalable GAN-based Architecture for Image Anomaly Detection. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Vol. 1. IEEE, 987–993.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
- [8] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. 2019. Towards multi-pose guided virtual try-on network. In *Proceedings of the IEEE International Conference on Computer Vision*. 9026–9035.
- [9] Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. 2021. Parser-Free Virtual Try-on via Distilling Appearance Flows. *arXiv preprint arXiv:2103.04559* (2021).
- [10] Riza Alp Güler and Iasonas Kokkinos. 2019. Holopose: Holistic 3d human reconstruction in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10884–10894.
- [11] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7297–7306.
- [12] Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. 2019. GarNet: A two-stream network for fast and accurate 3D cloth draping. In *Proceedings of the IEEE International Conference on Computer Vision*. 8739–8748.
- [13] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. 2020. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5052–5063.
- [14] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. 2019. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE International Conference on Computer Vision*. 10471–10480.
- [15] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. 2018. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7543–7552.
- [16] Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics*. Springer, 492–518.
- [17] Thibaut Issenuth, Jérémie Mary, and Clément Calauzènes. 2019. End-to-End Learning of Geometric Deformations of Feature Maps for Virtual Try-On. *arXiv preprint arXiv:1906.01347* (2019).
- [18] Thibaut Issenuth, Jérémie Mary, and Clément Calauzènes. 2020. Do Not Mask What You Do Not Need to Mask: a Parser-Free Virtual Try-On. *arXiv preprint arXiv:2007.02721* (2020).
- [19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. 2015. Spatial transformer networks. In *Advances in neural information processing systems*. 2017–2025.
- [20] Sargan Jandial, Ayush Chopra, Kumar Ayush, Mayur Hemani, Balaji Krishnamurthy, and Abhijeet Halwai. 2020. SieveNet: A Unified Framework for Robust Image-Based Virtual Try-On. In *The IEEE Winter Conference on Applications of Computer Vision*. 2182–2190.
- [21] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. 2020. BCNet: Learning Body and Cloth Shape from A Single Image. *arXiv preprint arXiv:2004.00214* (2020).
- [22] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.
- [23] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4401–4410.
- [24] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [25] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. 2019. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4501–4510.
- [26] Der-Tsai Lee and Bruce J Schachter. 1980. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences* 9, 3 (1980), 219–242.
- [27] Jituo Li, Juntao Ye, Yangsheng Wang, Li Bai, and Guodong Lu. 2010. Fitting 3D garment models onto individual human models. *Computers & graphics* 34, 6 (2010), 742–755.
- [28] Kedan Li, Min Jin Chong, Jingen Liu, and David Forsyth. 2020. Toward Accurate and Realistic Virtual Try-on Through Shape Matching and Multiple Warps. *arXiv preprint arXiv:2003.10817* (2020).
- [29] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. 2019. Self-Correction for Human Parsing. *arXiv preprint arXiv:1910.09777* (2019).
- [30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)* 34, 6 (2015), 1–16.
- [31] Matur Rahman Minar, Thai Thanh Tuan, Heejune Ahn, Paul Rosin, and Yu-Kun Lai. 2020. 3D Reconstruction of Clothes using a Human Body Model and its Application to Image-based Virtual Try-On. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- [32] Matur Rahman Minar, Thai Thanh Tuan, Heejune Ahn, Paul Rosin, and Yu-Kun Lai. 2020. CP-VTON+: Clothing Shape and Texture Preserving Image-Based Virtual Try-On. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- [33] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. 2020. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7365–7375.
- [34] Amir Hossein Raffee and Michael Sollami. 2020. GarmentGAN: Photo-realistic Adversarial Fashion Transfer. *arXiv preprint arXiv:2003.01894* (2020).
- [35] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. 2020. Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7690–7699.
- [36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
- [37] Debapriya Roy, Sanchayan Santra, and Bhabatosh Chanda. 2020. LGVTON: A Landmark Guided Approach to Virtual Try-On. *arXiv preprint arXiv:2004.00562* (2020).
- [38] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*. 2304–2314.
- [39] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 84–93.
- [40] Igor Santesteban, Miguel A Otaduy, and Dan Casas. 2019. Learning-Based Animation of Clothing for Virtual Try-On. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 355–366.
- [41] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [42] Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, Vol. 4. 109–116.
- [43] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. 2019. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514* (2019).
- [44] Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.* 30, 4 (2011), 90.
- [45] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [46] Raquel Vidas, Igor Santesteban, Elena Garces, and Dan Casas. 2020. Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. *arXiv preprint arXiv:2009.04592* (2020).
- [47] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. 2018. Toward characteristic-preserving image-based virtual try-on network. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 589–604.
- [48] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 52–67.
- [49] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. 2019. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE International*

- Conference on Computer Vision*. 1042–1051.
- [50] Simon N Wood. 2003. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65, 1 (2003), 95–114.
 - [51] Zhonghua Wu, Guosheng Lin, Qingyi Tao, and Jianfei Cai. 2019. M2e-try on net: Fashion from model to everyone. In *Proceedings of the 27th ACM International Conference on Multimedia*. 293–301.
 - [52] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. 2019. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE International Conference on Computer Vision*. 7760–7770.
 - [53] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. 2020. Towards Photo-Realistic Virtual Try-On by Adaptively Generating-Preserving Image Content. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7850–7859.
 - [54] Gokhan Yildirim, Nikolay Jetchev, Roland Vollgraf, and Urs Bergmann. 2019. Generating high-resolution fashion model images wearing custom outfits. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 0–0.
 - [55] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. 2019. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In *Proceedings of the IEEE International Conference on Computer Vision*. 10511–10520.
 - [56] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. 2020. Deep Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images. *arXiv preprint arXiv:2003.12753* (2020).

Synthesizing Indoor Scene Layouts in Complicated Architecture Using Dynamic Convolution Networks

HAO JIANG, SIQI WANG, and HUIKUN BI*,

Institute of Computing Technology, Chinese Academy of Sciences,

University of Chinese Academy of Sciences,

Beijing Key Laboratory of Mobile Computing and Pervasive Device, China

XIAOLEI LV, BINQIANG ZHAO, and ZHENG WANG, Alibaba Group, China

ZHAOQI WANG, Institute of Computing Technology, Chinese Academy of Sciences, China

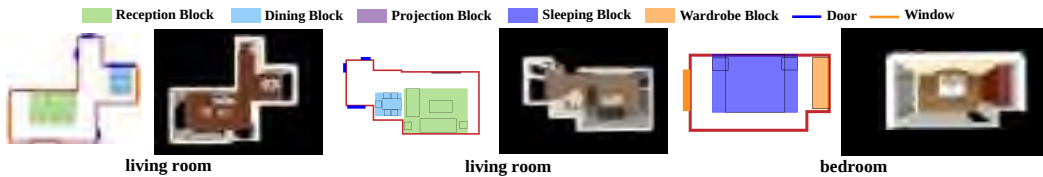


Fig. 1. We present a framework to synthesize indoor scene layouts using dynamic convolution networks, enabling us to easily generate plausible 3D indoor scenes in irregular-shaped architecture rooms. The indoor objects are grouped and then arranged by functional blocks. Here, we present two indoor scenes from the top-down view of the living rooms and one of the bedrooms. In each group, the left one shows 2D functional blocks, and the right one is the corresponding 3D scene.

Synthesizing indoor scene layouts is challenging and critical, especially for digital design and gaming entertainment. Although there has been significant research on the indoor layout synthesis of rectangular-shaped or L-shaped architecture, there is little known about synthesizing plausible layouts for more complicated indoor architecture with both geometric and semantic information of indoor architecture being fully considered. In this paper, we propose an effective and novel framework to synthesize plausible indoor layouts in various and complicated architecture. The given indoor architecture is first encoded to our proposed representation, called InAIR, based on its geometric and semantic information. The indoor objects are grouped and then arranged by functional blocks, represented by oriented bounding boxes, using dynamic convolution networks based on their functionality and human activities. Through comparisons with other approaches as well as comparative user

*Corresponding author.

Authors' addresses: Hao Jiang, jianghao@ict.ac.cn; Siqi Wang, 18829026208@163.com; Huikun Bi, bihuikun@ict.ac.cn, Institute of Computing Technology, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing Key Laboratory of Mobile Computing and Pervasive Device, No.6 Kexueyuan South Rd, Haidian, Beijing, China, 100190; Xiaolei Lv, yanjun.lxl@alibaba-inc.com; Binqiang Zhao, binqiang.zhao@alibaba-inc.com; Zheng Wang, lanjing.wz@alibaba-inc.com, Alibaba Group, Beijing, China; Zhaoqi Wang, zqwang@ict.ac.cn, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Rd, Haidian, Beijing, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

2577-6193/2021/5-ART \$15.00

<https://doi.org/10.1145/3451267>

studies, we find that our generated indoor scene layouts for diverse, complicated indoor architecture are visually indistinguishable, which reach state-of-the-art performance.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; • **Shape analysis**;

Additional Key Words and Phrases: indoor layouts, 3D indoor scene generation, convolution networks

ACM Reference Format:

Hao Jiang, Siqi Wang, Huikun Bi, Xiaolei Lv, Binqiang Zhao, Zheng Wang, and Zhaoqi Wang. 2021. Synthesizing Indoor Scene Layouts in Complicated Architecture Using Dynamic Convolution Networks. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 1 (May 2021), 16 pages. <https://doi.org/10.1145/3451267>

1 INTRODUCTION

Modeling and synthesis of structured, real, and informative indoor scenes has become a challenging and essential task, especially for computational design, gaming entertainment, VR&AR applications, and robotic navigation training in virtual environments [Dai et al. 2018; Lai et al. 2014; Sünderhauf et al. 2017; Taira et al. 2018]. Since we spend significant amount of time indoors, the indoor scene layouts continue to attract the attention of interior design companies (e.g., IKEA, HOME-STYLER) [Hom [n.d.]; IKE [n.d.]; pla [n.d.]], and researchers in the field of computer graphics and vision [Li et al. 2019; Qi et al. 2018; Ritchie et al. 2019; Wang et al. 2019; Zhang et al. 2018]. The goal of indoor scene layout modeling and synthesis is to automatically, quickly, and realistically reconstruct virtual 3D indoor scenes, including different types of rooms (e.g., living room, dining room, bedroom), based on our daily habits and activities.

Research has recently emerged to synthesize indoor scenes, which aims at generating plausible arrangements of a set of objects. Although there are many effective approaches in some architectures, it is challenging to generate an indoor layout in complicated architecture, since both the human activities and the geometric and semantic information of indoor architecture are not comprehensively considered.

1). *Geometric and semantic information of indoor architecture*

Many previous works of indoor layout modeling and synthesis [Li et al. 2019; Ritchie et al. 2019; Wang et al. 2019] used SUNCG [Song et al. 2017], a comprehensive dataset of realistic indoor scenes encompassing scene types (e.g., living room, bedroom, kitchen, office). The majority of these top-down view rooms, are rectangular-shaped or L-shaped (Fig. 2). However, in reality, indoor scenes tend to have more complicated and diverse architecture (Fig. 2(d)). Such simplifications of the rectangular-shaped or L-shaped architecture for indoor scenes cannot satisfy various and realistic human needs in the real world. For more complicated indoor architecture, the implied geometric and semantic information might have a direct impact on indoor layouts. For example, a sectional sofa usually is placed against a wall whose length is longer than the sofa. Also, the indoor lighting caused by windows on the wall will influence the indoor layouts.

2). *Human activities*

State-of-the-art methods tend to arrange indoor objects one after another iteratively [Ritchie et al. 2019; Wang et al. 2018]. However, people are more accustomed to roughly dividing the room into several blocks based on their activities and indoor architecture, such as a reception block, a projection block, and a dining block as illustrated in Fig. 2. After that, a group of objects will be placed in each block. Thus, the strategy in these works, which treating objects as individuals, cannot utilize the geometric and semantic information of indoor architecture sufficiently. Some pioneering works [Fisher et al. 2015; Merrell et al. 2011] grouped objects with the specific functionality and proposed to utilize indoor structured blocks to facilitate specific human activities. To describe semantic relations among objects, some works recently exploited the spatial-based relations to organize functional groups of objects [Li et al. 2019; Wang et al. 2019]. There are only some specific

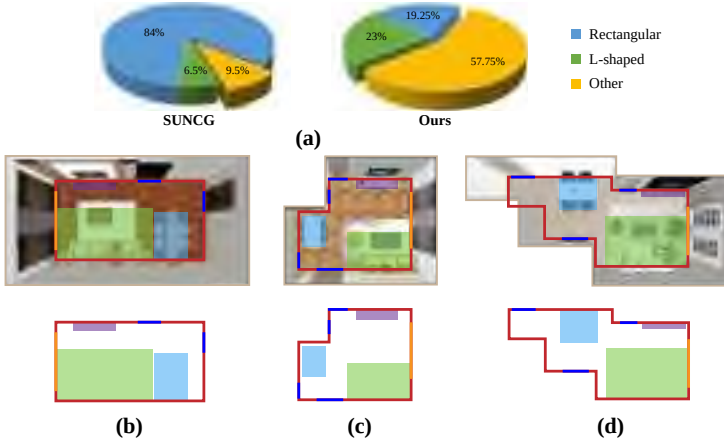


Fig. 2. Illustration of the statistical indoor architecture in SUNCG and our dataset. We randomly selected 400 indoor scenes from SUNCG and our dataset, respectively. The representative indoor architecture from SUNCG is (b) rectangular-shaped or (c) L-shaped. However, indoor scenes in reality tend to have more complicated and diverse architecture, shown in (d). The purple, green, and blue blocks refer to the projection block, reception block, and dining block, respectively. The yellow and blue lines on the red outline of each room represent the windows and doors, respectively.

indoor architectures and human activities being considered in the prior works, which are not sufficient for all indoor scenes.

To address these problems, we build an effective and novel framework to generate plausible indoor scene layouts. We assume that any indoor scene with complicated architecture is referred to as an irregular top-down view polygon. Given an indoor architecture around with a varied number of walls, we first encode the geometric and semantic information of indoor architecture into our proposed representation, called InAiR. Indoor objects to be arranged are organized by functional blocks, represented with oriented bounding boxes (OBBs), based on their functionality, human activities, and spatial relations. Each functional block here is assumed to be relative to a wall, called anchor-relative wall. We then employ dynamic convolution networks to predict the anchor-relative wall for given functional blocks, based on the captured multi-level geometric and semantic features of the indoor architecture. Then, the detailed layout of the functional block will be computed based on the selected anchor-relative wall. The final indoor layout will be generated after replacing each functional block OBB with a group of 3D shape objects.

The main contributions of this work include: i) we propose an intuitive, structured indoor architecture representation, called InAiR, to extract geometric and semantic information of realistic and complicated indoor architecture quickly and automatically. To our best knowledge, we are the first to encode the geometric and semantic information of complicated indoor architecture, not limited to rectangular-shaped or L-shaped architecture, to generate indoor scene layouts. ii) An effective and novel framework is built to synthesize indoor layouts of complicated architecture, which is satisfied with human habits and activities. Using dynamic convolution networks, the generated layout can jointly consider multi-level features. Through comparisons with state-of-the-art methods, our model achieves superior performance, especially in various challenging realistic indoor scenes.

2 RELATED WORK

There has been significant research on the indoor scene layout synthesis. Prior works synthesized indoor scene layout mainly focuses on modeling object-object relationships, as well as the occurrence and arrangement of objects within a room. Early work adopted rule-based constraints [Xu et al. 2002] and optimization of the cost functions based on design principles [Merrell et al. 2011; Yu et al. 2011] to model the object-object relationships. Gaussian mixtures [Fisher et al. 2015], relations annotations [Fu et al. 2017], and graphical models [Henderson et al. 2017] are frequently used to introduce the spatial relations of a group of objects.

With the availability of 3D indoor scene dataset (e.g., SUNCG [Song et al. 2017]), recent works on indoor scene synthesis proposed learning-based methods. Zhang et al. utilized a human-centric probabilistic grammar model to synthesize 3D room layouts [Qi et al. 2018]. Zhang et al. built a generative model using a feed-forward neural network that maps a prior distribution to the distribution of primary objects in indoor scenes for indoor environments synthesis [Zhang et al. 2018]. Other research showed the effectiveness of image-based deep convolutional generative models on indoor scenes generation [Ritchie et al. 2019; Wang et al. 2018]. The generative recursive autoencoders, called GRAINS, were designed to perform indoor objects grouping in the encoding and decoding phase [Li et al. 2019]. Combining a high-level relation graph representation with spatial prior neural networks, Wang et al. proposed a novel “plan-and-instantiate” conceptual framework for layout generation [Wang et al. 2019]. However, there has been little focus on synthesizing indoor scenes, especially generating layouts with more complicated architectures, which meet human habits and activities.

3 OVERVIEW

In this work, our goal is to build an effective and robust framework to generate plausible indoor scene layouts, which is satisfied with diverse architecture and human living habits. A room in our work refers to a completely enclosed space using walls, doors, and windows. Therefore, the outline of a top-down view room may be an irregular polygon. Considering the functionality and practicability of room types, we limited focus on living rooms, dining rooms, and bedrooms in this paper. We note that living rooms and dining rooms usually are interconnected spaces in ground truth indoor scenes, and such rooms are also considered in our work. Furthermore, our framework is easily extended to more room types.

Given an indoor scene and a set of objects’ labels, our framework is decomposed into four steps. Fig. 3 illustrates the pipeline of our framework to synthesize indoor scene layouts in complicated architecture.

Indoor Architecture Representation. To describe the complicated indoor architecture sufficiently, we first encode the given indoor scene with complicated architecture to a specific representation InAiR (see Sec. 4.1). All the geometric and semantic information of dynamic walls in architecture can be captured.

Assign Functional Blocks and Corresponding Objects. In order to describe the functionality and spatial relations of objects, we propose to use a functional block to organize a group of indoor labeled objects (see Sec. 4.2). Each functional block is represented by an OBB and structured to facilitate specific human activities, based on the statistics and analysis of indoor scene layouts. For example, a living room typically includes a reception block and a projection block, a dining room includes a dining block, a bedroom includes a sleeping block and a wardrobe block, etc.

We enable users to design several functional blocks to be arranged based on their intents and select the corresponding objects’ labels in each functional block. It is noted that all the indoor objects are

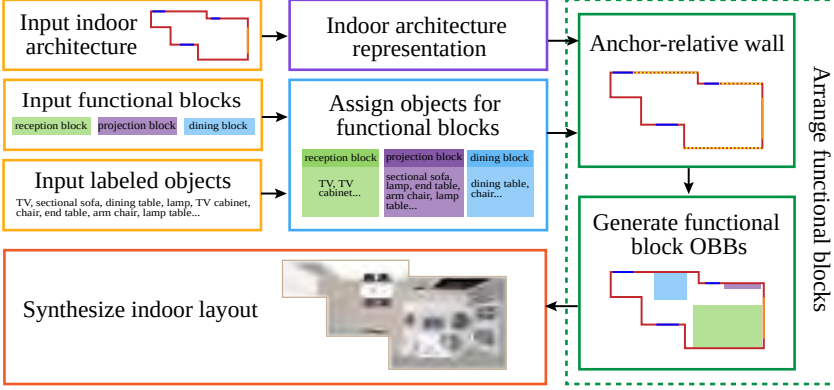


Fig. 3. Overview of our framework. Given an indoor architecture, several functional blocks, a group of labeled objects, our method encoded the indoor scene into a specific representation and organizes the objects with functional blocks based on human activities. Considering the rich geometric and semantic information of indoor architectures, our method proposes to exploit local and contextual features to synthesize indoor layouts in complicated architectures.

arranged by functional blocks. We build the corresponding functional block database, where each functional block organizes a group of 3D objects involved in certain spatial relations.

Arrange Functional Blocks. We assume that the location of each functional block is relative to a wall, defined as the anchor-relative wall. All the anchor-relative walls for functional blocks are computed simultaneously (see Sec. 4.2). We operate dynamic convolution networks on architecture representation to capture multi-level geometric and semantic features of room architecture, which is represented with an irregular polygon. The anchor-relative walls contribute to locating functional blocks roughly. Then, the extracted multi-level features of room architecture, as well as the embedded representation of the selected anchor-relative wall, are concatenated to compute the details of functional block OBBs, such as relative positions, size, offset, and orientations (see Sec. 4.2). The functional block OBBs are then arranged one after another.

Synthesize Indoor Layouts. To finalize the indoor scene layout, the generated functional block OBBs are replaced with a 3D functional block retrieved from our functional block database based on the functional block category including objects, and size information of the functional blocks.

4 METHODOLOGY

In this section, we describe our indoor architecture representation and present the framework for synthesizing indoor scene layouts in complicated architecture. The main symbols and corresponding explanations are shown in Table 1.

4.1 Indoor Architecture Representation InAiR

The rooms' architecture is a common condition that has a considerable impact on the overall indoor layouts. The indoor architecture representation, named *InAiR*, is specially proposed for a room with around N irregular walls (including the main wall, and partition wall, etc.). Each wall has both geometric and semantic information.

Each top-down view room is referred to as an irregular polygon, whose outline represents the walls of the room. For any wall $i, (i \in [1, N])$ of room τ , its geometric information includes length l^i

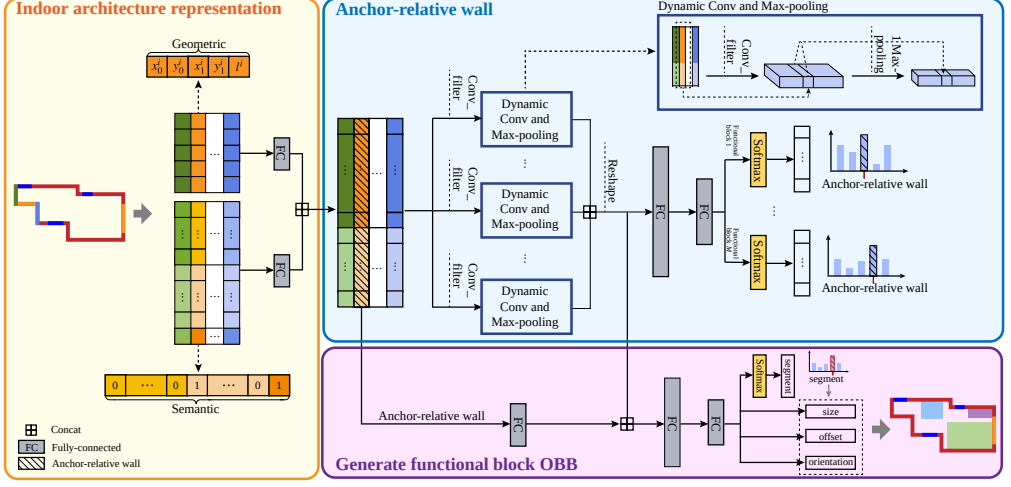


Fig. 4. Architecture of our network to generate functional block OBBs. The geometric and semantic information of the indoor architecture is extracted, encoded, and concatenated, shown in the yellow shaded region, which is then fed into the following anchor-relative wall module. We predict the anchor-relative wall for the corresponding functional block to be arranged, shown in the blue shaded region. Using the encoded information of the selected anchor-relative wall and the embedded geometric and semantic information of the indoor architecture, we predict the corresponding relative position, size, offset, and orientation of the functional block, shown in the purple shaped region.

Symbol	Explanation
τ	Room
N	The number of walls.
e_g^i	The geometric information of wall i .
e_s^i	The semantics of wall i .
e^i	The geometric and semantic embedding of wall i .
M	The number of functional blocks.
E^τ	The embedded architecture of room τ .
C^τ	The extracted features of room τ after filters and max-pooling.
γ	Functional block
I^γ	The selected anchor-relative wall for the functional block γ .
O^γ	The corresponding OBB for the functional block γ .

Table 1. The main symbols and corresponding explanations.

and positions, referring to 5 real numbers. It is noted that the wall with a door will be divided into two walls, as shown in Fig. 4. The position is represented with its endpoints (x_0^i, y_0^i) and (x_1^i, y_1^i) . The semantic information is represented with 13 binary indicators (4 bits for the type of walls; 8 bits for the type of connected rooms; 1 bit for windows on the wall). Namely, representation information for each wall is stored in an 18-dimensional vector. Therefore, the architecture of this room is represented with $X^\tau \in \mathbb{R}^{18 \times N}$.

Room	Functional block	Object	Illustration of the representative objects of each functional block				
Living and dining	Reception	sofa (sectional sofa, two seater sofa...), arm chair, swivel chair, end table, lamp, side table, accessory (plants, rug, cup, pillow...),...					
	Projection	TV, TV cabinet, accessory (plants, vases ...),...					
	Dining	dining table (round, square, ...), chair, armchair, accessory (plates, bowls, ...),...					
Bedroom	Sleeping	bed (king size, queen size, ...), night stand, lamp, accessory (vases, pillows, ...),...					
	Wardrobe	wardrobe, (L-shaped, corner,...)					

Table 2. The functional blocks and some corresponding objects. Based on the statistical results of our dataset, we show that there are three main functional blocks in the living and dining room. They are reception blocks, dining blocks, and projection blocks. The representative functional blocks in bedrooms are sleeping blocks and wardrobe blocks.

4.2 Functional Block Module

Indoor scenes often contain functional groups of objects, which are structured to facilitate specific human activities. In our work, we refer these grouped objects to *functional blocks*, represented with OBBs. Based on the statistical results of our dataset and room types, we show all the functional blocks appeared in our synthetic indoor scene and some corresponding objects in Table 2. Given an input of an indoor scene and a set of objects’ labels, we encourage users to design several functional blocks and assign the corresponding objects to each functional block based on their intents and activities. All the indoor objects are arranged by functional blocks. We assume that there are M functional blocks to be arranged in the room τ .

A functional block is placed based on its relative position, size, offset, and orientation information. We assume each functional block is relative to a wall, named anchor-relative wall. For example, a projection block including TV and a reception block with sofa in the living room are likely to be placed adjacent to a wall; all beds in sleeping blocks have at least one side leaning against a wall, etc.

Based on our proposed indoor architecture representation InAiR, we exploit dynamic convolution networks to predict anchor-relative walls for all the functional blocks simultaneously. Next, the functional block OBBs, including detailed position, size, offset, and orientation information relative to the respective predicted anchor-relative wall, are generated. Finally, each designed functional

block OBB is replaced with a corresponding 3D functional block retrieved from our functional block database based on the functional block category including objects, and size information of the functional blocks.

We exploit a hierarchical clustering method to create functional blocks other than grouping objects manually. Based on the definition of functional blocks, a group of objects are clustered into a functional block according to their spatial relations and human activities with respect to these objects. We use linkage criteria to get the smallest cluster distance among these objects. The clustering process will not stop until the cluster's diameter of a cluster exceeds the pre-defined threshold.

Anchor-relative Wall. To deal with the effect of indoor architecture, we arrange each functional block based on the combination of the geometric information of each wall and the contextual semantic information of a set of sequential walls. For example, a long sofa bed fails to be placed against a wall with length that is shorter than the length of the sofa, and the arrangement of a corner sofa relies on the contextual semantic information of the two adjacent walls. We divided the features of the indoor architecture into two categories: local features and contextual features. We utilized dynamic convolution networks to capture the local features and contextual features of the indoor architecture. The dynamic convolution operation is frequently used in natural language processing (NLP) to induce lexical-level automatically and sentence-level features from plain texts for event extraction [Baevski and Auli 2018; Chen et al. 2015; Kim 2014; Kim et al. 2016; Zhang and Wallace 2015]. Inspired by these works, we employ multiple dynamic convolution networks to obtain valuable multi-level features within architecture.

We used five variables to describe the geometric information of each wall in the room, which contains more diverse latent information than the semantic binary indicators. Given the architecture representation $X^\tau \in R^{18 \times N}$ of room τ , the geometric information of wall i are first embedded into a high-dimensional space e_g^i . The semantics of wall i is transformed into a low-dimensional embedding e_s^i . The geometric and semantic embeddings are then concatenated as the input e^i of the following modules, as shown in Fig. 4. The embedded architecture of room τ is E^τ with shape of $h \times N$.

The convolution layer aims to capture multi-level compositional semantics of a group of walls and compress this valuable information into feature maps. Here, a convolution operation involves n filters (size of $h \times w$), which is applied to a window of w walls to extract a new semantic feature¹. Each filter is operated on the whole information of several geometrically consecutive walls. The shape of the feature after every n convolution filters with a fixed w for room τ is $1 \times N \times n$. In order to consider different level semantics of architecture, we utilize a set of filters with a window size $w \in \{f_1, f_2, \dots, f_m\}$ for convolution.

As the shape of the feature after convolution is varied with the number of walls (N) in τ . We then take a max-pooling to extract the most valuable information from N walls. Using the max-pooling, the features of the varied architecture in each dynamic convolution network are transformed into a $1 \times 1 \times n$ feature. Therefore, the concatenated features after filters with size $w \in \{f_1, f_2, \dots, f_m\}$ and max-pooling can represent the multi-level geometric and semantic features of the architecture. We reshape it into a vector with a shape of $mn \times 1$, denoted by C^τ . The dynamic geometric and semantic information of indoor architecture is turned into fixed dimensions.

For the functional block γ ($\gamma \in 1, M$) to be arranged in room τ , we feed the vector after two fully connected layers into a softmax layer. We compute the probabilities of each i to be the anchor-relative

¹In order not to lose the information on the edge of E^τ , for any wall i (when $i < \lceil \frac{w}{2} \rceil$), we sequentially pad the embedded information of the adjacent $\lceil \frac{w}{2} \rceil - i$ walls for wall i . Similarly, when $i > N - \lceil \frac{w}{2} \rceil$, the embedded information of the adjacent $\lceil \frac{w}{2} \rceil - i - N$ walls for wall i will be padded for subsequent wall N .

wall. To predict the anchor-relative wall for the functional block γ in room τ , we employ the cross-entropy loss: $L_{anc} = \sum_{\gamma=1}^M \sum_{i=1}^N (p_{anc}^{i,\gamma} - \hat{p}_{anc}^{i,\gamma})^2$, where $p_{anc}^{i,\gamma}$ is the ground truth, and $\hat{p}_{anc}^{i,\gamma}$ is predicted probability of wall i that is the anchor-relative wall for the functional block γ .

Generate Functional Block OBBs. We use OBBs to represent functional blocks. Based on the selected anchor-relative wall I^γ for the functional block γ , we compute the relative position, size, offset, and orientation information of the functional block γ to place the corresponding OBB O^γ .

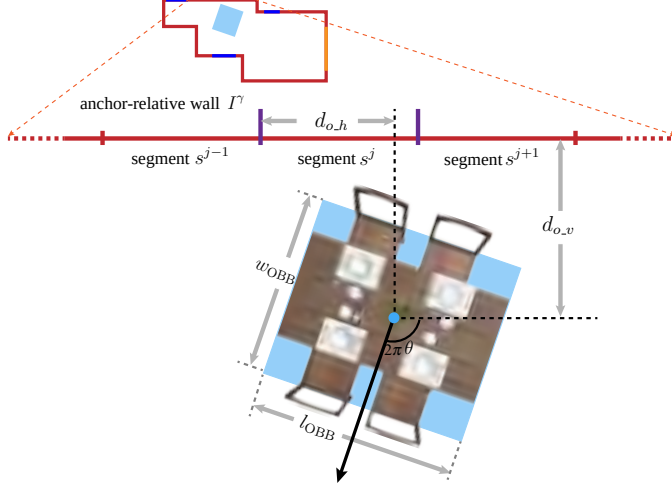


Fig. 5. Illustration of a functional block OBB respect to the predicted anchor-relative wall. The dining block, used as an example, comprises four chairs and a dining table.

The arrangement of O^γ respect to I^γ needs to take the overall architecture of τ into consideration, as well as the local information of I^γ . Therefore, the embedding of I^γ with semantic and geometric information, which is denoted by e^{I^γ} , is passed into a fully connected layer, and then concatenated with C^τ , which carries the overall architecture semantics of τ . After two fully connected layers, the vector is passed to the output layer.

To arrange O^γ more accurately, we divided the anchor-relative wall I^γ into t segments, shown in Fig. 5. We predict each probability of O^γ in segment s^j with $j \in [1, t]$, respectively, as well as the size, offset, and orientation of O^γ in the segment with the highest probability. We described an OBB using the width w_{OBB} and the length l_{OBB} . We introduce the offset with two offset values—one (i.e., $d_{o,h}$) in horizontal direction, and the other (i.e., $d_{o,v}$) in vertical direction, as shown in Fig. 5. The orientation is encoded using a 5-bit one-hot vector and a parameter $\theta \in [0, 1]$ (that is a real number). The first 5 bits are designed to indicate whether the functional block OBB is oriented at an angle of $0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ —respect to the anchor-relative wall I^γ —or none of the above, where p_{ori}^k with $k \in [1, 5]$ denotes each bit. In the oriented case (i.e., $k = 5$), the functional block will be rotated $2\pi\theta$ (unit: rad) about the center (Fig. 5).

There are four loss terms when generating functional block OBBs. We noted that the variables with a hat denote predicted values, and that without a hat denote ground truth. The losses for size, offset, and orientation of O^γ only focus on the OBB in the segment with the highest probability. We predefined the function $\Phi(a, b) = (\sqrt{|a|} - \sqrt{|b|})^2$.

- We use cross-entropy loss to show the probabilities in all segments: $L_p = \sum_{j=1}^t (p_{seg}^j - \hat{p}_{seg}^j)^2$, where $\hat{p}_{seg}^j$ is the predicted probability of O^γ located in segment s^j .

- Loss of the OBB's size: $L_{size} = \sum_{j=1}^t \mathbb{1}_j^{obj} (\Phi(w_{OBB}^j, \hat{w}_{OBB}^j) + \alpha \Phi(l_{OBB}^j, \hat{l}_{OBB}^j))$, where $\mathbb{1}_j^{obj}$ denotes the functional block OBB O^j appears in segment s^j .
- Loss of the OBB's offset: $L_{offset} = \sum_{j=1}^t \mathbb{1}_j^{obj} (\Phi(d_{o_h}^j, \hat{d}_{o_h}^j) + \beta \Phi(d_{o_v}^j, \hat{d}_{o_v}^j))$.
- Loss of the OBB's orientation is the sum of a cross-entropy loss and a reconstruction loss: $L_{ori} = \sum_{j=1}^t \mathbb{1}_j^{obj} (\sum_{k=1}^5 (p_{ori}^{j,k} - \hat{p}_{ori}^{j,k})^2) + \eta \sum_{j=1}^t \mathbb{1}_j^{obj} \Phi(\theta^j, \hat{\theta}^j)$.

Finally, the total loss of the network for generating each functional block OBB O^j respect to the corresponding selected anchor-relative wall I^j is defined as follows:

$$L_{total} = L_p + \mu_1 L_{size} + \mu_2 L_{offset} + \mu_3 L_{ori}, \quad (1)$$

where μ_1 , μ_2 , and μ_3 are the balance controllers.

4.3 Synthesize Indoor Layouts

We used the function block OBB's given size information to iteratively detect the existence of collision between the functional block OBB and the existing functional block OBBs. Our collision detection involves both broad phase and narrow phase. We begin by the broad phase, which used AABB (Axis-Aligned Bounding Box) to detect any possible collision. If a collision exists, SAT (Separating Axis Theorem) is then used for further detection [Bergen 2003]. When the collision had been detected, we would resample another anchor-relative wall and OBB's information (size, offset, and orientation). After arranging all the functional block OBBs, we prepared some 3D functional blocks as candidates using the functional block category including objects via a nearest neighbor search.

5 EXPERIMENT

In this section, we compared our method with the state-of-the-art methods and present quantitative and qualitative evaluation results. In our work, we trained our model on the 3D-Front dataset². We used 5000 epochs to train the anchor-relative wall module and 3000 epochs to train the functional block OBB module. And we chose $\mu_1 = 0.5$, $\mu_2 = 0.5$, and $\mu_3 = 0.3$ for (1). We summarize the additional details of the network in Table 3. The model was trained on a server with two P100 graphics cards and a 16-core CPU @2.50GHz. The training process took about 10 hours.

5.1 Results of Functional Blocks

Fig. 6 shows the predicted results of the anchor-relative wall, when arranging a reception block in a living room with distinct irregular-shaped architecture. Qualitatively, our method does a good job of predicting the anchor-relative wall, where the reception function area is located against all walls in various real complicated architecture. We note that it is essential and necessary for arranging the reception blocks with a specific size.

Fig. 7 shows the predicted results of the reception block in living rooms with various indoor architecture. We found that the location and size of our predicted functional blocks are plausible. We noted that in the cases with incorrect labels caused by manual labeling, our method still can generate robust and reasonable layout results, as shown in Figs. 7(d), 7(g), and 7(h).

Fig. 8 shows the comparison of the 2D layout of two living-dining rooms, which involve dining blocks, reception blocks, and projection blocks to be arranged. Each comparison is shown side by side: the left one is our synthesized layouts, and the right one is the one designed by designers. We found that the significant factors—architecture, semantics, and human activities—have been taking into accounts in our synthesized layouts, which is consistent with the intention and ideas of designers.

²The information of 3D-Front dataset can be found online at <https://tianchi.aliyun.com/specials/promotion/alibaba-3d-scene-dataset>

Layer Type	Input (dimensions)	Output (dimensions)	Additional Parameters
Indoor Architecture Representation			
Fully-connected	$X_{\text{geometric}}^{\tau}, 5 \times N$	$E_g^{\tau}, 48 \times N$	-
Fully-connected	$X_{\text{semantic}}^{\tau}, 13 \times N$	$E_s^{\tau}, 8 \times N$	-
Concat.	$E_g^{\tau}, 48 \times N, E_s^{\tau}, 8 \times N$	$E^{\tau}, 56 \times N$	-
Dynamic Convolution and Max-pooling			
(The parameters of only one dynamic convolution and max-pooling module are shown here. The filter size of each convolution is different, referred to f_m . We separately employ $m = 9, f_m \in \{1, 3, 5, \dots, 15, 17\}$ modules to extract features.)			
Convolution	$E^{\tau}, 56 \times N$	$\text{conv}_m, 1 \times N \times 200$	act:=ReLU, kernel:= $(56, f_m)$, $f_m \in \{1, 3, 5, \dots, 15, 17\}$, stride:= $(1, 1)$, filter number:=200
1 Max-pooling	$\text{conv}_m, 1 \times N \times 200$	$\text{pool}_m, 1 \times 1 \times 200$	-
Concat.	$\text{pool}_m, 1 \times 1 \times 200, m = 1, \dots, 9$	$\text{pool}_m, 1 \times 9 \times 200$	-
Reshape	$\text{pool}_m, 1 \times 9 \times 200$	$C^{\tau}, 1800 \times 1$	-
Anchor-relative Wall			
Fully-connected	$C^{\tau}, 1800 \times 1$	$fc_1^{\tau}, 900 \times 1$	act:=ReLU
Fully-connected	$fc_1^{\tau}, 900 \times 1$	$fc_2^{\tau}, n_r * M \times 1$	act:=ReLU, for living and dining room, $n_r = 36$; for bedroom, $n_r = 18$
Softmax	$fc_2^{\tau}, n_r * M \times 1$	$p_{\text{anc}}, n_r * M \times 1$	-
Functional Block OBBs			
Fully-connected	$e^{f_j}, 56 \times 1$	$fc_1^j, 200 \times 1$	act:=ReLU
Concat.	$C^{\tau}, 1800 \times 1, fc_1^j, 200 \times 1$	$fc_2^j, 2000 \times 1$	-
Fully-connected	$fc_2^j, 2000 \times 1$	$fc_3^j, 1000 \times 1$	act:=ReLU
Fully-connected	$fc_3^j, 1000 \times 1$	$fc_4^j, 110 \times 1$	act:=ReLU
Fully-connected & Softmax	$fc_4^j, 110 \times 1$	$p_{\text{seg}}, t \times 1$	$t = 10$
Fully-connected	$fc_4^j, 110 \times 1$	$OBB_{\text{size}}, 2t \times 1$	-
Fully-connected	$fc_4^j, 110 \times 1$	$OBB_{\text{of fset}}, 2t \times 1$	-
Fully-connected & Softmax	$fc_4^j, 110 \times 1$	$OBB_{\text{ori}}, 5t \times 1$	-
Fully-connected	$fc_4^j, 110 \times 1$	$OBB_{\theta}, t \times 1$	-

Table 3. Detailed architecture of our network.

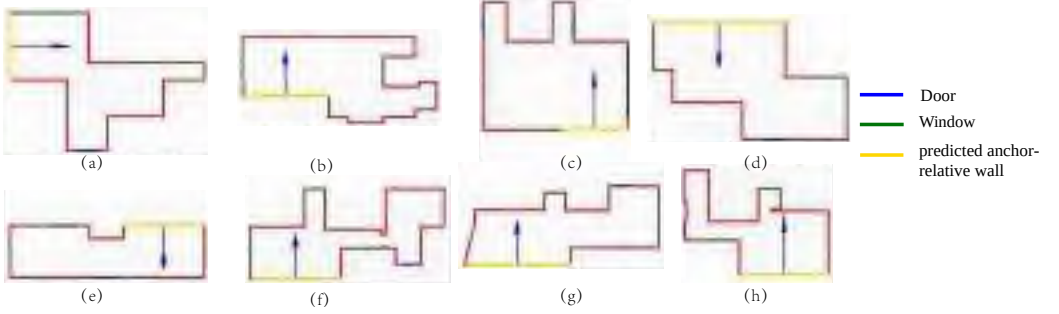


Fig. 6. The illustration of the predicted anchor-relative walls for arranging the reception blocks with our method. The red lines represent the wall. The yellow lines indicate the predicted anchor-relative wall and the blue arrows indicate the center and orientation of the corresponding anchor-relative walls. We note that our method can accurately predict anchor-relative walls in quite complicated architectures, which is consistent with the ground truth.

5.2 Synthesizing New Indoor Layouts

We compare our method with Fast & Flexible method [Ritchie et al. 2019] and present some synthetic indoor layouts for living and dining rooms and bedrooms in Fig. 9. As compared in the top three

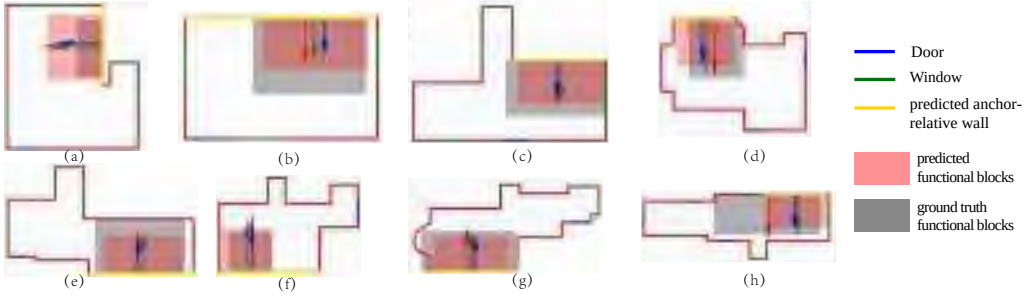


Fig. 7. Illustration of the predicted reception blocks with our method. The gray rectangles represent the manually labeled locations of the reception blocks, and the red arrows represent where the labeled blocks are pointing to each of the corresponding anchor-relative walls. The red rectangles represent the predicted results by our method. The green arrows and the blue arrows indicate the predicted top two segments to arrange blocks.

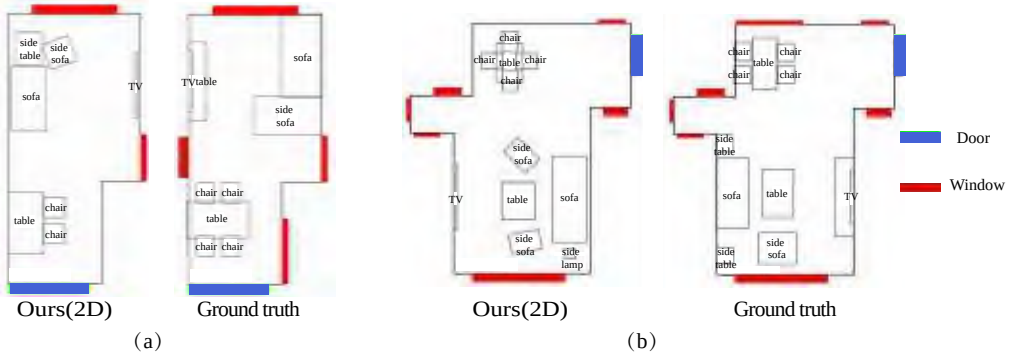


Fig. 8. Comparisons of our 2D synthesized indoor layouts and the ground truth. There are dining blocks, reception blocks, and projection blocks to be arranged. In each comparison, the left one is our synthesized layouts and the right one is the one designed by designers.

rows in Fig. 9, Fast & Flexible method fails to generate plausible indoor layouts of complicated architecture. By an iterative strategy to arrange objects one after another, Fast & Flexible was unable to compute reasonable arrangement of some given objects (Fig. 9(a)). Some objects are placed in the center of the room (Fig. 9(c)), neglecting spatial relations. Using multi dynamic convolution networks, our proposed method can capture multi-level features of the indoor architecture. Our synthetic indoor layouts are more plausible and reasonable for indoor scenes with complicated architecture. More synthesized 2D and 3D indoor layouts in Fig. 10 and Fig. 11.

5.3 Paired Comparison User Study

We designed a paired comparison user study to evaluate the synthetic indoor layouts using our method. We randomly selected 58 scenes, including 40 living & dining room, and 18 bedrooms. We generated top-down view images for this user study which combined with the other two groups of scenes, respectively. This yield 2×58 comparison pairs (ours vs. ground truth and ours vs. fast & flexible, respectively), scenes from one pair are based on the same outline.

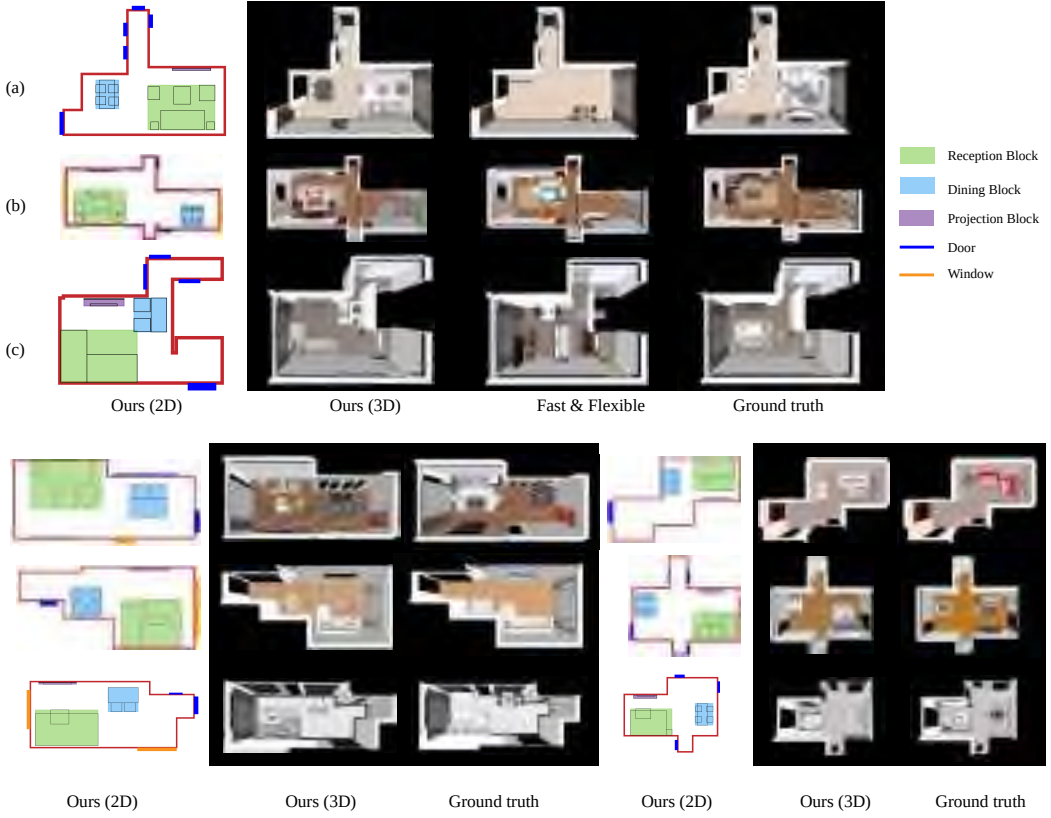


Fig. 9. Comparisons of synthesized indoor layouts. Our 2D, 3D generated results are compared with Fast & Flexible [Ritchie et al. 2019] method and ground truth in the top three rows. The bottom three rows illustrate additional compared examples.

Room type	Ours vs.	
	Fast & Flexible	Ground truth
Living and dining	65.00 ± 13.51	41.00 ± 8.38
Bedroom	63.89 ± 14.75	41.67 ± 11.45

Table 4. The experimental results of our user study that indicates the distribution of the percentage ($\pm$ standard error) of forced-choice comparisons.

We recruited 2×10 participants for these two comparisons, who are all graduate students in the university, to participate in our paired comparison user study. Avoiding making forced and inaccurate perception votes, they are required to perceptually select which functional block and the overall layout are more plausible based on the given two top-down view rendered scenes. To avoid the effect of material or texture appearance, the objects rendered in compared scenes here are shown with solid colors. We use one-sample t-tests to determine the Confidence Interval (CI), and paired-sample

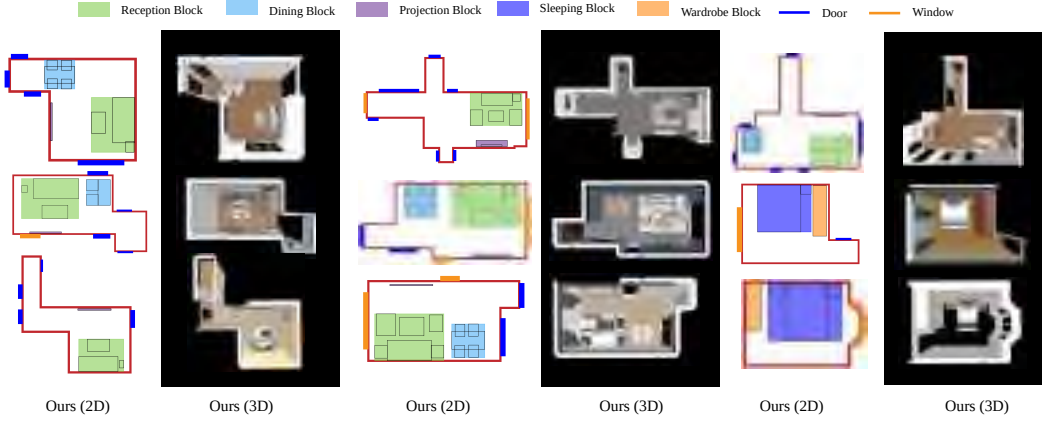


Fig. 10. Additional synthesized 2D and 3D indoor layouts.



Fig. 11. Additional synthesized 3D indoor layouts.

t-tests to compare the difference of the true mean of two sets of data with 95% confidence. The conventional significance for the entire analysis was determined at $\alpha = 0.05$, two tailed.

Table 4 shows the participants’ preference between the compared scenes. Compared with the Fast & Flexible, our results are preferred for both types of rooms. We found that our method is more robust when synthesizing indoor layouts of more complicated and challenging architecture. Comparing with the professionally designed scenes produced by layout designers, our method is less preferred instead. We suggested that this might be caused by a more accurate location of objects (e.g., beds are usually leaning up against a wall closely without any gaps), which is more visually sensitive to humans, especially in 2D top-down view plans. Producing all the accurate locations for all the objects is still challenging.

We conduct the second user study to verify our analysis. Ten participants (who have not participated the previous user study) were required to rate 40 3D indoor layouts with a random order on a scale of 1 to 5, 1 means the least plausible, and 5 means the most plausible). The layouts to be rated include 20 of our synthetic results and the corresponding ground truth, which comprise 15 living-dining rooms and 5 bedrooms. In the living and dining room, the mean of the raw scores for our method is 3.59 ± 0.79 , and the mean of the ground truth is 3.75 ± 0.72 . In the bedroom, the mean of the raw scores for our method is 4.12 ± 0.66 , and the mean of the ground truth is 4.14 ± 0.58 . The experimental results show that the proportion of ours is $48.89\% \pm 10.72\%$ in living-dining rooms and $49.88\% \pm 8.05\%$ in bedrooms. From this experiment, we found that there is no evidence of significant preference for the layout results produced by professional layout designers.

From the two user studies, we observed that: (1). Due to the human sensitivity and subtle differences, users are capable of selecting a better result in a paired comparison of the two layout results,

which makes it challenging to produce highly accurate and robust indoor layouts in complicated architectures. (2). However, the layout results using our method is preferred compared to the fast & flexible (i.e., the state-of-the-art method). (3). Additionally, when we randomly mixed our synthetic indoor scene results with the professionally designed layouts, the two types of results were given a very consistent evaluation, indicating that our synthetic results are visually consistent and highly acceptable for the participants.

6 CONCLUSION

In this paper, we focus on indoor scene layouts synthesis in complicated architecture. Unlike the previous works generating indoor layouts in rectangular or L-shaped rooms, we propose an intuitive, and structured indoor architecture representation, called InAiR, to extract geometric and semantic information of indoor architecture. The outputs of InAiR are then used in an effective and novel framework that is built to synthesize indoor layouts, where we employ dynamic convolution networks to capture local and contextual multi-level features of the indoor architecture. The objects to be arranged are organized into functional blocks, which facilitate specific human activities and habits. To our best knowledge, for the first time, we synthesized indoor layouts for complicated architecture. Through comparisons with state-of-the-art methods, our model achieves superior performance, especially in various challenging realistic indoor scenes.

In future work, adopting deep learning models with stronger representative capabilities might further improve our results, such as the dynamic aggregation convolution method proposed by Chen et al [Chen et al. 2020]. Due to the complexity of the human activities, it might also help considering both the macro-arrangement of furniture from a functional point of view and the indoor decoration style to meet human needs.

7 ACKNOWLEDGMENTS.

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2017YFB1002602, in part by the National Natural Science Foundation of China under Grant 62002345 and 61532002, and in part by the Alibaba Research Fellowship.

REFERENCES

- [n.d.]. EasyHome HomeStyler. <https://www.homestyler.com/int/>. Accessed 2019-8-11.
- [n.d.]. The IKEA Home Planner. <https://www.ikea.com>. Accessed 2019-8-11.
- [n.d.]. Planner5d. <https://planner5d.com>. Accessed 2019-8-11.
- Alexei Baevski and Michael Auli. 2018. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853* (2018).
- Gino Bergen. 2003. *Collision Detection in Interactive 3D Environments*. <https://doi.org/10.1201/9781482297997>
- Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. 2020. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11030–11039.
- Yubo Chen, Liheng Xu, Liu Kang, Daojian Zeng, and Jun Zhao. 2015. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. In *Meeting of the Association for Computational Linguistics*.
- Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. 2018. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4578–4587.
- Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric Scene Synthesis for Functional 3D Scene Modeling. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 179.
- Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. 2017. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 201.
- Paul Henderson, Kartic Subr, and Vittorio Ferrari. 2017. Automatic Generation of Constrained Furniture Layouts. *arXiv preprint arXiv:1711.10939* (2017).
- Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*.

- Yoon Kim, Yacine Jernite, David A Sontag, and Alexander M. Rush. 2016. Character-Aware Neural Language Models. In *AAAI*.
- Kevin Lai, Liefeng Bo, and Dieter Fox. 2014. Unsupervised feature learning for 3d scene labeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3050–3057.
- Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. 2019. GRAINS: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)* 38, 2 (2019), 12.
- Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive furniture layout using interior design guidelines. In *ACM transactions on graphics (TOG)*, Vol. 30. ACM, 87.
- Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song Chun Zhu. 2018. Human-centric Indoor Scene Synthesis Using Stochastic Grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5899–5908.
- Daniel Ritchie, Kai Wang, and Yu-an Lin. 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6182–6190.
- Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1746–1754.
- Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. 2017. Meaningful maps with object-oriented semantic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5079–5085.
- Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. 2018. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7199–7209.
- Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2019. PlanIT: planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 132.
- Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2018. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 70.
- Ken Xu, James Stewart, and Eugene Fiume. 2002. Constraint-based automatic placement for scene composition. In *Graphics Interface*, Vol. 2. 25–34.
- Lap-Fai Yu, Sai Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley Osher. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Trans. Graph.* 30, 4 (2011), 86.
- Ye Zhang and Byron C. Wallace. 2015. A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification. In *IJCNLP*.
- Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, and Qixing Huang. 2018. Deep Generative Modeling for Scene Synthesis via Hybrid Representations. *arXiv preprint arXiv:1808.02084* (2018).

Conditional Diffusion for Interactive Segmentation

Xi Chen, Zhiyan Zhao, Feiwu Yu, Yilei Zhang, Manni Duan*

Alibaba Group

{xizhi.cx, zhiyan.zzy, feiwu.yfw, yilei.z}@alibaba-inc.com, manyi@taobao.com

Abstract

In click-based interactive segmentation, the mask extraction process is dictated by positive/negative user clicks; however, most existing methods do not fully exploit the user cues, requiring excessive numbers of clicks for satisfactory results. We propose Conditional Diffusion Network (CDNet), which propagates labeled representations from clicks to conditioned destinations with two levels of affinities: Feature Diffusion Module (FDM) spreads features from clicks to potential target regions with global similarity; Pixel Diffusion Module (PDM) diffuses the predicted logits of clicks within locally connected regions. Thus, the information inferred by user clicks could be generalized to proper destinations. In addition, we put forward Diversified Training (DT), which reduces the optimization ambiguity caused by click simulation. With FDM, PDM and DT, CDNet could better understand user's intentions and make better predictions with limited interactions. CDNet achieves state-of-the-art performance on several benchmarks.

1. Introduction

Interactive segmentation has been a topic of research for a long while; various forms of interactions have been explored. Human could provide bounding boxes [26, 30, 15], scribbles [16, 6, 1], or clicks [27, 20, 32, 11] to express the segmentation intentions, which guide the algorithm for the mask extraction process. The segmentation target could be anything that users want, which requires interactive segmentation to be a flexible tool and makes it a challenging vision task. In this work, we address click-based interactive segmentation, and we aim to improve upon existing works by better understanding user's intentions.

For click-based interactive segmentation, users places positive/negative clicks (red/ green points in Fig. 1) to indicate foreground/ background regions. In general, a user's click contains two layers of information: the first layer is spatial – the location of the foreground/background could

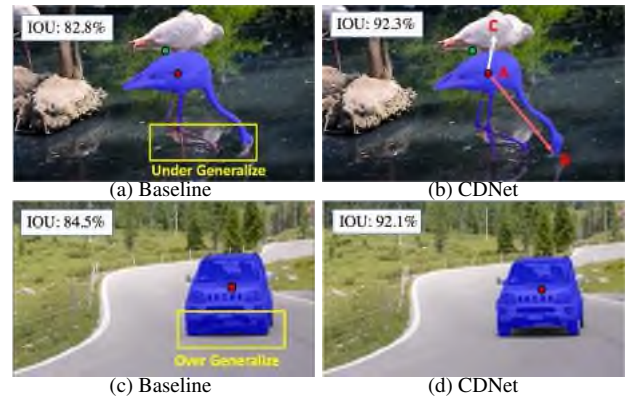


Figure 1. Comparison of baseline method and CDNet. Positive /negative clicks are marked in red and green. Diffusion flows are visualized in colored arrows. As the red arrow denotes, CDNet propagates information from clicks to target destinations, while suppressing the invalid flow denoted by the white arrow.

be indicated by the distribution of clicks; the second layer is visual – the label of regions around the clicks could be inferred by visual similarity. In a standard pipeline, the first step is encoding clicks into distance maps [32, 27, 11, 31], Gaussian maps [20, 23, 21, 14], or super-pixels [22]; next, encoded maps is concatenated with the original image and fed into a segmentation network to make predictions. This kind of method exploits good use of spatial information but ignores the visual hint. Consequently, we see two problems: 1) like the example in Fig. 1 (a), the label of a click often fails to be generalized to target regions further away from clicks, even if parts of the target region has almost the same appearance as the click. 2) the labels could spill over to the wrong regions, even if the target has clear boundaries, as demonstrated in Fig. 1 (c).

To address the aforementioned issues, we explored to model the affinity between different locations and diffuse the representations from clicks to unlabeled regions. A straightforward solution is modeling the affinity simply based on feature similarity. However, it would lead to over-generalization: in Fig. 1 (b), the information should not propagate from A to C (under the assumption that the target is the bottom flamingo), even though they are visually simi-

*Corresponding Author

lar. To deal with over-generalization, constraining the diffusion within a instance/ semantic- level range around clicks is an option that can be thought of directly. Again, since the target was flexibly defined and could be either part, instance, or stuff, a fixed constrain would not be a cure-all. In general, it is a dilemma that we expect to enlarge the diffusion range and avoid over-generalization simultaneously.

Facing this challenge, we conduct an in-depth study of information diffusion and propose Conditional Diffusion Network (CDNet), which diffuses information from clicks and dynamically constrains the diffusion destination. Specifically, two components are designed:

- **Feature Diffusion Module (FDM)** generates a conditional affinity map. It firstly considers feature similarity to diffuse the representations from clicks to all unlabeled regions. We then constrain the diffusion destination by a primitive prediction of foreground/ background per the distribution of clicks and the content of the image. FDM propagates features in full-image with a global perspective.
- **Pixel Diffusion Module (PDM)** constructs a conditional diffusion source on the predicted logits, and leverages color similarity to diffuse the labels of clicks. It constrains the diffusion in local regions with uniform color to avoid over-generalization. Meanwhile, the diffusion is carried out iteratively to enlarge the range of destination. PDM mainly refines the details.

FDM and PDM work in synergy to extract representations with both high-level and low-level consistency in a coarse-to-fine manner. To better train CDNet, we develop a practical training regime called Diversified Training (DT), which eases the optimization ambiguity caused by click simulation. Combing FDM/PDM/DT together, CDNet better exploits the information contained in clicks by diffusing the representations of clicks to correct destinations and makes better predictions with fewer clicks.

Large amounts of experiments have been conducted on GrabCut [26], Berkeley [24], SBD [8], and DAVIS [25] datasets. Results prove the effectiveness of our method and show that our CDNet achieves state-of-the-art performance. Our contributions could be summarized as follows: 1) We formulate click-based interactive segmentation as a process of conditional diffusion and propose CDNet, which predicts better segmentation results with fewer clicks. 2) We design FDM and PDM, which propagate the labeled information of clicks to enhance unlabeled regions on different levels. 3) To better train CDNet, we develop Diversified Training, which reduces the optimization ambiguity caused by click simulation.

2. Related Works

Classical Methods. Before the era of deep learning, researchers take interactive segmentation as an optimization problem. GrabCut [26] uses the Gaussian mixture model to solve the problem of max-flow in color space. Geodesics [7] calculates geodesic between clicked points and other pixels to predict segments with minimum energy cost. [6] applied random walk algorithm to predict the labels of unseeded pixels. [12] proposes a high order method with the constrain of label consistency. These classical methods model the relationship between pixels according to low-level similarity, which enables them to predict segmentation results with local consistency. However, lacking high-level semantic information limits the performance of classical methods.

Deep Learning Methods. The first deep learning-based method [32] embeds clicked points into distance maps and uses a fully convolutional network to predict the mask of foreground and background. RIS-Net [18] adds a local branch to refine the predicted result around human clicks. [22] uses super-pixels to embed clicked points to provide guidance with local consistency. [17] predicts multiple potential results and train another network to choose from them. FCANet [20] underlines the importance of the first click and proposes first click attention to get better results. BRS [11] uses backward propagation to finetune the guidance map in an online manner. f-BRS [27] refines the intermediate feature to get more precise masks with faster speed compared with BRS. Most of these learning-based methods only use clicks to generate the guidance map to indicate the rough location of the target object. BRS and f-BRS use the given labels of clicked points to fine-tune the network, but online learning brings extra computation during inference and makes them hard to deploy. Compared with classical methods, deep learning models get better performance. However, they are not utilizing the user inputs to the full potential.

3. Method

3.1. Pipeline Overview

The pipeline of Conditional Diffusion Network (CDNet) is shown in Fig.2. The blocks in blue demonstrate a commonly used baseline, on which we add two diffusion modules in red. First, we embed positive and negative clicks as two Gaussian maps and concatenate them with the original RGB image to get the 5-channel input. Second, the input is fed into a segmentation network to extract high-level features. In this work we use DeeplabV3+ [3] with ResNet-50 [9] backbone. Then, the stride-8 high-level feature and the Gaussian maps are sent into Feature Diffusion Module (FDM). In this block, the labeled features around clicks could be propagated to prospective unlabeled regions. Next, following DeeplabV3+, we make feature fusion with

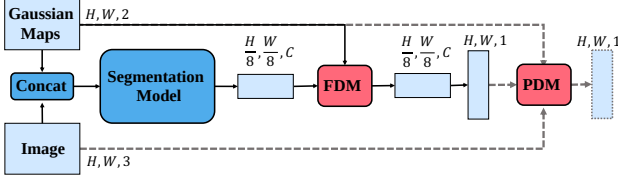


Figure 2. An overview of our Conditional Diffusion Network. FDM denotes Feature Diffusion Module. PDM denotes Pixel Diffusion Module. The dotted lines mean PDM only exists during inference.

low-level features and upsample the predicted logits to the size of the original image. Afterward, Pixel Diffusion Module (PDM) takes the original image, Gaussian Maps, and predicted logits to propagate the logits from clicks to their neighbors iteratively, which refines the prediction with low-level consistency. PDM only exists during inference. Training supervision is applied on the logits before PDM.

Both FDM and PDM are easily extensible and could be simple plugins for different segmentation models. With FDM and PDM, the effect of clicks could be amplified as their labeled representations are propagated to wider ranges, which enables the model to make better predictions.

3.2. Feature Diffusion Module

Feature Diffusion Module (FDM) propagates the labeled high-level features from clicks to conditional destinations. It enables unlabeled regions to be better represented via matching labeled templates and helps the labels of clicks to be generalized to both wide and accurate destinations.

The implementation of FDM is inspired by self-attention series [5, 10, 29, 33]. We first revisit the formulation of self-attention. Then we elaborate on the architecture of FDM.

3.2.1 Revisiting Self-Attention

Non-local Network [29] proposes a standard formula for self-attention, its variants [5, 10, 29, 33] are widely applied in the task of semantic segmentation and proven to be effective. The formulation could be summarized as Eq. (1) (2), where $\mathbf{A}$ denotes the affinity matrix which measures the dependency between features of each two positions. $\mathbf{x}$ stands for the input feature. $\mathbf{g}, \theta, \phi$ are transformation functions which are implemented with 1×1 Convs.

$$\mathbf{y}^{HW \times C} = \text{Softmax}(\mathbf{A}^{HW \times HW}) \times \mathbf{g}(\mathbf{x})^{HW \times C} \quad (1)$$

$$\mathbf{A} = \theta(\mathbf{x})^{HW \times C} \times \phi(\mathbf{x})^{C \times HW} \quad (2)$$

With self-attention, the information in $\mathbf{x}$ could be propagated cross long-distance between every two positions, which helps to build more unified feature representations with the global context.

3.2.2 FDM Overview

Regarding Eq. (1) as a process of information diffusion, the affinity matrix in Eq. (2) assigns equal chances for each location to act as the diffusion source; the diffusion destination is decided by only considering semantic similarity. It works for semantic segmentation, but it is not an optimal solution for interactive segmentation for two reasons: 1) as the labels of positive/ negative clicks are given, features around clicks are more informative and should be prioritized for diffusion; 2) since the foreground/ background are dynamically defined by clicks, we could not constrain the diffusion destination statistically using instance or semantic similarity.

To address the aforementioned problem, FDM introduces two additional features via dynamically re-weighting the affinity matrix: it highlights the diffusion flows from clicks, and constrains the diffusion destinations in the meantime. Formulated as Eq. (3), FDM uses two conditional affinity matrix $\mathbf{CA}_{F/B}$ to model the diffusion flow for foreground/ background information, and add the diffusion results together. The pipeline of FDM is demonstrated in Fig. 3, we first calculate the raw affinity matrix $\mathbf{A}$ following Eq. (2). Then, we generate conditional affinity matrices by re-weighting. The details of the conditional affinity would be introduced in the next paragraph.

$$\mathbf{y} = \mathbf{CA}_F \times \mathbf{g}(\mathbf{x}) + \mathbf{CA}_B \times \mathbf{g}(\mathbf{x}) \quad (3)$$

3.2.3 Conditional Affinity

We re-weight the raw affinity map by setting source constraint and destination constraint. Source constraint highlights the diffusion flows starting from clicks; Destination constraint defines a rough range for the diffusion, which prevents the label of clicks to be over-generalized. With these constraints, in Fig. 1 (b), features from A could be diffused to B, but would not be propagated to C.

Concretely, we generate source constrain maps $\mathbf{S}_{F/B}$ by placing Gaussian kernels on foreground/ background clicks, with the amplitude and the standard derivation both set as 1. Besides, we calculate destination constrain maps $\mathbf{D}_{F/B}$ through adding an auxiliary head on the input feature of FDM, which makes a primitive prediction for the foreground/ background regions. $\mathbf{D}_{F/B}$ is not expected to be accurate, it is used to control the probability for each position to collect information from the foreground/ background source. From another angle, FDM could be considered as a further refinement based on $\mathbf{D}_{F/B}$.

Since $\mathbf{S}_{F/B}, \mathbf{D}_{F/B}$ are normalized to $[0, 1]$, we directly re-weight the raw affinity matrix $\mathbf{A}$ according to Eq. (4). As denoted in the right block of Fig. 2, we first reshape $\mathbf{A}$ to shape $\mathbb{R}^{HW \times H \times W}$ and do element-wise multiplication with $\mathbf{S} \in \mathbb{R}^{1 \times W \times H}$ to highlight the flows starting from

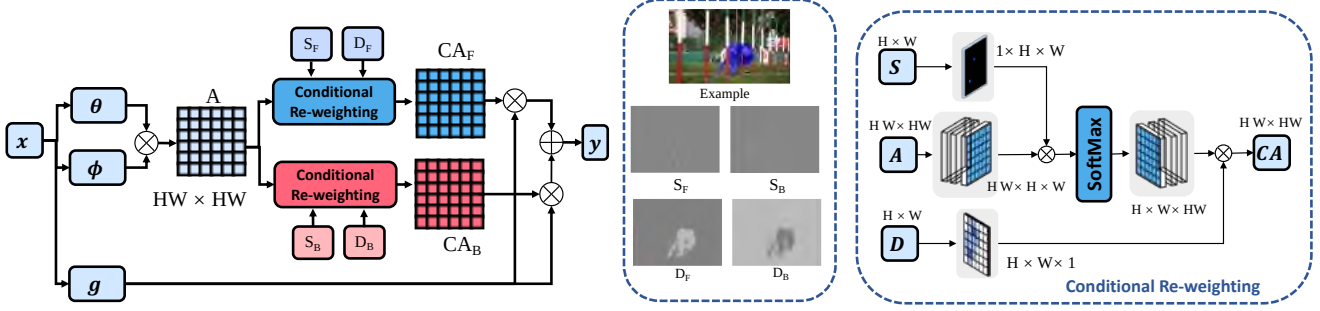


Figure 3. Demonstration of Feature Diffusion Module. We first measure the dependency between features and generate a raw affinity matrix A ; Then, we apply source constrain and destination constrain in Conditional Re-weighting Block to get a Conditional Affinity Matrix CA . Guided by the Conditional Affinity Map, features are propagated from source (clicked points) to target destinations to get the enhanced representation. The concrete structure of Conditional Re-weighting Block is demonstrated in the right part.

clicks. Then, softmax function normalizes the summation of source features that could be aggregated for each target. Afterward, the modified affinity map is reshaped to $\mathbb{R}^{H \times W \times HW}$ and multiplies the $D \in \mathbb{R}^{H \times W \times 1}$ to suppress the flows across foreground/background splits. Thus, we get two conditional affinity matrices $CA_{F/B}$ which control the information flow from foreground/ background clicks to corresponding target regions.

$$CA_{F/B} = \text{Softmax}(S_{F/B} \odot A) \odot D_{F/B} \quad (4)$$

3.2.4 Function Analysis

FDM propagates features from positive/ negative clicks to corresponding destinations. It could also be analyzed from the perspective of information gathering. For features predicted as foreground/ background with high confidence, they only gather information from foreground/ background clicks, which assists to make more unified representations for the target. For features with uncertain predictions, they have equal chances to gather information from the foreground and background sources. Thus, it could make a more reliable prediction by matching the foreground and background templates according to feature similarity.

From the perspective of optimization, FDM enforces each unlabeled feature to get closer to features of clicks with the same label, while enlarging the distance with clicks with opposite labels, which contributes to more unified and more discriminative representations.

3.3. Pixel Diffusion Module

Pixel Diffusion Module (PDM) is designed to complement FDM and focus on the details that could not be refined on high-level features. It follows the formula of FDM and propagates information from clicks to unlabeled regions according to affinity. Requiring the representations with rich details, PDM is performed on the full-resolution predicted logits.

3.3.1 PDM Overview

Limited by computing resources, the formulation of FDM could not be directly applied to the full-resolution logits. Therefore, we constrain the pixel diffusion in local regions and implement the diffusion iteratively. Following the basic form of FDM, we formulate PDM as Eq. (5), where A_{ij} represents the affinity between logit i and its neighbor j ; y^0 denotes the pixel diffusion source. Information is propagated from each logit i to its neighboring regions N_i iteratively.

$$y_i^{t+1} = \sum_{j \in N_i} \text{Softmax}(A_{ij}) \cdot y_j^t \quad (5)$$

Given the pipeline of PDM, there are still two key areas for consideration: 1) How to highlight the diffusion flows starting from clicks. 2) How to enlarge the diffusion destination while avoiding over-generalization.

First, we highlight the flows from clicks by constructing a conditional diffusion source, in which the information concentration around clicks is augmented. Thus, the labels of clicks get higher priorities to be propagated out; Second, we calculate the affinity using color similarity, and constrain each iteration of diffusion within a small range. In this way, the diffusion would be truncated when it meets boundaries or a sharp color change. At the same time, we conduct the pixel diffusion iteratively, thus the enhanced logits could diffuse further step by step in regions with uniform colors.

The Pipeline of PDM is demonstrated in Fig.4. Given the input image, the predicted logits, and the clicks, we construct the conditional diffusion source and propagate the information of the source in the local neighborhood iteratively. Finally, we add the original logits as a residual on the diffusion result and set the threshold as 0 to get the binary prediction mask. Noticing that the whole pipeline of PDM could be implemented with Conv layers. PDM runs on GPUs with high efficiency.

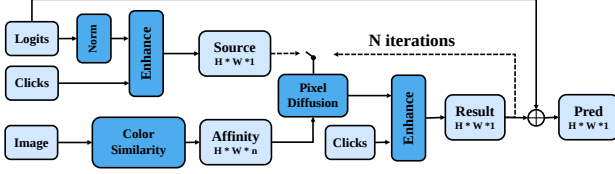


Figure 4. Pipeline of Pixel Diffusion Module (PDM). PDM takes the original image, Gaussian maps of clicks, and the predicted logits as input. It propagates information from clicks to neighbors iteratively to get the refined prediction map.

3.3.2 Conditional Diffusion Source

We construct the conditional diffusion source by augmenting the information concentration around clicks. Considering the network is trained with Sigmoid function and Cross-entropy loss, the absolute value of logits could be large. To manipulate the logits, we first normalize the values of logits into a controllable range. Concretely, we normalize the logits into $[-1, 1]$ according to Eq. (6). Then, we enhance the normalized logits y_{norm} via placing Gaussian kernels at clicked positions as in Eq. (7), with amplitude and standard derivative set to 1 and 10.

$$y_{norm} = \text{Tanh}(y_{raw}) \times 2 \quad (6)$$

$$y^0 = \text{Enhance}(y_{norm}) = y_{norm} + G_F - G_B \quad (7)$$

3.3.3 Dynamic Diffusion Range

We leverage color similarity to measure the affinity, which is a robust low-level feature used in some traditional methods [26,13]. The affinity between pixel i and pixel j as in Eq.(8). σ_i denotes the standard derivation.

$$A_{ij} = -|I_i - I_j| / \sigma_i^2 \quad (8)$$

Each diffusion iteration is constrained within the local range of n neighbors; the affinity is also calculated locally. In this work, we use four 3×3 convolution filters with dilation $\{1,2,4,8\}$ to sample the neighbors, so $n = 4 \times 8 = 32$. Then, we enlarge the diffusion range by applying the diffusion iteratively. Thus, information flows could go further in regions with uniform color, and would be truncated when encountering edges.

3.3.4 Function Analysis

For regions around clicks, PDM propagates logits from clicks to visually similar neighbors, which guarantees the correct prediction around clicks could be generalized in local regions. For regions far from clicks, PDM also makes refinement with local-consistency, it enforces adjacent pixels with similar colors to predict similar labels.

3.4. Diversified Training

Ambiguity is a common problem for interactive segmentation; divergence between segmentation result and user's true intention frequently happens. For example, in Fig.5, when only one click on the leg of the rider was placed as foreground, there could be many possible and reasonable targets: the leg, the rider, or the entirety of rider/motorcycle.

Some works [17,19] focus on the inference procedure to tackle the ambiguity. They propose to predict multiple masks and require the user or a selection network to pick one of them. However, our analysis concludes that if the model is well-trained, the ambiguity can be naturally reduced during inference when more clicks are sequentially placed; the real challenge lies in training, where the ambiguity is difficult to reduce even with bigger numbers of clicks. Many previous works [27,11,32,22] simply simulate interactions during training by randomly sampling positive/ negative clicks inside/ outside the given ground truth mask. There is no guarantee that clicks could clarify the outline of the given ground truth. Consequently, it is hard to train the model well, as the optimization target varies.

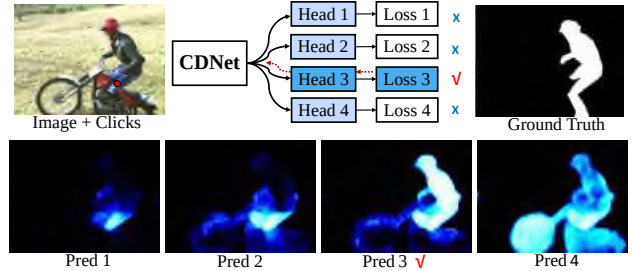


Figure 5. Demonstration of Diversity Training. We predict N more potential results and chose the one most similar to the ground truth to propagation the gradients. The original head is not shown for simplicity.

In this work, we reduce the ambiguity by developing Diversified Training (DT). Inspired by [17,19], we explore the latent diversity; differently, we focus on the training process instead of inference. Fig.5 depicts the pipeline of DT. We add another N latent heads during training and remove them during inference. In this work, we set $N = 4$. We supervise these latent heads with diversity loss and click loss. As Eq. (9), diversity loss calculates the cross-entropy of each latent head and chooses the minimum one to initiate backward propagation. With this setup, the ambiguity is eased by permitting all reasonable predictions. We also design a click loss to enforce each latent head to make correct predictions around labeled clicks. As Eq. (10), M_i denotes the Gaussian mask of clicks, which only keeps the gradients in clicked regions. The total loss is the sum of the original binary cross-entropy loss, the diversity loss, and the click loss

as Eq. (11). In this work, we set a_0, a_1, a_2 as 0.5, 1, 1.

$$\mathcal{L}_{div} = \min(\mathcal{L}_{bce}(\mathbf{P}_i, \mathbf{gt}) \mid i \in [1, n]) \quad (9)$$

$$\mathcal{L}_{click} = \sum_{i \in [1, n]} \mathcal{M}_i \mathcal{L}_{bce}(\mathbf{P}_i, \mathbf{gt}) \quad (10)$$

$$\mathcal{L}_{total} = a_0 \mathcal{L}_{bce}(\mathbf{P}_0, \mathbf{gt}) + a_1 \mathcal{L}_{div} + a_2 \mathcal{L}_{click} \quad (11)$$

We only keep the original head during inference. Although removed for prediction, latent heads make contributions during training; they assist the model to learn better representations. Based on the representations, the original head learns the projection relation between the distribution of input clicks and the prospective mask, which enables the original head to make better predictions during inference when given enough clicks.

4. Experiments

4.1. Experiment Configurations

Implementation Details. For the 5-channel input, we embed user clicks into two Gaussian Maps with the amplitude as 1, the standard derivation as 10. Following f-BRS [27], we use the same Map Fusion block to adjust the 5-channel input to 3-channel tensor using 1×1 convolutions and LeakyReLU. So that the 3-channel tensor could be fed into a ResNet-50 [9] backbone pretrained on ImageNet [4]. During inference, we applied the same cropping strategy as f-BRS. Starting from the third click, we calculate a minimum box around the predicted mask and expand the box with 40% along sides. Then, we crop the image according to the box and apply interactive segmentation only on the Zoom-In region.

Training Hyper-Parameters. We train our CDNet on SBD [8] train set with 8498 images. We crop training images with 320×480 . For data augmentation, we applied random rotation, flip, random resize (0.75 1.25), random brightness (-0.25 0.25), random contrast (-0.15 0.4), and RGB shift (shift limit = 10). We use Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ for 120 epochs. The first two epochs are the warm-up stage in which the learning rate increases linearly from 0 to 5×10^{-4} . For the remaining epochs, cosine annealing learning rate is applied. We train our model on 4 GPUs with batch size 32 using asynchronous BatchNorm.

Training Click Simulation. Positive/ negative clicks are simulated by sampling points inside/outside the ground truth mask following [32]. The number of foreground and background points is randomly chosen in [1, 10] and [0, 10]

with a probability decay rate of 0.7.

Evaluation Protocol. For fair comparisons, we follow the protocol of previous works [32, 18, 18, 22, 27, 11], and generate clicks automatically: The first click is placed on the center of the ground truth mask. Following clicks are placed at the center of the largest error region iteratively until reaching the targeted Intersection over Union (IoU) or the max click number.

Evaluation Metrics. We report the average Number of Click (NoC) required to reach the target IoU and set the target IoU as 85% and 90%. We set the default max number of clicks as 20 and report the Number of Failure (NoF) that could not reach the target IoU with 20 clicks. Since response time is important for industrial applications, we also report Second Per Click (SPC) to measure the speed of our method on a single 1080 Ti GPU.

4.2. Comparison to state-of-the-art

We compare our Conditional Diffusion Network with other state-of-the-art click-based methods on four benchmarks. Comparison results could be found in Tab. 1.

- **GrabCut** [26] : GrabCut dataset contains 50 images. It is commonly used to evaluate the performance of interactive segmentation models.
- **Berkeley** [24] : Berkeley dataset contains 96 images with 100 instance masks for testing.
- **SBD** [8] : SBD is a relatively larger dataset with 2,802 test images with 6,671 instance masks.
- **DAVIS** [25] : DAVIS dataset is annotated for the task of video object segmentation, which contains 10 videos. We sample the same 345 frames as BRS [11].

Result Analysis. As shown in Tab. 1, CDNet outperforms other models on all four datasets with large margins. We do not include FCANet [20] because it uses more training data than other works.

Speed Analysis. Interactive segmentation is often used in annotation tools that need immediate feedback. Hence, inference speed is an important factor. FDM models the affinity across the full image, but it is applied on features with low resolution. PDM is applied on full-resolution logits, but it diffuses the information in the local neighborhood. Therefore, the budgets brought by FDM and PDM are affordable. In Tab. 2, we compare the running speed of our method on DAVIS dataset, with f-BRS [27] and BRS [11], the previous SOTA methods. f-BRS [27] and BRS [11] apply online learning to fine-tune the parameters of the network, which enables them to make accurate predictions with few clicks.

Method	GrabCut		Berkeley	SBD		DAVIS	
	NoC@85	NoC@90	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
Graph cut [2]	7.98	10.00	14.22	13.6	15.96	15.13	17.41
Geodesic matting [7]	13.32	14.57	15.96	15.36	17.60	18.59	19.50
Random walker [6]	11.36	13.77	14.02	12.22	15.04	16.71	18.31
Euclidean star convexity [7]	7.24	9.20	12.11	12.21	14.86	15.41	17.70
Geodesic star convexity [7]	7.10	9.12	12.57	12.69	15.31	15.35	17.52
Growcut [28]	–	16.74	18.25	–	–	–	–
DOS w/o GC [32]	8.02	12.59	–	14.30	16.79	12.52	17.11
DOS with GC [32]	5.08	6.08	–	9.22	12.80	9.03	12.58
Latent diversity [17]	3.20	4.79	–	7.41	10.78	5.05	9.57
RIS-Net [18]	–	5.00	–	6.03	–	–	–
CM guidance [22]	–	3.58	5.60	–	–	–	–
BRS [11]	2.60	3.60	5.08	6.59	9.78	5.58	8.24
f-BRS-B-50 [27]	2.50	2.98	4.34	5.06	8.08	5.39	7.81
Ours	2.22	2.64	3.69	4.37	7.87	5.17	6.66

Table 1. Evaluation results on GrabCut, Berkeley, SBD and DAVIS datasets. NoC@85/ 90 denotes the average Number of Clicks required the get IoU of 85/ 90%.

However, online learning is time-consuming and hard to deploy. Results show that CDNet not only surpasses BRS and f-BRS on accuracy, but also is evidently faster.

Method	baseline	Ours	BRS[27]	f-BRS [27]
SPC (s)	0.20	0.23	1.47	0.32
NoC@90	8.42	6.66	7.93	7.81

Table 2. Comparison for inference speed on DAVIS dataset. Speeds are measured with the same hardware settings.

Method	NoF ₂₀ @90	NoF ₁₀₀ @90	NoC ₁₀₀ @90
Baseline	84	64	24.03
BRS [11]	77	51	20.89
f-BRS [27]	78	50	20.70
Ours	65	48	18.59

Table 3. Experiment for NoC₁₀₀ on DAVIS dataset. NoF₁₀₀@90 denotes the Number of Failure images that could not reach IoU 0.9 under 100 clicks. NoC₁₀₀@90 means the average Number of Clicks required to reach IoU 0.9 under 100 clicks.

Analysis for 100 Clicks. Following f-BRS [27], in Tab.3, we also report the metric under 100 clicks on DAVIS dataset. The motivation is that the traditional NoC₂₀ evaluates images requiring 20 clicks and images requiring 200 clicks with the same NoC results. However, many images need more than 20 clicks, which makes NoC₂₀ not distinguishing for difficult images. Results show that our method gets better performance than f-BRS with a clear margin under the metric of NoC₁₀₀.

4.3. Ablation Studies

We carry out plenty of ablation studies to verify the effectiveness of our method. We choose DAVIS [25] dataset

to evaluate the performance. Since the masks in DAVIS are annotated with high quality, and images in DAVIS cover various scenarios, the result on DAVIS is more convincing. We first prove the effectiveness of our three core components: FDM, PDM, and DT. Then we dive into details to give an in-depth analysis for FDM and PDM.

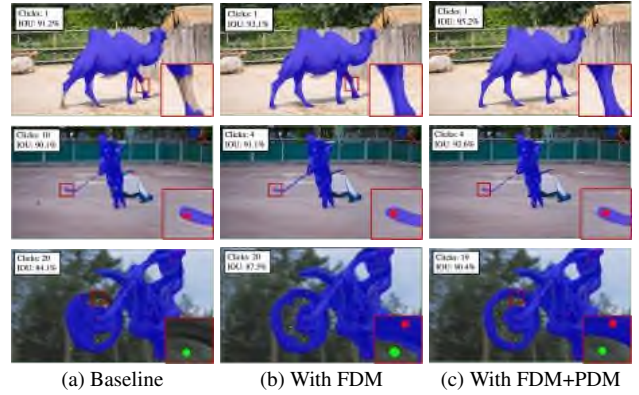


Figure 6. Qualitative results on DAVIS dataset. Three columns demonstrate the result of the baseline method, our method with FDM, and our method with FDM+PDM.

Effectiveness of Core Components. In Tab.4, we verify the effectiveness of FDM, PDM, and DT. The baseline listed in the first row is a DeeplabV3+ [3] with 5-channel input. Results demonstrate that three new modules all bring steady improvements. Baseline⁺ with DT could serve as a stronger baseline without extra computation during inference. FDM brings clear improvement with only 0.01 seconds of inference time. PDM requires affordable 0.02 seconds, while brings significant improvement for the NoC. Equipped with all three proposed components, CDNet posts remarkable performance gains with reasonable computation

Version	DT	FDM	PDM	NoC@80	NoC@85	NoC@90	NoF@85	NoF@90	SPC (s)
Baseline				4.27	5.60	8.42	52	84	0.198
Baseline ⁺	✓			4.24	5.47	8.14	51	84	0.198
CDNet		✓		4.10	5.40	7.64	51	72	0.208
CDNet			✓	4.07	5.39	7.06	49	63	0.221
CDNet	✓	✓	✓	3.89	5.17	6.66	46	61	0.230

Table 4. Ablation studies for core components of CDNet on DAVIS dataset. **FDM** denotes Feature Diffusion Module. **PDM** denotes Pixel Diffusion Module. **DT** denotes Diversified Training. For the metrics, NoC stands for the average Number of Clicks required to get target IoU. NoF denotes Numbers of Failure cases that could not reach the target IoU in 20 clicks. SPC means Second Per Click.

Version	S	D	NoC@85	NoC@90	SPC (s)
Baseline			5.60	8.42	0.198
Auxiliary loss			5.60	8.61	0.198
Non-local [29]			5.87	8.81	0.202
Source2Full	✓		5.56	8.49	0.203
Des2Des	✓	✓	5.62	8.10	0.208
Source2Des ⁻	✓	✓	5.53	8.04	0.208
FDM	✓	✓	5.40	7.64	0.208

Table 5. Ablation studies for Feature Diffusion Module on DAVIS dataset. S, D denotes Source Constrain, Destination Constrain.

overhead.

Qualitative comparisons for FDM and PDM are shown in Fig. 6. Three columns demonstrate the results of baseline, CDNet with FDM, and CDNet with FDM+PDM. FDM enables the features of clicks to be generalized to wider regions, which assists to correct large regions of false prediction. PDM refines the boundary with low-level consistency, which helps to capture fine details as shown in zoomed-in patches. The positions of clicks are not exactly the same for each column, because the clicks are generated according to the evaluation protocol introduced in 4.1.

Experiments for FDM. In Tab. 5, we prove the effectiveness of our proposed conditional affinity. First, as FDM introduces the auxiliary supervision for the destination constraint, we add an auxiliary loss on the baseline without FDM to make the comparison in the second row. Then, we develop four variants of FDM with different settings of source and destination constraints: **Non-local** could be regarded as a variant of FDM without any constraint; **Source2Full** denotes the version only with source constraint; **Des2Des** uses the destination constrain map in FDM to constrain both the diffusion source and destination; **Source2Des⁻** applies both source /destination constraints, but remove the supervision for destination constraints and make the model learn it in an end-to-end manner.

Results show that: 1) A single auxiliary loss could not bring improvement. 2) Non-local exert a negative effect. We analyze that it is hard to directly learn a conditional affinity map with fixed convolutional filters for interactive segmentation. 3) Single source /destination constrain do

bring improvements compared with the raw non-local layer, but it is still sub-optimal compared with FDM. Thus, the conditional affinity in FDM is proven to be effective.

Amplitude	0	0.2	0.5	1	2
NoC@90	6.89	6.87	6.86	6.66	6.72

Table 6. Experiment for the enhancement amplitude for PDM on DAVIS dataset with 10 times of iteration.

Iterations	0	1	5	10	20
SPC (s)	0.208	0.216	0.224	0.230	0.243
NoC@90	7.54	7.15	6.74	6.66	6.66

Table 7. Experiment for the diffusion iteration of PDM on DAVIS dataset.

Experiments for PDM. PDM constructs a conditional diffusion source and diffuses the information in local regions iteratively. The conditional diffusion source is constructed by enhancing the logits around clicks using Gaussian kernels as Eq. (7). In Tab. 6, we prove the effectiveness of the conditional source by changing the amplitude of Gaussian kernels. It can be observed that source enhancement brings steady improvements, and we simply set the amplitude to 1.

In Tab. 7, we explore the trade-off between speed and accuracy for different diffusion iterations. The performance reaches saturation at about 10 times refinement. Considering the trade-off between accuracy and speed, we set the iteration number as 10 in this work.

5. Conclusion

In this paper, we formulate click-based interactive segmentation as a process of information diffusion and propose Conditional Diffusion Network. We design a Feature Diffusion Module and a Pixel Diffusion Module to propagate information from clicked points to target regions. Experiments show that our method is effective on four benchmarks and sets new state-of-the-art.

References

- [1] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 392–399, 2014. [1](#)
- [2] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 105–112. IEEE, 2001. [7](#)
- [3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. [2](#), [7](#)
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [6](#)
- [5] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. [3](#)
- [6] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006. [1](#), [2](#), [7](#)
- [7] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3129–3136. IEEE, 2010. [2](#), [7](#)
- [8] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011. [2](#), [6](#)
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [2](#), [6](#)
- [10] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–612, 2019. [3](#)
- [11] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5306, 2019. [1](#), [2](#), [5](#), [6](#), [7](#)
- [12] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Non-parametric higher-order learning for interactive segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3201–3208. IEEE, 2010. [2](#)
- [13] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. [5](#)
- [14] Hoang Le, Long Mai, Brian Price, Scott Cohen, Hailin Jin, and Feng Liu. Interactive boundary prediction for object selection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 18–33, 2018. [1](#)
- [15] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *2009 IEEE 12th international conference on computer vision*, pages 277–284. IEEE, 2009. [1](#)
- [16] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transactions on Graphics (ToG)*, 23(3):303–308, 2004. [1](#)
- [17] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 577–585, 2018. [2](#), [5](#), [7](#)
- [18] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2746–2754. IEEE, 2017. [2](#), [6](#), [7](#)
- [19] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, Sim-Heng Ong, and Jiashi Feng. Multiseg: Semantically meaningful, scale-diverse segmentations from minimal user input. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 662–670, 2019. [5](#)
- [20] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13339–13348, 2020. [1](#), [2](#), [6](#)
- [21] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. *arXiv preprint arXiv:1805.04398*, 2018. [1](#)
- [22] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11602–11611, 2019. [1](#), [2](#), [5](#), [6](#), [7](#)
- [23] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625, 2018. [1](#)
- [24] Kevin McGuinness and Noel E O’connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010. [2](#), [6](#)
- [25] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. [2](#), [6](#), [7](#)
- [26] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ” grabcut” interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004. [1](#), [2](#), [5](#), [6](#)
- [27] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition*, pages 8623–8632, 2020. [1](#), [2](#), [5](#), [6](#), [7](#)
- [28] Vladimir Vezhnevets and Vadim Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. In *proc. of Graphicon*, volume 1, pages 150–156. Citeseer, 2005. [7](#)
 - [29] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. [3](#), [8](#)
 - [30] Jiajun Wu, Yibiao Zhao, Jun-Yan Zhu, Siwei Luo, and Zhuowen Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 256–263, 2014. [1](#)
 - [31] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243*, 2017. [1](#)
 - [32] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016. [1](#), [2](#), [5](#), [6](#), [7](#)
 - [33] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. [3](#)

Exploiting Diverse Characteristics and Adversarial Ambivalence for Domain Adaptive Segmentation

Bowen Cai^{1,2}, Huan Fu¹, Rongfei Jia¹, Binqiang Zhao¹, Hua Li² and Yinghui Xu¹

¹Alibaba Group

²Institute of Computing Technology, Chinese Academy of Sciences
 {kevin.cbw, fuhuan.fh, rongfei.jrf, binqiang.zhao}@alibaba-inc.com
 {lihua}@ict.ac.cn

Abstract

Adapting semantic segmentation models to new domains is an important but challenging problem. Recently enlightening progress has been made, but the performance of existing methods is unsatisfactory on real datasets where the new target domain comprises of heterogeneous sub-domains (*e.g.*, diverse weather characteristics). We point out that carefully reasoning about the multiple modalities in the target domain can improve the robustness of adaptation models. To this end, we propose a condition-guided adaptation framework that is empowered by a special attentive progressive adversarial training (APAT) mechanism and a novel self-training policy. The APAT strategy progressively performs condition-specific alignment and attentive global feature matching. The new self-training scheme exploits the adversarial ambivalences of easy and hard adaptation regions and the correlations among target sub-domains effectively. We evaluate our method (DCAA) on various adaptation scenarios where the target images vary in weather conditions. The comparisons against baselines and the state-of-the-art approaches demonstrate the superiority of DCAA over the competitors.

Introduction

Due to the expensive labeling costs, there is a growing interest in studying weakly-supervised or unsupervised approaches for semantic segmentation in recent years. One of the widely investigated techniques is synthetic-to-real unsupervised domain adaptation (UDA) (Hoffman et al. 2016; Tsai et al. 2018; Zhang, David, and Gong 2017; Zou et al. 2019), which expects to transfer knowledge from labeled synthetic datasets to unlabeled real domains. Recent solutions can be mainly divided into two streams, including feature alignment and self-training. The former utilizes adversarial training to align both the image styles and semantic feature distributions. As a promising alternative, self-training focuses on iteratively generating pseudo-labels with high confidence and fewer noises for target images, and retrain the networks using the produced labels. However, the performance of these models is still far from satisfactory on real datasets. One possible reason is that existing methods assume that images in the target domain are captured under similar conditions, while a new real domain can have

diverse characteristics. For example, the images taken in the same place with the same viewpoint may show significant differences because of weather changes.

The above analysis motivates us to study a more realistic UDA setting for semantic segmentation, *i.e.*, the images in a target domain has diverse characteristics. Taking “weather” as an example, a naive idea is to enlarge the synthetic dataset to cover as many weather conditions as possible and then directly adopt existing high-performing UDA methods. Unluckily, we find that the performance gains are not that remarkable. A possible reason is that the adversarial adaptation learning process may be more challenging when images in both domains show multiple modalities, resulting in semantic distortion translations and ambiguous feature alignment near weather condition boundaries. In fact, minimizing the discrepancy between synthetic and real domains is challenging (Saito et al. 2018), not to mention that each domain consists of several sub-domains (*e.g.*, Cityscapes*: {*Cloudy, Foggy, Rainy*}). Moreover, the learned pseudo-labels for target images contain significant noise when the atmosphere states are unconstrained.

In this paper, we argue that carefully reasoning about the diverse characteristics in the target domain may improve the effectiveness of adapted segmentation model. Towards this goal, we propose a condition guided adaptation framework (DCAA) empowered by a special attentive progressive adversarial training (APAT) mechanism and a novel self-training policy. We illustrate our observations and idea in Figure 1. For the APAT strategy, we first introduce a condition-guided style translator to generate realistic characteristic conditionable samples from synthetic images. Furthermore, we develop a condition attention module to exploit the complementarities of multi-modalities. Lastly, we study condition-specific adversarial training to align semantic feature distributions under each condition and further mitigate the global discrepancy.

For the novel self-training scheme, we investigate the correlations among target sub-domains to produce robust attentive pseudo-labels for target images. Importantly, we point out that the dense output from the discriminators implies adversarial ambivalence for hard adaptation regions. We thus consider the ambivalence to further improve the self-training process by revising the pseudo-labels and emerging the hard adaptation regions. We evaluate our DCAA on several chal-

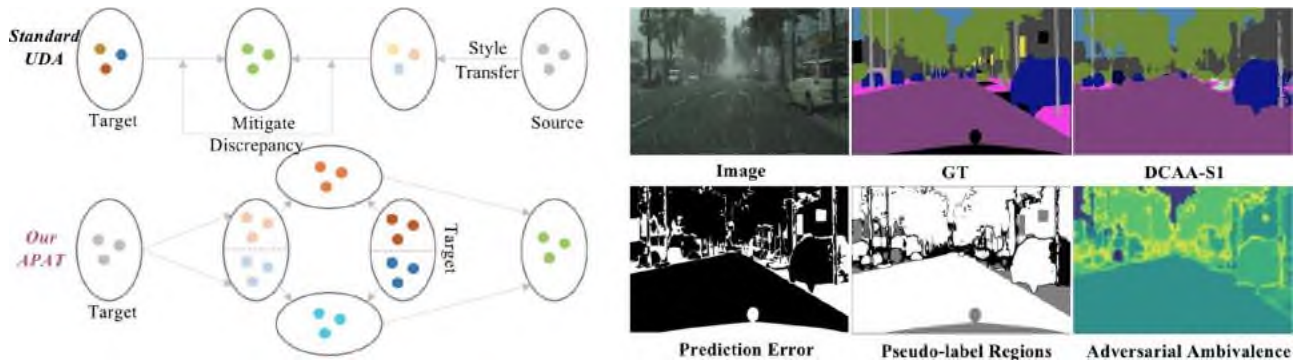


Figure 1: Left: A comparison between the typical domain adaptation process (Left Top) and our attentive progressive adversarial training (APAT) scheme (Left Bottom). Right: Adversarial Ambivalence and Pseudo-label. DCAA-S1: The first-stage prediction of DCAA. Prediction Error: The inconsistency between GT and DCAA-S1 (White). Pseudo-label Regions: “Black” represents the hard adaptation regions (without pseudo-labels); “Gray” and “White” denote regions with incorrect and correct pseudo-labels, respectively. Adversarial Ambivalence: The adversarial confidence obtained from the discriminator. “Yellow” means the regions are not well adapted. We find that the adversarial ambivalence shows consistency with the prediction error, thus can revise the pseudo-label related constraints. We refer to the “Introduction” section for more analyses of our motivations.

challenge adaptation scenarios, including GTA5→Cityscapes* (Richter et al. 2016; Cordts et al. 2016; Sakaridis et al. 2018; Hu et al. 2019), SYNTHIA→Cityscapes* (Ros et al. 2016), and GTA5→BDD100K (Yu et al. 2020a). Qualitative and quantitative comparisons against baselines and SOTA UDA methods demonstrate the superiority of our DCAA.

Related Work

Recent unsupervised domain adaptive segmentation solutions mainly focus on two points, *i.e.*, feature alignment (Du et al. 2019; Yang and Soatto 2020; Wang et al. 2020; Kim and Byun 2020; Xiao, Gu, and Zhang 2020; Li et al. 2020; Yang et al. 2019; Zhang and Wang 2020; Chen et al. 2019a; Wang, Cui, and Zhu 2018; Fu et al. 2019; Zhao et al. 2020; Gong et al. 2019a; Yu et al. 2020b) and self-training (Zhou et al. 2017; Zhang, David, and Gong 2017; Pan et al. 2020; Pei et al. 2018; Chen, Xue, and Cai 2019; Vu et al. 2019). Among the feature alignment approaches, FCN in the wild (Hoffman et al. 2016) was the first to introduce adversarial learning for global feature alignment and category statistic matching. AdaptSegNet (Tsai et al. 2018) modeled the structured spatial similarities on the semantic output space. Subsequently, many promising strategies has been devised to learn domain invariant features, such as conditional generator (Hong et al. 2018), task-specific decision boundaries (Saito et al. 2018), co-training (Luo et al. 2019), and patch distribution matching (Tsai et al. 2019). Other works show that style transfer (Zhu et al. 2017a) could largely mitigate the domain gap (Zhang et al. 2018; Sun et al. 2019). Thus, researchers exploited techniques such as semantic consistency (Hoffman et al. 2018), channel-wise feature alignment (Wu et al. 2018), symmetric adaptation consistency (Chen et al. 2019b), and controllable translation degrees (Gong et al. 2019b) to improve both the style translation and adaption processes. Self-training has recently gained considerable interests for UDA. For example, CBST (Zou et al. 2018) introduced an iterative class-balanced self-

placed learning policy to alternatively performed balanced pseudo-label generation and model retraining. The following works demonstrated the superiority of self-training by studying confidence regularization (Zou et al. 2019), bidirectional adversarial translation and self-supervised learning (Li, Yuan, and Vasconcelos 2019), and category-wise centroids (Zhang et al. 2019). These works always overlooked the diverse characteristics of natural images in a target domain, thus might produce sensitive models.

There are two adaptation settings that are closely related to our studied setting, *i.e.*, multi-target domain adaptation (MTDA) (Gholami et al. 2020) and compound domain adaptation (CDA) (Liu et al. 2020; Chen et al. 2019c). In MTDA, researchers expected to learn a general domain-agnostic model for largely unconstrained target domains. For example, the widely studied PACS dataset (Li et al. 2017) contains domains such as photo, cartoon, sketch, and art painting. In CDA, algorithms were developed to discover the unknown sub-domains in a target domain. In contrast, we study the setting where the sub-domains (from a single target domain) are strongly correlated in content, and the sub-domain labels are available (*e.g.*, BDD100K). We mainly take efforts into modeling the correlations (complementarities) of the multi-modalities. Thus, our proposed approach is largely different from these MTDA and CDA methods. We clarify that assigning image-level (sub-domain) labels to a target domain is inexpensive in practice. We could even learn a condition classification network for the purpose.

Methodology

In the section, we take a special case that the target images are under two weather conditions to illustrate our method. Let $X^S = \{x_i^s\}_{i=1}^N$ and $Y^S = \{y_i^s\}_{i=1}^N$ be the source images and corresponding dense labels. The target images consist of two parts, *i.e.*, $X^T = \{X^{T1}, X^{T2}\}$, where T_1 and T_2 represent the special environmental conditions in this case. Our goal is to transfer knowledge from S to T so that a learned

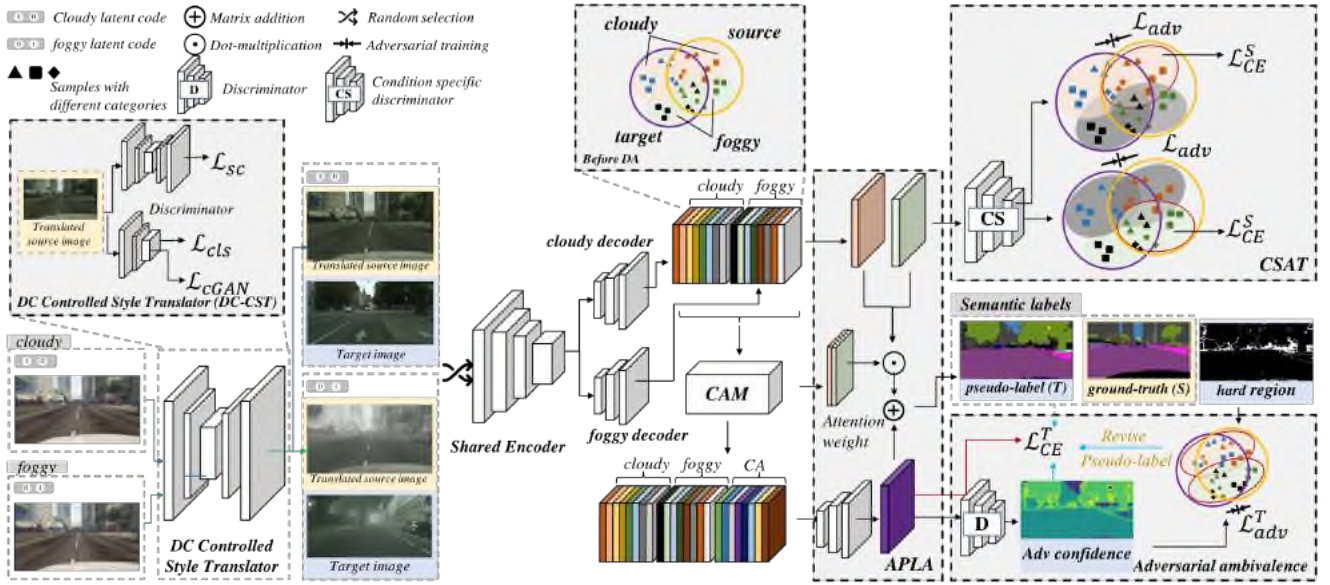


Figure 2: DCAA. We take GTA5→Cityscapes: $\{Cloudy, Foggy\}$ as an example to illustrate our DCAA. The framework is empowered by a special attentive progressive adversarial training scheme (CGST + CAM + CSAT) and a novel self-training policy (APLA + Adversarial Ambivalence).

segmentation model can produce promising parsing for images in the target domain. To this end, we develop a condition guided adaptation framework (DCAA), as shown in Figure 2. In the following, we will present our attentive progressive adversarial training (APAT) mechanism and self-training policy in detail.

Attentive Progressive Adversarial Training

Previous works (Luo et al. 2019; Hoffman et al. 2018; Tsai et al. 2018, 2019) preferred to take adversarial training to straightforwardly mitigate the source and target discrepancy. In contrast, we find that progressively reducing the domain shift by considering the weather conditions in the target domain would improve the adaptation process. Here, our APAT solution are constructed with three policies including condition guided style transfer, condition attention, and condition-specific adversarial training.

Condition Guided Style Transfer (CGST). We are here to learn a conditional generator $G_{ST}(X^S, c) : X^S \rightarrow X^T = \{X^{T_1}, X^{T_2}\}$ such that $G_{ST}(X^S)$ has the same distribution as X^{T_1} or X^{T_2} according to the specific latent variable $c \in \{0, 1\}$ (represents $\{T_1, T_2\}$). Thus, we develop a condition guided style transfer (CGST) following standard conditional GANs (Mirza and Osindero 2014) coupled with a semantic consistency constraint. We borrow concepts in StarGAN (Choi et al. 2018) and BicycleGAN (Zhu et al. 2017b) to improve the translation.

Specifically, we adopt a multi-level concatenation strategy (Zhu et al. 2017b) to inject c into G_{ST} as the translation guidance. We perform adversarial training to enforce G_{ST} producing realistic images with preferred styles. The adver-

sarial constraint takes the form as:

$$\mathcal{L}_{cGAN} = \mathbb{E}_{x^t \sim X^T} [\log(D_T(x^t))] + \mathbb{E}_{x^s \sim X^S} [\log(1 - D_T(G_{ST}(x^s, c)))] \quad (1)$$

where D_T is the discriminator with respect to X^T . The guidance from the injecting operation alone may not guarantee weather conditionable translations. Thus, we add an auxiliary condition classification loss (Choi et al. 2018) to further constraint the solution space:

$$\mathcal{L}_{cls} = \mathbb{E}_{x \sim X^T, c} [-\log(D_{cls}(c|x))] + \mathbb{E}_{x \sim X^T, c} [-\log(D_{cls}(c|G_{ST}(x)))], \quad (2)$$

where D_{cls} is the condition classifier which shares the same backbone with D^T .

We find that the obtained translator driven by constraints above would generate images with semantic distortion or inconsistency. For example, the trees in source images may become unreasonable building materials after the translation. We provide a remedy by incorporating a semantic consistency constraint which can be expressed as:

$$\mathcal{L}_{sc} = \mathbb{E}_{(x,y) \sim (X^S, Y^S)} [-\log(F_{seg}(y)|G_{ST}(x, c))], \quad (3)$$

where F_{seg} is a segmentation network. We use a light version of UperNet (Xiao et al. 2018) in the paper.

The full objective for our CGST can be expressed as:

$$\mathcal{L}_{CGST} = \mathcal{L}_{cGAN} + \mathcal{L}_{cls} + \lambda_{sc} \mathcal{L}_{sc}, \quad (4)$$

where the trade-off parameter λ_{sc} is set to 5.0 in our paper.

Condition Attention Module (CAM). After obtaining the stylized images $\hat{X}^S = \{\hat{X}^{S_1}, \hat{X}^{S_2}\} = G_{ST}(X^S, c)$, the trivial solution is to learn some multi-head networks to

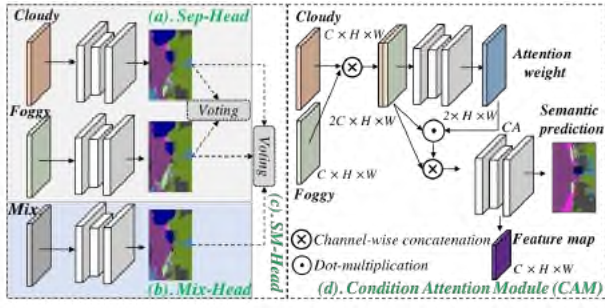


Figure 3: Several options in incorporating weather condition priors, including Sep-Head, Mix-Head, SM-Head, and our CAM. Dot lines refer that we use mean voting to capture the final prediction in the inference stage.

explicitly incorporate weather priors, as shown in Figure 3. We clarify that the effortless procedures may not sufficiently exploit the correlations of multi-modalities. We thus develop the condition attention module to address the issue.

In detail, we feed $x \in \{\hat{X}^S, X^T\}$ into a shared encoder (*Enc*) and two individual decoders (Dec^{C_1} and Dec^{C_2}) to capture:

$$\begin{aligned} f^{C_1}(x), p^{C_1}(x) &= Dec^{C_1}(Enc(x)), \\ f^{C_2}(x), p^{C_2}(x) &= Dec^{C_2}(Enc(x)), \end{aligned} \quad (5)$$

where $f^{C_1}(x), f^{C_2}(x) \in R^{D \times H \times W}$ are the semantic feature maps with dimension D , and $p^{C_1}(x), p^{C_2}(x) \in R^{L \times H \times W}$ are the probability predictions. In our paper, D is set to 256, and $L = 19$ is the number of semantic categories. Then, we learn attention weights $\mathcal{W} \in R^{2 \times H \times W}$ with a convolutional block (*Conv*) from $f^{C_1}(x)$ and $f^{C_2}(x)$ to capture the enhanced feature f . In further, we concatenate $f^{C_1}(x)$, $f^{C_2}(x)$, and f together, and feed them to an additional parsing head (*CA*) to secure both $f^{CA}(x)$ and $p^{CA}(x)$. The operations can be expressed as:

$$\begin{aligned} \mathcal{W} &= Conv(f^{C_1}(x) \oplus f^{C_2}(x)), \\ f &= \mathcal{W}^{C_1} \otimes f^{C_1}(x) + \mathcal{W}^{C_2} \otimes f^{C_2}(x), \\ f^{CA}(x), p^{CA}(x) &= CA(f^{C_1}(x) \oplus f^{C_2}(x) \oplus f), \end{aligned} \quad (6)$$

where $\oplus$ is the concatenation operation, $\otimes$ is the element-wise product operation, and $\mathcal{W}^{C_i}$ is the i th channel of $\mathcal{W}$.

Condition-Specific Adversarial Training (CSAT). To optimize the network, we introduce condition-specific adversarial training to encourage the decoders, *i.e.*, Dec^{C_1} , Dec^{C_2} , and *CA*, to focus on different characteristics. This allows alleviating the discrepancy between source and target feature distributions in a progressive manner. Specially, we employ three discriminators (D^{C_1} , D^{C_2} , and D^{CA}) for each decoder, respectively. D^{C_i} addresses the specific condition i , thus operates on $\hat{X}^{S_i}$ and X^{T_i} to match the distributions $p^{C_i}(\hat{X}^{S_i})$ and $p^{C_i}(X^{T_i})$. And D^{CA} targets at transferring knowledge from $\hat{X}^S$ to X^T , thus is a standard domain clas-

sifier. The encoder and decoders are optimized by:

$$\begin{aligned} \mathcal{L}_{adv}(x^t) &= - \sum_{h,w} \log(D^{CA}(p^{CA}(x_{hw}^t))) \\ &\quad - \sum_i \delta(x^t \in X^{T_i}) \sum_{h,w} \log(D^{C_i}(p^{C_i}(x_{hw}^t))), \end{aligned} \quad (7)$$

where $\delta(\cdot)$ is an indicator function which satisfies $\delta(True) = 1$ and $\delta(False) = 0$. Our discriminators follow the PatchGAN (Isola et al. 2017) fashion. Note that, instead of learning source-to-target adaptation, we perform target-to-source adaptation so that we can obtain the adversarial ambivalences for target images. We will introduce the adversarial ambivalence later.

Importantly, with the provided source-domain $\hat{X}^S$ and Y^S pairs, we have a corresponding condition-specific segmentation loss:

$$\begin{aligned} \mathcal{L}_{CE}^S(\hat{x}^s) &= - \sum_{hw} \sum_{l=1}^L [y_{hw}^s \log(p_{hw}^{CA}(\hat{x}^s)) + \\ &\quad \sum_i \delta(\hat{x}^s \in \hat{X}^{S_i}) y_{hw}^s \log(p_{hw}^{C_i}(\hat{x}^s))]. \end{aligned} \quad (8)$$

Self-training

The proposed self-training policy aims to produce reliable pseudo-labels for target images and study adversarial ambivalence to revise the pseudo-labels and emerging the hard adaptation regions.

Attentive Pseudo-Label Assignment (APLA). Benefiting from the attentive progressive adversarial training scheme, we can obtain an adaptation model M_0 . As a result, we can secure pseudo-labels for target images where the correctness is guaranteed to a certain degree. In detail, given a target image x^t , M_0 will produce predictions $p^{C_1}(x^t)$, $p^{C_2}(x^t)$, and $p^{CA}(x^t)$, as well as the attention weights $\mathcal{W}(x^t)$. The attentive pseudo-labels y^t are generated as:

$$\begin{aligned} p(x^t) &= 0.5 * \sum_{i=1}^K \mathcal{W}^{C_i}(x^t) \otimes p^{C_i}(x^t) + 0.5 * p^{CA}(x^t) \\ y_{hw}^t(x^t) &= \begin{cases} l & l = \arg \max_l p_{hw}(l|x^t) \\ & \& p_{hw}(l|x^t) > \lambda_p \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \quad (9)$$

where hw is the spatial location, l represents the category label, and λ_p denotes the pre-defined threshold.

Adversarial Ambivalence. With the pseudo-labels, we can directly transfer knowledge in a more explicit manner, *i.e.*, supervised learning:

$$\tilde{\mathcal{L}}_{CE}^T(x^t) = - \sum_{h,w} \sum_{l=1}^L y_{hw}^t(x^t) \log(p_{hw}^{CA}(x^t)). \quad (10)$$

However, the learned pseudo-labels are with many noises. Several previous works addressed the issue by adding regularization terms (Zou et al. 2019) or discovering hard adaptation images (Shin et al. 2020). For the latter, researchers

Method	Road	SW	Build	Wall	Fence	Pole	Light	Sign	Veg.	Terrace	Sky	Person	Rider	Car	Truck	Bus	Train	Moter	Bike	mIoU
GTA5 → Cityscapes*																				
Source Only	50.1	17.1	51.7	19.7	17.6	20.4	22.5	2.6	76.6	18.1	72.1	56.0	27.1	70.8	4.7	8.7	0.2	11.1	35.1	30.6
AdaptSegNet	80.4	16.4	71.9	14.3	17.2	18.0	30.1	20.3	75.3	15.0	74.8	53.0	26.0	76.0	12.8	32.4	2.7	24.1	29.5	36.3
CLAN	86.6	12.7	74.0	18.6	16.0	22.7	32.6	20.1	73.3	25.8	76.6	54.5	28.0	80.3	24.3	31.8	0.1	18.5	25.8	38.0
AdvEnt	88.2	33.1	78.3	26.0	11.1	31.3	34.7	26.3	74.5	22.0	63.4	55.3	29.9	82.6	25.0	25.0	0.01	20.0	31.7	39.9
CRST	90.6	42.8	78.5	21.0	12.1	35.1	35.8	25.9	69.7	14.6	78.2	61.3	30.2	81.7	16.0	34.8	5.04	28.7	47.5	42.6
CAG-UDA	87.1	37.9	82.7	26.3	29.9	35.1	41.2	39.1	82.3	27.8	76.3	61.6	32.1	83.6	17.8	34.3	18.8	19.2	30.1	45.4
AMEAN++	88.0	41.5	81.1	36.5	27.3	30.7	45.0	27.3	82.6	37.3	73.5	61.7	32.4	85.0	27.6	42.1	13.6	17.5	23.7	46.0
OCDA++	73.5	29.8	67.7	18.7	22.7	26.7	35.9	22.4	81.1	32.3	72.2	48.1	25.6	78.1	11.2	12.8	1.8	20.7	25.8	37.2
MDTA-ITA	80.1	23.6	76.0	17.6	24.2	25.4	34.5	14.2	77.7	18.3	73.5	49.7	12.3	79.7	16.9	22.9	5.1	6.8	10.8	35.2
DCAA	90.5	52.8	84.9	38.7	30.4	37.6	45.5	42.4	84.5	45.5	80.7	63.8	36.2	86.3	34.9	45.1	19.7	14.9	42.5	51.4
SYNTHIA → Cityscapes*																				
Source Only	58.6	21.3	65.7	8.0	0.0	22.3	1.6	9.3	70.1	-	71.7	58.4	25.9	64.1	-	10.7	-	14.1	24.6	32.9
AdaptSegNet	84.7	44.1	74.1	8.7	0.2	17.5	3.8	4.3	71.3	-	69.5	47.8	16.4	62.2	-	23.5	-	10.5	31.5	35.6
CLAN	77.6	31.2	69.9	5.4	0.2	21.8	5.6	6.6	68.6	-	73.9	50.8	19.1	56.6	-	24.4	-	5.4	27.5	34.0
CRST	62.2	23.4	75.6	6.4	1.6	33.8	22.8	30.9	77.8	-	71.5	48.6	24.2	79.5	-	15.2	-	12.9	48.4	39.7
CAG-UDA	71.9	32.8	71.7	19.6	7.7	34.5	0.0	16.6	80.3	-	80.9	51.9	32.5	79.4	-	26.5	-	22.1	51.9	42.5
DCAA	67.7	30.7	74.7	22.9	14.1	33.2	38.5	30.4	82.4	-	81.6	60.1	31.2	80.8	-	40.2	-	14.5	54.7	47.4

Table 1: Performance on Cityscapes*. We reproduce other methods using their released codes following their best configurations. We clarify these reproduced scores under our setting are reliable considering the improvements over the baseline (Source Only). ++: We improve these CDA methods by providing the target sub-domain labels, and using our CGST.

compute the image-level confidence based on the pixel-level pseudo-label confidence, then employ pseudo-label supervision and adversarial learning for easy and hard samples, respectively. In contrast, we exploit the adversarial ambivalences by considering the dense discriminator outputs (PatchGAN) to further regulate the easy and hard adaptation regions. For each target image, we utilize pseudo-labels and the adversarial confidence to supervise the easy adaptation regions in a soft manner, and constraint the hard adaptation regions via adversarial loss. We take the regions where $y_{hw}^t(x^t) = 0$ as the hard adaptation regions.

Specially, we feed $p(x^t)$ into the discriminator D^{CA} to obtain an adaptation confidence map $d(x^t)$. Then, we rectify $\tilde{\mathcal{L}}_{CE}^T(x^t)$ by incorporating pixel-wise weights as:

$$\mathcal{L}_{CE}^T(x^t) = - \sum_{h,w} d_{hw}(x^t) \sum_{l=1}^L y_{hw}^l(x^t) \log(p_{hw}^{CA}(x^t)). \quad (11)$$

We filter out the easy adaptation regions to constraint the remained regions as:

$$\mathcal{L}_{adv}^T(x^t) = - \sum_{h,w} \delta(y_{hw}^t(x^t) = 0) \log(D^{CA}(p^{CA}(x_{hw}^t))). \quad (12)$$

In the self-training stage, $\mathcal{L}_{CE}^S$ and $\mathcal{L}_{CE}^T$ encourage the network produces domain invariant features, and $\mathcal{L}_{adv}^T$ allows the network focusing on pushing hard adaptation regions to the feature space. We set λ_p to 0.6 to generate more pseudo-labels for target images benefiting from the adversarial ambivalence mechanism. Besides, we adopt M_0 (with fixed parameters) to produce pseudo-labels and adversarial ambivalences. We learn a new model M_1 that shares the same network architecture with M_0 . The parameters are ini-

tialized via the pre-trained M_0 .

Experiments

We refer to the supplemental materials for the implementation details and more analyses of our approach. Our codes will be made public available.

Adaptation Scenarios. We evaluate our proposed approach on three challenging adaptation scenarios, *i.e.*, GTA5→Cityscapes* and SYNTHIA→Cityscapes*, and GTA5→BDD100K. Cityscapes* is a combined benchmark consisting of Cityscapes-cloudy (the commonly used Cityscapes dataset), Cityscapes-Foggy, and Cityscapes-Rainy. Specifically, Cityscapes-Cloudy provides 3,475 images with a resolution of 2048×1024 , which are officially split into 2,975 training images and 500 validation images. The images in Cityscapes-Foggy are captured by simulating different levels of foggy to images in Cityscapes-Cloudy. Cityscapes-Rainy selects 262 training images and 33 validation images in Cityscapes-Cloudy to attach 36 degrees of rain to generate 9,432\1,118 realistic rainy scenes. We randomly collect 12 rainy images for each reference image to construct our final Cityscapes-Rainy, *i.e.*, 3,144 images for training, and 396 images for evaluation. The enlarged target benchmark (Cityscapes*) is now has 9,094 (2975 + 2975 + 3144) images for training and 1,396 (500 + 500 + 396) images for evaluation. BDD100K is constructed for heterogeneous multitask learning where the images are with diverse weather conditions. We take BDD100K: {Rainy, Snowy, Cloudy} as our target domain following OCDA (Liu et al. 2020). There are 4,855\215, 5,307\242, and 4,535\346 training\test images for the three weather conditions, respectively. We also evaluate our trained model on the unseen BDD100K-Overcast domain (627 images).

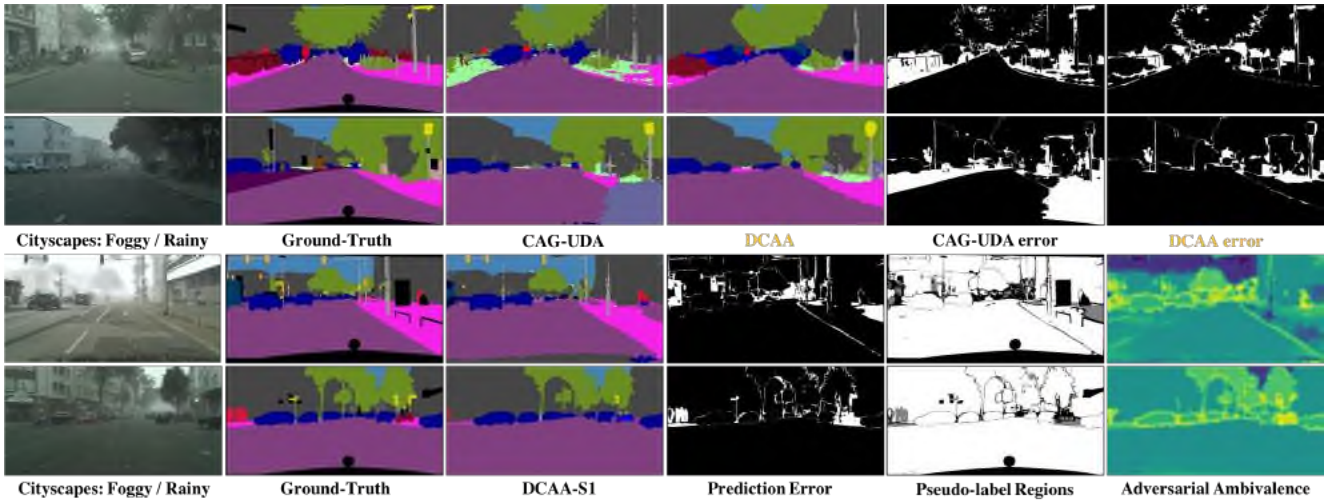


Figure 4: Qualitative Results. We refer to Figure 1 for the explanations of these maps. From the top part, our predictions contain more details compared with CAG-UDA. From the bottom part, the adversarial ambivalence is potent to indicate the incorrectly predicted regions and revise the pseudo-label related constraints.

The synthetic dataset GTA5 contains 24,966 images with a size of 1914×1052 , and has the consistent 19 semantic categories as Cityscapes. SYNTHIA is another synthetic benchmark that offers 9,400 images of size 1914×1052 . We use the 16 common category labels as Cityscapes for evaluation.

Compared Methods. We reproduce several widely investigated SOTA methods, including AdaptSegNet (Tsai et al. 2018), CLAN (Luo et al. 2019), CRST (Zou et al. 2019), ADVENT (Vu et al. 2019), and CAG-UDA (Zhang et al. 2019) using their shared codes and their best configurations for benchmark performance reporting. Directly evaluate their methods using the public models will result in lower scores since their models are trained only for Cityscapes-Cloudy. Moreover, we make the sub-domain labels available to improve the open domain adaptation approaches OCDA (Liu et al. 2020) and AMEAN (Chen et al. 2019c). We also compare our DCAA with a SOTA multi-target domain adaptation method MDTA-ITA (Gholami et al. 2020).

General Performance. The scores are reported in Table 2 and Table 1. Generally, DCAA ranks 1st on all the stud-

Method	Rainy	Snowy	Cloudy	Avg.	Overcast
Source Only	16.2	18.0	20.9	18.9	21.2
AdaptSegNet	20.2	21.2	23.8	22.1	25.1
CBST	21.3	20.6	23.9	22.2	24.7
IBN-Net	20.6	21.9	26.1	22.8	25.5
PyCDA	21.7	22.3	25.9	23.3	25.4
OCDA	22.0	22.9	27.0	24.5	27.9
OCDA++	23.1	22.8	29.2	25.7	29.8
MDTA-ITA	22.6	23.2	27.5	24.9	24.9
DCAA	31.0	34.6	37.4	34.7	38.3

Table 2: GTA5→BDD100K. During the training process, the target sub-domains consists of “Rainy”, “Snowy”, and “Cloudy”, while “Overcast” is unseen.

ied adaptation scenarios. For GTA5→BDD100K, DCAA achieves consistently promising improvements (about 9.0%) over all the target sub-domains, even for the unseen “Overcast” images. The results demonstrate that our approach can effectively exploit multi-modalities’ characteristics and is potent to produce high-performing adaptation models. For GTA5→Cityscapes*, DCAA outperforms the baseline (Source Only) by a large margin (51.4% vs. 30.6%), while yields a significant improvement (6.0%) over the SOTA method CAG-UDA. Besides, we find that OCDA and MDTA-ITA can not perform well on the segmentation adaptation scenarios. For example, OCDA only captures slight improvements over the AdaptSegNet baseline (1.1% ~ 2.4%) as reported in Table 1 and Table 2. A possible reason is that they focuses on disentangling the domain-style codes from the content codes. It may easy to disentangle global style codes for image classification problems as studied in their paper. However, extracting content independent pixel-level style codes would be more challenging. For SYNTHIA→Cityscapes*, DCAA produces appealing semantic parsing for target images with a remarkable mean IoU (47.4%). Moreover, we present some qualitative results in Figure 4. Our DCAA can preserve more details in boundary areas and revise some incorrectly labeled regions predicted by other methods, especially for images with bad weather conditions. Furthermore, we find that the mIoU reported in previous publications (training and evaluating on Cityscapes-Cloudy) are consistently higher (about 5%) than the reproduced versions (training and evaluating on Cityscapes*). The large performance gap shows that trivially learning an adaptation model without explicitly exploiting the diverse weather conditions may not sufficiently discovering their correlations.

Method	Road	SW	Build	Wall	Fence	Pole	Light	Sign	Veg.	Terrace	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
Condition Attention Module (Stage1 Score)																				
Mix-Head	88.0	41.5	81.1	36.5	27.3	30.7	45.0	27.3	82.6	37.3	73.5	61.7	32.4	85.0	27.6	42.1	13.6	17.5	23.7	46.0
Sep-Head	83.6	41.1	76.9	31.8	29.4	30.5	44.2	30.8	83.1	28.6	77.4	60.7	32.4	84.9	25.0	40.5	11.0	17.9	30.2	45.3
SM-Head	85.1	41.3	79.0	33.7	30.2	31.2	45.3	30.3	82.7	30.1	75.8	61.2	31.9	84.8	25.1	39.1	9.6	19.8	29.5	45.6
CAM	89.9	46.4	82.5	37.7	29.3	34.1	45.7	25.4	83.1	33.5	76.8	62.3	32.9	85.3	28.3	39.9	2.6	23.3	37.4	47.2
Condition-Specific Adversarial Training (Building on CAM)																				
DAT-S1	84.0	38.0	80.1	28.2	29.8	28.9	44.9	31.7	82.7	28.9	77.4	61.2	30.7	82.2	22.2	30.4	17.3	22.8	39.8	45.3
CSAT-S1	89.9	46.4	82.5	37.7	29.3	34.1	45.7	25.4	83.1	33.5	76.8	62.3	32.9	85.3	28.3	39.9	2.6	23.3	37.4	47.2
DAT-S2	88.3	42.0	83.7	37.3	26.8	27.4	43.1	34.8	85.5	33.1	79.2	60.8	30.1	85.2	29.9	41.8	14.7	23.3	41.6	47.8
CSAT-S2	90.5	52.8	84.9	38.7	30.4	37.6	45.5	42.4	84.5	45.5	80.7	63.8	36.2	86.3	34.9	45.1	19.7	14.9	42.5	51.4
Attentive Pseudo-Label Assignment ($\lambda_p = 0.9, \tilde{\mathcal{L}}_{CE}^T$) (Baseline = DCAA-S1)																				
Baseline	89.9	46.4	82.5	37.7	29.3	34.1	45.7	25.4	83.1	33.5	76.8	62.3	32.9	85.3	28.3	39.9	2.6	23.3	37.4	47.2
MaxV	89.8	44.1	83.2	34.7	29.3	29.5	42.7	30.9	86.0	44.3	80.6	60.7	31.2	86.2	32.2	30.1	17.7	26.1	34.7	48.1
MeanV	89.6	48.4	83.6	38.8	30.1	27.1	38.9	28.4	86.1	43.0	79.8	62.4	29.4	85.8	37.0	32.7	16.7	32.7	34.9	48.7
APLA ⁺	90.5	47.0	83.7	35.0	17.0	31.2	43.6	36.6	85.9	44.3	77.7	63.8	34.3	88.2	30.6	38.7	8.6	34.9	54.9	49.8
Adversarial Ambivalence + APLA ($\lambda_p = 0.6$)																				
Baseline ($\tilde{\mathcal{L}}_{CE}^T$)	88.2	44.8	82.5	36.0	22.5	34.1	43.0	36.8	84.5	44.9	77.5	62.0	25.7	86.2	29.5	37.6	1.1	29.6	50.7	48.3
$\mathcal{L}_{CE}^T$	90.6	50.9	84.0	41.0	22.1	36.1	45.8	34.3	85.1	36.4	79.4	64.5	37.7	85.0	33.0	46.5	18.9	22.3	47.3	50.6
$\mathcal{L}_{CE}^T + \mathcal{L}_{adv}^T$	90.5	52.8	84.9	38.7	30.4	37.6	45.5	42.4	84.5	45.5	80.7	63.8	36.2	86.3	34.9	45.1	19.7	14.9	42.5	51.4

Table 3: Ablation Studies. Si: The i th stage results. APLA⁺: When studying APLA, we take λ_p as 0.9 to avoid introducing too many noisy pseudo-labels. MaxV and MeanV mean that we obtain the pseudo-labels via the simple max voting and mean voting operations.

Ablation Studies

We conduct ablation experiments based on the adaptation scenario GTA5→Cityscapes* to discuss our DCAA. We refer to the supplemental materials for other studies.

Condition Attention Module. We examine some other possible options that explicitly incorporate the weather priors, including Mix-Head, Sep-Head, and SM-Head. The differences are shown in Figure 3. From Table 3, either SM-head (45.6%) or Sep-Head (45.3%) even slightly decreases the performance of Mix-Head by 0.4% ~ 0.7%. This observation explicates that trivially incorporate weather priors may show negative influences. Instead, CAM (47.2%) effectively exploits the special characteristics and correlations of multi-modalities.

Condition-Specific Adversarial Training. We study the effectiveness of the standard adversarial training strategy and our progressive condition-specific adversarial training policy in minimizing the source and target discrepancy. In Table 3, DAT denotes that all the discriminators are standard domain classifiers to distinguish source features from target features. We find that the second-stage DAT-S2 (47.8%) only reaches a slight better mIoU over our first-stage CSAT-S1 (47.2%), and is marginally below CSAT-S2 (51.4%). This observation demonstrates that directly aligning source and target feature distributions is challenging when the target domain is heterogeneous.

Self-Training. We make comparisons with different baselines and variants to discuss our self-training policy. The scores and compared methods are presented in Table 3. We find that the produced pseudo-labels by MaxV and MeanV

can improve the baseline by 0.7% ~ 1.3%, while our APLA⁺ impressively yields a mIoU of 49.8% (+2.7%). This observation shows that trivial pseudo-label generation strategies may not sufficiently exploit the complementarities of multi-modal semantics. Furthermore, the revised $\tilde{\mathcal{L}}_{CE}^T$ outperforms $\tilde{\mathcal{L}}_{CE}^T$ by a convincing margin (+2.3%) when the pseudo-labels are with many noises. At the same time, the proposed adversarial training for hard adaptation regions ($\mathcal{L}_{adv}^T$) can further boost the mIoU to 51.4%. The improvements demonstrate that the proposed adversarial ambivalence is potent to suppress the adverse impact of noisy pseudo-labels. It can also drive the network focusing on pushing hard adaptation regions to a shared feature space.

Conclusion

In the paper, we notice that images in a target domain may show diverse characteristics in typical domain adaptation studies. We clarify carefully reason about the multi-modalities may enhance the adaptation performance. Thus, we propose a novel adaptation approach DCAA that is empowered by an attentive progressive adversarial training mechanism and a self-training policy. The former migrate the source and target discrepancies in a progressive manner. The later first produces pseudo-labels for target images by investigating the correlations among target sub-domains, then exploiting the adversarial ambivalence to regularize the noisy pseudo-labels and constraint the hard adaptation regions. We conduct various comparisons and ablation studies to discuss our methods intensely. Experimental results on several challenging adaptation scenarios demonstrate the superiority of method in capturing high-performing models.

References

- Chen, C.; Chen, Z.; Jiang, B.; and Jin, X. 2019a. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3296–3303.
- Chen, M.; Xue, H.; and Cai, D. 2019. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE International Conference on Computer Vision*, 2090–2099.
- Chen, Y.-C.; Lin, Y.-Y.; Yang, M.-H.; and Huang, J.-B. 2019b. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1791–1800.
- Chen, Z.; Zhuang, J.; Liang, X.; and Lin, L. 2019c. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2248–2257.
- Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; and Choo, J. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8789–8797.
- Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Du, L.; Tan, J.; Yang, H.; Feng, J.; Xue, X.; Zheng, Q.; Ye, X.; and Zhang, X. 2019. SSF-DAN: Separated Semantic Feature Based Domain Adaptation Network for Semantic Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 982–991.
- Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Zhang, K.; and Tao, D. 2019. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2427–2436.
- Gholami, B.; Sahu, P.; Rudovic, O.; Bousmalis, K.; and Pavlovic, V. 2020. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing* 29: 3993–4002.
- Gong, M.; Xu, Y.; Li, C.; Zhang, K.; and Batmanghelich, K. 2019a. Twin auxiliary classifiers gan. In *Advances in neural information processing systems*, 1330–1339.
- Gong, R.; Li, W.; Chen, Y.; and Gool, L. V. 2019b. DLOW: Domain flow for adaptation and generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2477–2486.
- Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, 1989–1998. PMLR.
- Hoffman, J.; Wang, D.; Yu, F.; and Darrell, T. 2016. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*.
- Hong, W.; Wang, Z.; Yang, M.; and Yuan, J. 2018. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1335–1344.
- Hu, X.; Fu, C.-W.; Zhu, L.; and Heng, P.-A. 2019. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8022–8031.
- Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.
- Kim, M.; and Byun, H. 2020. Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12975–12984.
- Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.
- Li, G.; Kang, G.; Liu, W.; Wei, Y.; and Yang, Y. 2020. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision*, 440–456. Springer.
- Li, Y.; Yuan, L.; and Vasconcelos, N. 2019. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6936–6945.
- Liu, Z.; Miao, Z.; Pan, X.; Zhan, X.; Lin, D.; Yu, S. X.; and Gong, B. 2020. Open Compound Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12406–12415.
- Luo, Y.; Zheng, L.; Guan, T.; Yu, J.; and Yang, Y. 2019. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2507–2516.
- Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- Pan, F.; Shin, I.; Rameau, F.; Lee, S.; and Kweon, I. S. 2020. Unsupervised Intra-domain Adaptation for Semantic Segmentation through Self-Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3764–3773.
- Pei, Z.; Cao, Z.; Long, M.; and Wang, J. 2018. Multi-adversarial domain adaptation. *arXiv preprint arXiv:1809.02176*.
- Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V. 2016. Playing for Data: Ground Truth from Computer Games. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *European Conference on Computer Vision (ECCV)*, volume 9906 of LNCS, 102–118. Springer International Publishing.
- Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and

- Lopez, A. M. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3723–3732.
- Sakaris, C.; Dai, D.; Hecker, S.; and Van Gool, L. 2018. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 687–704.
- Shin, I.; Woo, S.; Pan, F.; and Kweon, I. S. 2020. Two-phase Pseudo Label Densification for Self-training based Domain Adaptation. In *European Conference on Computer Vision*, 532–548. Springer.
- Sun, R.; Zhu, X.; Wu, C.; Huang, C.; Shi, J.; and Ma, L. 2019. Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4360–4369.
- Tsai, Y.-H.; Hung, W.-C.; Schuster, S.; Sohn, K.; Yang, M.-H.; and Chandraker, M. 2018. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7472–7481.
- Tsai, Y.-H.; Sohn, K.; Schuster, S.; and Chandraker, M. 2019. Domain adaptation for structured output via discriminative patch representations. In *Proceedings of the IEEE International Conference on Computer Vision*, 1456–1465.
- Vu, T.-H.; Jain, H.; Bucher, M.; Cord, M.; and Pérez, P. 2019. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2517–2526.
- Wang, D.; Cui, P.; and Zhu, W. 2018. Deep Asymmetric Transfer Network for Unbalanced Domain Adaptation. In *AAAI*.
- Wang, Z.; Yu, M.; Wei, Y.; Feris, R.; Xiong, J.; Hwu, W.-m.; Huang, T. S.; and Shi, H. 2020. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12635–12644.
- Wu, Z.; Han, X.; Lin, Y.-L.; Gokhan Uzunbas, M.; Goldstein, T.; Nam Lim, S.; and Davis, L. S. 2018. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 518–534.
- Xiao, J.; Gu, S.; and Zhang, L. 2020. Multi-Domain Learning for Accurate and Few-Shot Color Constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3258–3267.
- Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 418–434.
- Yang, J.; Xu, R.; Li, R.; Qi, X.; Shen, X.; Li, G.; and Lin, L. 2019. An Adversarial Perturbation Oriented Domain Adaptation Approach for Semantic Segmentation. *arXiv preprint arXiv:1912.08954*.
- Yang, Y.; and Soatto, S. 2020. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4085–4095.
- Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020a. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2636–2645.
- Yu, X.; Liu, T.; Gong, M.; Zhang, K.; Batmanghelich, K.; and Tao, D. 2020b. Label-noise robust domain adaptation. In *International Conference on Machine Learning*, 10913–10924. PMLR.
- Zhang, Q.; Zhang, J.; Liu, W.; and Tao, D. 2019. Category Anchor-Guided Unsupervised Domain Adaptation for Semantic Segmentation. In *Advances in Neural Information Processing Systems*, 433–443.
- Zhang, Y.; David, P.; and Gong, B. 2017. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 2020–2030.
- Zhang, Y.; Qiu, Z.; Yao, T.; Liu, D.; and Mei, T. 2018. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6810–6818.
- Zhang, Y.; and Wang, Z. 2020. Joint Adversarial Learning for Domain Adaptation in Semantic Segmentation. In *AAAI*, 6877–6884.
- Zhao, S.; Gong, M.; Liu, T.; Fu, H.; and Tao, D. 2020. Domain Generalization via Entropy Regularization. volume 33.
- Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.
- Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017b. Toward multimodal image-to-image translation. In *Advances in neural information processing systems*, 465–476.
- Zou, Y.; Yu, Z.; Liu, X.; Kumar, B.; and Wang, J. 2019. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, 5982–5991.
- Zou, Y.; Yu, Z.; Vijaya Kumar, B.; and Wang, J. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, 289–305.

Personalized Adaptive Meta Learning for Cold-start User Preference Prediction

Runsheng Yu,¹ Yu Gong,² Xu He,¹ Bo An,¹ Yu Zhu,² Qingwen Liu,² Wenwu Ou²

¹ School of Computer Science and Engineering, Nanyang Technological University

² Alibaba Group

runshengyu@gmail.com, gongyu.gy@alibaba-inc.com, hexu0003@e.ntu.edu.sg, boan@ntu.edu.sg, zy143829@alibaba-inc.com, xiangsheng.lqw@alibaba-inc.com, santong.oww@taobao.com

Abstract

A common challenge in personalized user preference prediction is the cold-start problem. Due to the lack of user-item interactions, directly learning from the new users' log data causes serious over-fitting problem. Recently, many existing studies regard the cold-start personalized preference prediction as a few-shot learning problem, where each user is the task and recommended items are the classes, and the gradient-based meta learning method (MAML) is leveraged to address this challenge. However, in real-world application, the users are not uniformly distributed (i.e., different users may have different browsing history, recommended items, and user profiles). We define the major users as the users in the groups with large numbers of users sharing similar user information, and other users are the minor users), existing MAML approaches tend to fit the major users and ignore the minor users. To address this cold-start task-overfitting problem, we propose a novel personalized adaptive meta learning approach to consider both the major and the minor users with three key contributions: 1) We are the first to present a personalized adaptive learning rate meta-learning approach to improve the performance of MAML by focusing on both the major and minor users. 2) To provide better personalized learning rates for each user, we introduce a similarity-based method to find similar users as a reference and a tree-based method to store users' features for fast search. 3) To reduce the memory usage, we design a memory agnostic regularizer to further reduce the space complexity to constant while maintain the performance. Experiments on MovieLens, BookCrossing, and real-world production datasets reveal that our method outperforms the state-of-the-art methods dramatically for both the minor and major users.

1 Introduction

Recommender Systems (RS) help people to discover the items they prefer (Guo et al. 2017; Qu et al. 2016). In order to train a well-performing personalized user preference predictor, enough interactions with users are indispensable. To address this challenge, many researchers take advantage of the offline supervised training methods, which leverage the historical data (log data) to train the model. However, when the log data is lacking, these supervised training methods cause the over-fitting issues (Vanschoren 2018), which is known as the *cold-start problem*.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

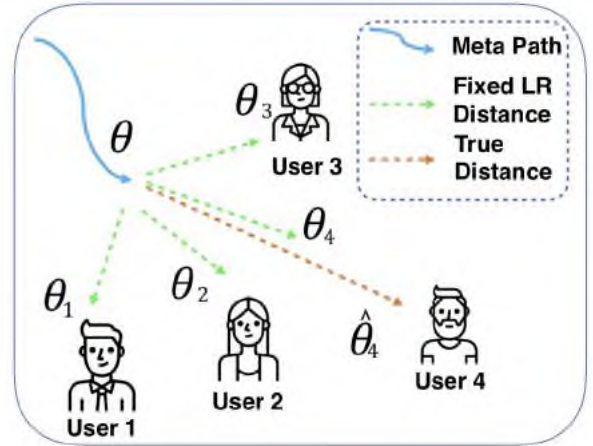


Figure 1: An example illustrating that MAML has limitations in the cold-start user imbalanced dataset. Users 1, 2, 3 are the major users and user 4 is the minor user.

To train a well-performing model in the cold-start problem, meta learning-based approaches are introduced (Lee et al. 2019; Dong et al. 2020). Intuitively, learning with a few samples can be viewed as a few-shot learning problem and meta learning, especially gradient-based meta learning (e.g., Model Agnostic Meta Learning, MAML (Finn, Abbeel, and Levine 2017)), which aims to adapt to any task with only a few steps of parameter update, has been proven as one of the most successful approaches for these problems. Thanks to its good generalization ability, MAML has already been leveraged into various domains, including computer vision, natural language process, and robotics (Gui et al. 2018; Madotto et al. 2019; Finn et al. 2017). Recently, in the RS area, meta learning is introduced for the cold-start problem for either users or items, which treats the users/items as tasks, log data as samples, and learns to do fast adaptation when meeting new tasks (Dong et al. 2020; Pan et al. 2019; Lee et al. 2019; Luo et al. 2020).

However, these MAML methods assume that the distribution of tasks (users) is uniform. But in the recommendation systems, the user distribution is not always balanced, i.e., the user profile, browsing history, as well as the features of

recommended items are not always uniformly distributed¹, which harms the performance of MAML. For example, regarding the user profile, as shown in Tab. 1, most values are imbalanced. In terms of the browsing histories, the distribution of the browsing histories is also non-uniform (Pan et al. 2019). Therefore, simply learning from these data will overfit the users with similar feature values since these users own the major similar features and fitting these users can already achieve good-enough *average* prediction performance (like the MSE results in Tab. 1), which is known as the cold-start user (task) overfitting problem. To illustrate the problem clearly, we regard a collection of users with similar feature values as a group and define the users as the major users when the number of users in these groups are large, and others users are the minor users. Now, we give an example to illustrate how imbalanced distribution harms the performance of MAML: as shown in Fig. 1, assuming that an MAML strategy aims to learn to do fast adaptation to the four cold-start users (three major users (users 1-3) and one minor user (user 4)). Different locations in the blue square indicate feature values (embeddings) for different users. Since three of them are near each other (have similar feature values), the model may *not* focus on the minor user (user 4) because fitting the major users can already achieve good prediction performance.

To address these limitations, in this paper, we propose a personalized adaptive meta learning approach to improve the performance of MAML in the cold-start user preference prediction problem. The main idea is to set different learning rates for different users so as to find task-adaptive parameters for each user. Our main contributions are as follows: 1) We are the first to introduce a novel user-adaptive learning rate based meta learning in RS to improve the performance of MAML by focusing on both the major and minor users. 2) To provide better-personalized learning rates, we introduce a method to find similar users as a reference and a tree-based method to store users’ features for fast search. 3) To reduce the memory usage, we introduce a memory agnostic regularizer to reduce the space complexity while maintaining good prediction performance. Experiments on MovieLens-1M, BookCrossing, and real-world production datasets from one of the largest e-commerce platforms reveal that our method outperforms state-of-the-art methods dramatically for both the minor and the major users.

2 Related Works

In this section, we discuss some related works including gradient-based meta learning for the imbalanced dataset and the meta learning for cold-start recommendation..

Gradient-Based Meta Learning for Task Overfitting Problem. MAML based methods have been widely adopted for studying the few-shot learning problem (Finn, Abbeel, and Levine 2017; Li et al. 2017; Xu, van Hasselt, and Silver 2018; Chen et al. 2018; Ravi and Larochelle 2016; Lee and Choi 2018). To consider task-adaptive challenges, the vector of learning rates, the block-diagonal preconditioning matrix, latent embedding, and interleaving warp-layers are designed

¹They are collectively named as the feature values.

The ratio of features				The major users		The minor users	
Age	Gender	Zipcode	Occup	Ratio	MSE	Ratio	MSE
72.4%	71.7%	56.7%	64.0%	0.648	1.413	0.352	1.506

Table 1: The ratio ($\frac{\text{the number of users having certain feature values}}{\text{total number of users}}$) of the top 30% largest number of feature values that users own², ratio of certain users ($\frac{\text{the number of users in certain group}}{\text{total number of users}}$), and the Mean Squared Errors (MSE) from one of the MAML methods (Lee et al. 2019) in MovieLens. The method used to split the cold-start major and minor users can be found in Sec. 5.

to learn parameters for different tasks respectively (Li et al. 2017; Park and Oliva 2019; Rusu et al. 2018; Flennerhag et al. 2020). However, those methods still do not explicitly consider the task overfitting problem. To take this problem into consideration, Bayesian TAML introduces different initiation parameters and inner gradient approaches to handle class imbalance, task imbalance, as well as out-of-distribution tasks together by higher-order statistics (Flennerhag et al. 2020). However, in RS, a users’ feature space is high-dimensional and those statistics are still hard to represent the embedding completely. As far as we know, we are the *first* to propose a novel meta learning method focusing on the two key challenges in cold-start RS: 1) comparing with the common-seen few-shot learning problem which does not explicitly have the task information, the dimension of features for the task information (like the user profile) in our problem is large, and 2) task does not obey the uniform distribution (even long-tail) while the standard few-shot learning problem is assumed uniformly distributed.

Meta Learning for Cold-start Recommendation. In RS, meta learning is mainly used to do fast adaptation with only few samples for users and items, including the item cold-start/warm-up problem (Vartak et al. 2017; Pan et al. 2019) and the user cold-start problem (Dong et al. 2020; Lee et al. 2019; Bharadhwaj 2019). For the item cold-start problem, linear weight adaptation is introduced to represent the items by the users who have viewed it (Vartak et al. 2017) while Meta-Embedding is given to generate different embeddings for cold-start processes (Pan et al. 2019). For the user cold-start approaches, MAML is used to address the challenge by learning fast adaptation to new users. But those approaches do not consider the task overfitting problems. Differently, we are the *first* to investigate the personalized learning rate based meta learning to address the cold-start user overfitting problem in RS, considering both major and minor users together.

3 Problem Formulation

This section discusses the gradient-based meta learning in RS and how user imbalance affects the performance of MAML. Formally, following (Lee et al. 2019; Finn, Abbeel, and Levine 2017), the objective of meta learning in RS scenario can be described as: $\arg \min_{\theta} \left[\frac{1}{|\tau|} \sum_{\tau_i \in \tau} L_{\tau_i}(\theta_i) \right]$ s.t. $\theta_i = \theta - \alpha \nabla_{\theta} L_{\tau_i}(\theta)$, where i is the i -th index for user $u_i \in \mathcal{U}$ (or task, since

each user is treated as a task in our RS settings), L_{τ_i} is the loss function for user i (e.g., mean squared error) and the item-feedback set $\tau_i = \{\langle \text{item}_i^j, \text{score}_i^j \rangle_{j=1}^{\hat{N}}\}$ is the set of items $\langle \text{item}_i^j \rangle_{j=1}^{\hat{N}}$ ($\hat{N}$ items in total) and their corresponding feedbacks $\langle \text{score}_i^j \rangle_{j=1}^{\hat{N}}$ (e.g., ranking score) by user u_i . L_τ is the loss for different tasks and $\tau \supseteq \tau_i$ is the union set of the item-feedback subsets for all the users. α is the (inner) Learning Rate (LR). For simplification, we use L_τ to replace $\frac{1}{|\tau|} \sum_{\tau_i \in \tau} L_{\tau_i}$. Before we discuss the user overfitting problem, we present a special case that helps us to analyze the problem better: users are clustered into two groups with densities p_1, p_2 ($p_1 \geq p_2$). Users' actual preferences are x_1, x_2 for group 1 and 2 respectively. Now, we show that MAML ignores the minor users in this case:

Lemma 1. Suppose that the loss function is defined as $L = \sum_{i=1}^2 p_i (\theta_i - x_i)^2$, where x_1, x_2 are the actual preferences for task 1 and 2, and $\theta_i = \theta - \alpha \nabla_{\theta} L_{\tau_i}(\theta)$, when $p_1 \geq p_2$, $x_1 \geq x_2$, we have $(\theta_1^* - x_1)^2 \leq (\theta_2^* - x_2)^2$, where θ^* is the optimal parameter θ .

All the proofs can be found in Appendix A. This case indicates that the MAML method tends to optimize the major users preferentially.

4 Methodology

This section presents a practical adaptive Learning Rate (LR)-based meta-learning method to address the user overfitting problem, including 1) why adaptive LR can solve the user overfitting problem and how to apply adaptive LR by end-to-end training, 2) a similarity-based learning approach to improve the performance by considering similar users and an approximated tree-based implementation for fast search, and 3) a memory agnostic regularizer to further reduce space complexity to constant.

4.1 Adaptive Learning Rate based MAML

To consider both major and minor users together, we propose an adaptive learning rate approach to design different gradient steps for different users :

$$\arg \min_{\theta} \left[\frac{1}{|\tau|} \sum_{\tau_i \in \tau} L_{\tau_i}(\theta_i) \right], \text{ s.t. } \theta_i = \theta - \alpha(h_i) \nabla_{\theta} L_{\tau_i}(\theta), \quad (1)$$

where $h_i \in \mathbb{H}$ is the i -th user's feature embeddings³ (or user embedding for short). The difference between ours and MAML is that the LR in our method is a mapping from the user embedding to a real number rather than a fixed LR. Intuitively, with an adaptive learning rate, the meta agent can fit any user even if it is far from the meta strategy (like the user 4 in Fig. 1). Here, an analysis is given to illustrate the adaptive learning rate can get better results in user imbalanced dataset:

Lemma 2. Based on Lemma 1, we further defined as $L' = \sum_{i=1}^2 p_i (\theta'_i - x_i)^2$, s.t. $\theta'_i = \theta - \alpha(h_i) \nabla_{\theta} L_{\tau_i}(\theta)$, (x_1, x_2

³User's feature embeddings can include different user information like the user profile embeddings and user browsing history embeddings.

are the target values for task 1 and 2, where $x_2 > x_1$) there exists α_1 and α_2 that satisfying

$$(\theta_2^* - x_2)^2 \geq (\theta_1^* - x_1)^2 \text{ and } L^* \geq L',$$

where L^* and L' are the optimal loss functions for standard MAML and adaptive MAML respectively. θ^* is the optimal θ' parameters.

Lemma 2 indicates that learning a personalized adaptive learning rate provides a lower loss value for minor users than the standard MAML method, and is able to achieve a lower total loss value than standard method ($L^* \geq L'$). This coincides with our conjecture that the adaptive LR meta learning methods perform better than the fixed LR methods when facing an imbalanced distribution.

Therefore, the main challenge is to capture the relationship among the user distribution, the parameters of network, and user own learning rate. To construct it, we leverage the end-to-end data-driven learning approach to map each user's features into different high-dimensional embedding and obtain the individual learning rate adaptively. Based on this idea, we propose the Personalized Adaptive Meta Learning (PAML) method, a direct way to train the network by taking the gradient descent of Eq. (1):

$$\begin{aligned} \theta &= \theta - \beta \sum_{\tau_i} \nabla_{\theta} L_{\tau_i}(\theta_i) \left(I - \alpha(h_i; \psi) \nabla_{\theta}^2 L_{\tau_i}(\theta) \right), \\ \psi &= \psi + \beta \nabla_{\psi} L_{\tau}(\theta) \left(\sum_{\tau_i} \nabla_{\psi} \alpha(h_i; \psi) L_{\tau_i}(\theta) \right). \end{aligned}$$

where θ and ψ are the parameters for the model and learning rate⁴. β is the outer learning rate. The two equations above are drawn by the chain rule.

4.2 Approximated Tree-based PAML

The key for PAML is to get accurate personalized $\alpha(h)$ for different users. However, directly using Fully-Connected (FC) layers to learn $\alpha(h_i)$ is hard since LR is related to the task distribution and FC layers are not capable to memorize a large number of the users. To address the challenge, one straightforward idea is to consider other similar users' features as a reference since users with similar feature embeddings share similar LR. Based on this idea, we introduce the similarity-based method to find users with similar high-level features (the embedding) and interests.

Formally speaking, when a new user u_i with embedding $h_i \in \mathbb{H}$ comes, our goal is to find the users with the most similar embeddings to u_i . Here, we define a similarity function $s_k : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ to estimate the embedding similarity between user u_i and user u_k . After computing top- K nearest (the most similar) users with the values of similarity, these values can be treated as a reference to obtain a personalized adaptive learning rate. Thus, we can get an LR function for a new user u_i : $\alpha(h_i) := \alpha'(h_i) + \tilde{\alpha}$, by considering the user information and the user similarities together, where $\tilde{\alpha} = \sum_k \frac{s_k}{\sum_j s_j + \sigma} \alpha(h_k)$ is the weighted average sum

⁴For simplification, we use $\alpha(h_i)$ to replace $\alpha(h_i; \psi)$ if not specially mentioned.

of the nearest users' LR, s_k is the k -th existing nearest user's similarity value to user i , $\alpha'(h_i)$ is neural network modules mapping from user embedding to a real value, and σ is a small value (10^{-5}). Here, we set the Gaussian kernel function $s_k = \exp(-\delta \|h_i - h_k\|^2)$ as the similarity function, where δ is a constant and $\|\cdot\|$ is the 2-norm. Due to the limited space, details about how the kernel-based function can be used to estimate the similarity can be found in Appendix B.

Since we need to find the top- K nearest users in every training step, it is necessary to find a fast searching approach to accelerate the training process. Here, we leverage the kd-tree (Muja and Lowe 2014) as the basic structure to store users' embeddings. Specifically, we first initialize the tree structure with several users' ($\geq K$) embeddings and its corresponding LR (the warm-up stage). Then, when a new user u_i with embedding h_i comes, we search the K -th nearest users in the tree. After that, we add the new user embedding and its LR into that tree and rebuild that tree. We remove the least frequent used user's embedding and its LR when the tree-size is larger than the memory size we set manually. Based on this structure, the time complexity of searching each user can be reduced from $\mathcal{O}(n)$ (brute force method) to $\mathcal{O}(\sqrt{n})$ for two-dimension embedding cases (Yianilos 1993).

However, this structure induces bias because the user embedding layers are dynamically updated during the training process but the previous users' embeddings stored in the tree nodes are fixed, causing the new embeddings and the old embeddings unfitted. In order to coordinate them, we also let the tree nodes be dynamically updated by gradient descent: $\text{node}_j = \text{node}_j - \beta \nabla_{\text{node}_j} L_\tau$, where node_j indicates the j -th node value (the user's embedding). The tree-based structure will also be updated once the embeddings change. The learning-rates stored in nodes are also updated similarly.

To further reduce space usage and speed up, we use the approximated store and search methods in place of the exact (precious) ones. That is, we can take advantage of the randomized kd-tree algorithm (Muja and Lowe 2014) to achieve the approximation. These structures can be easily implemented by the open-source python package (e.g., Pyflann (Muja and Lowe 2013)). We name it as Approximated-Tree PAML (AT-PAML). The loss function is the same as Eq. (1) and the pseudo-code can be found in Algs. 1 and 2.

4.3 Regularizer-based PAML

However, for large-scale real-world applications (e.g., Netflix) which involve more than millions of users, even the linear space complexity methods (the space size is proportional to the number of user embeddings) to store user embedding is unacceptable. Thus, a constant space complexity algorithm needs to be proposed. A widely adopted method to achieve long-term memorization is the continual learning approach (Aljundi et al. 2018; Kirkpatrick et al. 2017), which designs losses and training schemes to memorize the previous tasks. Inspired by their works, we design an auxiliary loss function to help the model to remember the users it has seen. That is, for any two users u_i and u_j , memorizing the information of those two users means the loss values for those

Algorithm 1: Personalized Adaptive Meta Learning

Input: User distribution $p(\mathcal{U})$, the learning rate β .

- 1 Initialize the meta-policy with parameters θ ;
- 2 **for** episode T **do**
- 3 sample N of users from $p(\mathcal{U})$;
- 4 **for** each user $i \in U$ **do**
- 5 Split the support set τ_i and query set $\hat{\tau}_i$ randomly;
- 6 Extract user embedding h_i ;
- 7 **if** Approximated Tree-based method **then**
- 8 **if** Warm-up stage **then**
- 9 Set LR as α ;
- 10 Store_Node(h_i, α);
- 11 **else**
- 12 $\langle h_k \rangle_{k=1}^K, \langle \alpha(h_k) \rangle_{k=1}^K =$
 Search_Tree(h_i);
- 13 $s := \{s_k\}_{k=1}^K =$
 $\{\langle \phi(h_i), \phi(h_k) \rangle_{\mathcal{H}}^2\}_{k=1}^K$;
- 14 Obtain $\alpha(h_i) =$
 $\sum_k \frac{s_k}{\sum_j s_j + \sigma} \alpha(h_k) + \alpha'(h_i)$;
- 15 Store_Node($h_i, \alpha(h_i)$);
- 16 **else if** Regularized-based method **then**
- 17 Obtain the learning rate α directly by
 $\alpha(h_i)$;
- 18 Update $\theta^i \leftarrow \theta - \alpha(h_i) \nabla_{\theta} \mathcal{L}_{\tau_i}(\cdot; \theta)$ with support set τ_i ;
- 19 **if** Regularized-based method **then**
- 20 update θ by Eq. (4) with query set
 $\hat{\tau} = \langle \hat{\tau}_i \rangle_{i=1}^N$;
- 21 **else if** Approximated Tree-based method **then**
- 22 update θ by Eqs. (5) & (6) with query set
 $\hat{\tau} = \langle \hat{\tau}_i \rangle_{i=1}^N$. (Eqs. (4), (5) and (6) are shown in Appendix C).

Output: a well-trained meta-strategy.

users are similar: i.e., $\|L_{\tau_i}(\theta_i) - L_{\tau_j}(\theta_j)\|$. Similarly, the multi-user loss is $\sum_{u_i, u_j \in \mathcal{U}; i > j} \|L_{\tau_i}(\theta_i) - L_{\tau_j}(\theta_j)\|$. It is easy to understand because forgetting a task means its loss value for that task is higher than the loss value of other tasks. If the model can obtain qualified and similar loss values for each user, we can guarantee that the model has memorized all existing users' information.

However, it is hard to implement it into practice since computing multi-user loss needs to calculate all the user-pair losses together. To address this challenge, we turn to optimize its upper bound:

$$\sum_{u_i, u_j \in \mathcal{U}; i > j} \|L_{\tau_i}(\theta_i) - L_{\tau_j}(\theta_j)\| \leq \sum_i (|\mathcal{U}| - 1) \left\| \frac{\partial L_{\tau_i}}{\partial \theta} \right\|^2 \|\alpha(h_i)\| + C,$$

where C is a constant. *The proof of the upper bound can be found in Appendix A.2.* Notice that the sum of $\left\| \frac{\partial L_{\tau_i}}{\partial \theta} \right\|^2 \|\alpha(h_i)\|$ can be divided into each task (user) loss as a regularizer to enable the network to remember each

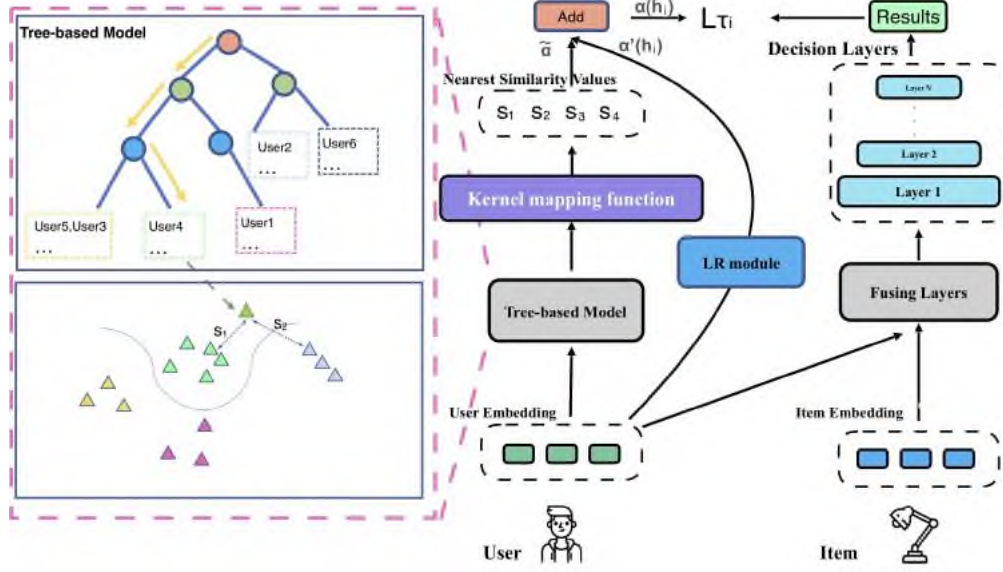


Figure 2: The network structure of the tree-based PAML. The right part is the network for inner update while the left part is the tree-based module. Specifically, *Fusing Layer* indicates the concatenation of user and item embeddings, *Add* is an adding function: $\alpha'(h_i) + \tilde{\alpha}$. *Layer*1, $\dots$ *N* means fully connected layers and $S_1, \dots, S_4$ are similarity values.

Algorithm 2: Store and Search approaches

```

1 Def Store_Node ( $h_i, \alpha$ ) :
2   if Memory is full then
3     Remove the embeddings and LR that are least
       recently used;
4   Store the embeddings, LR, and build the standard
       kd-tree by embedding (Friedman, Bentley, and
       Finkel 1976);
5   return the kd-tree;
6 Def Search_Tree ( $h_i$ ) :
7   Search the nearest neighbours (Yianilos 1993) and
       their corresponding LRs;
8   return the nearest neighbours and their
       corresponding LRs ;

```

user information: i.e., for user i , the regularizer L_i^r is set as: $L_i^r = \|\frac{\partial L_{\tau_i}}{\partial \theta}\|^2 \|\alpha(h_i)\|$. Intuitively, when $L_i^r = 0$, it indicates that the meta strategy has already obtained a good score for user i without training. Therefore, this term reinforces the model to remember the users it meets. Moreover, since $|\mathcal{U}| - 1$ is a constant, we can replace it with a positive real number γ to balance $L_{\tau_i}^r$ and L_{τ_i} . We name the PAML with this REGularizer as *REG-PAML*. Since we do not need any extra space to store user embedding, the space complexity is *constant*. The total loss function for REG-PAML is $\mathcal{L} = \sum_i L_{\tau_i} + \gamma L_i^r$. The relationship to implicit MAML (Rajeswaran et al. 2019) can be found in Appendix C. The pseudo-code can also be found in Alg. 1.

5 Empirical Studies

This section validates our methods in various environments and tasks, including the rating prediction (in the MovieLens-1M dataset and the BookCrossing dataset) as well as the CTR prediction (the real-world production dataset). We also analyze the performance of our methods for both the minor and major users and conduct ablation studies.

5.1 Experimental Setup

Datasets. We use both open-source datasets as well as real-world production dataset to evaluate the performance of our methods in user imbalanced dataset, including MovieLens-1M (Harper and Konstan 2015), BookCrossing (Ziegler et al. 2005) as well as production dataset (collected from Taobao e-commerce platforms, which is somewhat similar to (Zhao et al. 2018; Guo et al. 2019)). The data pre-processing scheme is: 1) We rank the users with the number of their log data and choose the 80% users least log data as the cold-start users (the value 80% is based on the Pareto Principle). 2) We randomly split the users by 7 : 1 : 2 for training, validation, and testing. 3) We tick out the users when they either have blank or wrong features (including the ages are less than 10 or larger than 100 as well as the location features with garble) or the items they viewed are less than a threshold (two items for rating prediction and two clicked items for CTR prediction). 4) We separate the support and query sets for each user with a ratio of 80%:20% randomly. Details about the features we use, and the statistics can be found in Appendix D.1 & D.3.

Major and minor users separating. To testify whether our methods provide better prediction results for both minor users and major users, we need to define a criteria to split the major and minor users. However, as we mentioned above, finding the minor users are hard because there is not an exact

method to precisely cluster the users by their features. Here we define a simple but not completely precise rule to *approximately* split the minor and major users: since distribution of the feature embeddings is the key to classify whether a user is minor or major user, we split the minor and major users by the following criteria: if the user with more than two of features which are in the top 30% largest number of feature values set, we regard it as the major user. Otherwise, it is the minor user. The major and minor users are 65% and 35% in MovieLens, 72% and 28% in BookCrossing, and 71% and 29% in the Taobao dataset.

Baselines. To validate our methods, we compare them with several State-Of-The-Art (SOTA) methods, including MeLU (Lee et al. 2019), Meta-SGD (Li et al. 2017), and transfer learning (Tan et al. 2018). 1) MeLU (Lee et al. 2019). MeLU is one of the state-of-the-art gradient-based meta-learning methods in RS. Since it is very similar to other user-based meta learning (Finn, Abbeel, and Levine 2017; Bharadhwaj 2019), we use it as the standard gradient-based meta-learning method. 2) Meta-SGD (Li et al. 2017). Meta-SGD is an adaptive learning rate meta-learning method. Comparing with standard MAML using a fixed (inner) learning rate, it uses different learning rates for each parameter. But those adaptive parameters do not rely on the user embedding. We use it as a baseline to testify the performance of our embedding-based adaptive LR methods. 3) Transfer-learning (Tan et al. 2018). We choose the standard transfer-learning approach which has been widely adopted to address the user imbalance issue as the baseline. Specifically, we train the model with all users in training data and fine-tune the trained model in each user support set in the test data.

Network structures. For the AT-PAML, as shown in Fig. 2, it contains five modules, the embedding modules (including the embedding layers for each feature respectively, we use an FC layer to concatenate them together), the tree-based module, the learning-rate module, the decision module as well as the fusing module. The embedding modules are the standard embedding layers (Pennington, Socher, and Manning 2014). The tree-based module is what we discussed in Sec. 4.2. The LR module is two FC layers with ReLU as activation function. The decision module (except for the last layer) is based on the FC layer with ReLU as activation function. For the CTR prediction, the last layer is a two-dimension softmax layer. For the rating prediction, it is a real value without any activation function. The fusing layer is to concatenate all user and item embeddings together. More details can be found in Appendix D.2. For the REG-PAML and PAML, except that they do not have the tree module, other modules are similar.

Our methods. We validate our proposed methods, including the *Approximated Tree-based PAML* (AT-PAML) as well as *Regularizer PAML* (REG-PAML). For ablations, we implement vanilla PAML and REG-PAML with different γ values (we use $\gamma = 10^{-3}$ as default gamma value). We use Pytorch⁵ as the deep learning library and we use the ADAM as the optimizer (Kingma and Ba 2014). All the experiments are done on a single GeForce GTX 1080 Ti. The production dataset

is obtained by the company’s recommendation pipeline. The package for the two-tailed student’s t-test is from⁶. The running time for REG-PAML and PAML are about 2 hours, and for AT-PAML is about 4 hours on a single GTX 1080 GPU for MovieLens. For memory space, the space size for AT-PAML is 2.12GB, while that of the REG-PAML/ PAML is about 669MB in MovieLens.

Evaluation metrics. Following (Lee et al. 2019; Ren et al. 2019), we leverage different evaluation metrics to testify different aspects of our methods, including Mean Squared Error (MSE) and average Normalized Discounted Cumulative Gain (NDCG) for rating prediction results; as well as Area Under the Curve (AUC) and weighted Negative Entropy Loss (NEL) for CTR prediction results. To evaluate the performance difference between the minor and major users, we use the two-tailed student’s t-test for statistical hypothesis test. The null hypothesis is that 2 independent samples (minor users and major users) have identical average (expected) values. If the p-value is high, indicating that the expected values of the minor users and major users are not different. On the contrary, when the p-value is low, it shows that the expected values of the minor users and major users are different. Therefore, when a method focuses on both minor and major users, it does **not** have a low p-value. More detailed definitions can be found in Appendix D.6.

5.2 Results and Discussions

Rating Prediction. We first analyse the rating prediction results. As shown in Table 2, our methods (AT-PAML & REG-PAML) outperform other SOTA methods in different metrics, especially for REG-PAML, which has reached 3 out of 6 best results. For the AT-PAML, it has two best results and two second-best results, implying that this method is also well-performing. From the Figs. 5 and 6 in Appendix D.7, we can find that REG-PAML has fewer outliers, indicating the robustness of REG-PAML. Moreover, in Fig. 5, Meta-SGD has some outliers with large values, which might be the reason that Meta-SGD does not perform well in the BookCrossing dataset. Moreover, both the AT-PAML and REG-PAML have lower third quartile comparing with SOTAs, showing that our methods can achieve good-enough performance for most of the cold-start users.

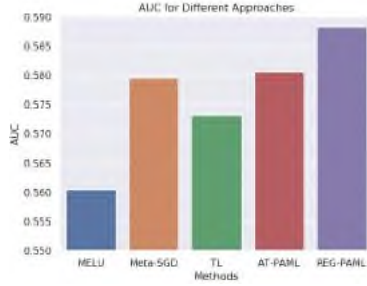
CTR prediction. We then testify our method in the CTR prediction task. As shown in Figs. 3(a) and 3(c), Our methods achieve the best (0.5882) and the second-best (0.5806) results in the AUC metric comparing with other SOTA methods (the best result of SOTA methods is 0.5796). One interesting phenomenon is that for AUC, MeLU gets the worst result among all the methods, indicating that the vanilla meta learning does not work well in this dataset. For the NEL, our methods also get the top two results (-0.5382, -0.5426) among all other methods (the best result of SOTA methods is -0.5080). Moreover, MeLU does not perform well enough comparing with ours, indicating the importance of building a learnable mapping between user embedding to LR. Fig. 3(b) reveals some statistical information on the NEL. We can find that

⁵<https://pytorch.org/>

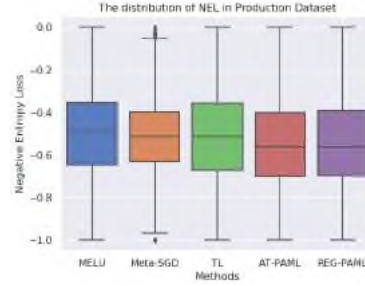
⁶https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

	MovieLens-1M			BookCrossing		
	MSE ↓	Avg nDCG@3 ↑	Avg nDCG@5 ↑	MSE ↓	Avg nDCG@3 ↑	Avg nDCG@5 ↑
MeLU (Lee et al. 2019)	1.451±0.022	0.793±0.002	0.800±0.001	4.019±0.101	0.864±0.002	0.914±0.002
Meta-SGD (Li et al. 2017)	1.340±0.054	0.773±0.007	0.794±0.007	5.197±0.089	0.867±0.001	0.921±0.002
Transfer-Learning (Tan et al. 2018)	<i>1.308±0.016</i>	0.778±0.001	0.796±0.002	4.522±0.376	0.862±0.010	0.919±0.005
AT-PAML (ours)	1.322±0.007	0.788±0.003	0.806*±0.006	3.991±0.114	0.852±0.001	0.930*±0.002
REG-PAML (ours)	1.210*±0.029	0.779±0.002	0.803±0.001	3.928*±0.176	0.868±0.001	0.917±0.001

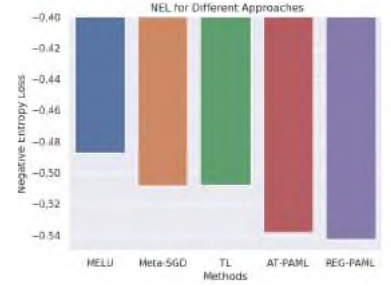
Table 2: Comparison of different methods on the MovieLens and the BookCrossing datasets. The best results are highlighted in **bold** and the second-best results are in *italic*. Avg means average. The mean and standard deviation are reported by 3 independent trials. * denotes statistically significant improvement over the best baseline method (measured by t-test with p-value < 0.05).



(a) The AUC for different methods in test dataset. TL is the abbreviation for transfer learning. Higher is better.

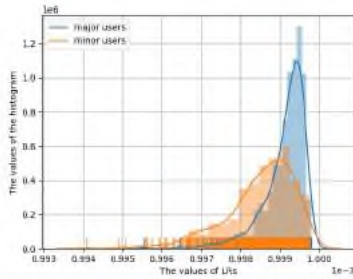


(b) The box plot of the user NEL in test dataset.

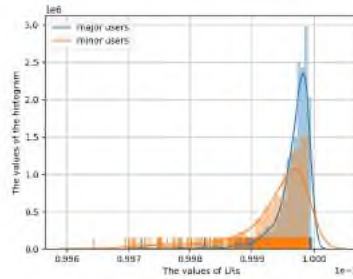


(c) The NEL for different methods in the production dataset. Lower is better.

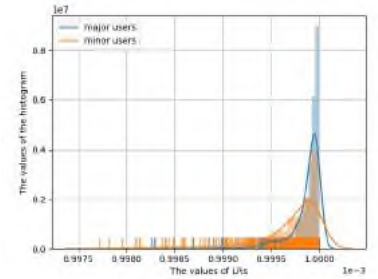
Figure 3: The performance and statistics of different methods in the production dataset. The results are reported by 3 independent trials. The NEL for REG-PAML has statistically significant improvement over the best baseline method (measured by t-test with p-value < 0.05).



(a) The distribution of minor and major user LR for REG-PAML ($\gamma = 10^{-3}$).



(b) The distribution of minor and major user LR for REG-PAML ($\gamma = 10^{-5}$).



(c) The distribution of minor and major user LR for PAML.

Figure 4: The distribution of LR for REG-PAML ($\gamma = 10^{-3}$), REG-PAML ($\gamma = 10^{-5}$), and PAML.

the mean and the first quartile of the NEL distribution for Meta-SGD are larger than ours. This result reveals that our methods can make better prediction results than other meta learning methods for both the minor and major users.

Minor users. We conduct experiments to show whether our methods help to improve the performance of minor users. As shown in Tab. 4, AT-PAML outperforms all the SOTAs in both minor users and major users significantly. Also, from the two-tailed student's t-test analysis, we find that the AT-PAML has the highest p-value, indicating that with a high probabi-

ity the expected values for minor users and major users are similar. We also notice that SOTA meta-learning methods (MeLU and Meta-SGD) have relatively low p-values, revealing that the expected values for minor and major users are different with high probability. The results reveal that our methods are not only well-performing in the whole datasets but also focus on both the minor and major users than the SOTA meta-learning methods. Moreover, we also visualize the LR for each user, as shown in Fig. 4. Combining with Tab. 3, the closer the LR of minor and major users are, the

	MovieLens		
	MSE	Avg nDCG@3	Avg nDCG@5
REG-PAML ($\gamma = 10^{-5}$)	1.509 $\pm$ 0.029	0.773 $\pm$ 0.004	0.799 $\pm$ 0.003
REG-PAML ($\gamma = 10^{-4}$)	1.227 $\pm$ 0.026	0.779 $\pm$ 0.003	0.803 $\pm$ 0.002
REG-PAML ($\gamma = 10^{-3}$)	1.210 $\pm$ 0.013	0.791* $\pm$ 0.002	0.814* $\pm$ 0.004
REG-PAML ($\gamma = 10^{-2}$)	1.191* $\pm$ 0.021	0.777 $\pm$ 0.002	0.803 $\pm$ 0.004
PAML	1.520 $\pm$ 0.034	0.787 $\pm$ 0.001	0.809 $\pm$ 0.004
AT-PAML	1.322 $\pm$ 0.007	0.788 $\pm$ 0.003	0.806 $\pm$ 0.006

Table 3: Ablations of different methods on MovieLens-1M. The best results are highlighted in **bold** and the second-best results are in *italic*. Avg means average. The results are reported by 3 independent trials. * denotes statistically significant improvement over PAML (measured by t-test with p-value<0.05).

	MovieLens-1M		
	MSE (MAU)	MSE (MU)	P-value
MELU	1.413 $\pm$ 0.025	1.506 $\pm$ 0.021	0.071
Meta-SGD	1.300 $\pm$ 0.059	1.403 $\pm$ 0.045	0.024
Transfer-Learning	1.288 $\pm$ 0.016	1.340 $\pm$ 0.013	0.227
REG-PAML (ours)	1.194 $\pm$ 0.019	1.236 $\pm$ 0.011	0.312
AT-PAML (ours)	1.284 $\pm$ 0.005	1.382 $\pm$ 0.011	0.018

Table 4: Results of different MSEs for MAJOR Users (MAU) and Minor Users (MU) on MovieLens. The best results are highlighted in **bold** and the second-best results are in *italic*. P-value is drawn by two-tailed student’s t-test between the minor user MSEs and major user MSEs over the same methods.

higher the MSEs, showing different LRs do play an important role in different user preference predictions. One interesting phenomenon is that during the experiments, we notice a *failure case* shown in Fig. 8 in Appendix D.7, which reveals when the method is not capable to distinguish the minor users and the major users, it cannot perform well. These results also empirically show the significance of different LRs for different users. Also, the t-SNE (Maaten and Hinton 2008) visualization of user embeddings (Fig. 11 in Appendix D.7) show that our methods can distinguish the minor and major users. Tab. 5 reveals that γ affects the performance of both major and minor users.

Ablations. We do ablation studies to evaluate whether our proposed modules work. As shown in Tab. 3, PAML with tree-based structures and regularizer ($\gamma = 10^{-3}$) perform better than the vanilla one under the MSE and the Avg nDCG@3 metrics. Also, AT-PAML and REG-PAML achieve lower values in the MSE metric for both minor and major users (Tab. 5 in Appendix D.7). An interesting phenomenon is that with the decrease of parameter γ , the MSE values become higher while the p-values become lower, meaning that the model performs worse and focuses more on major users. This result reveals that the regularizer does play an important role to achieve better performance and drive the model to focus on both minor and major users.

	MovieLens-1M		
	MSE (MAU)	MSE (MU)	P-value
REG-PAML ($\gamma = 10^{-5}$)	1.464 $\pm$ 0.032	1.601 $\pm$ 0.021	0.008
REG-PAML ($\gamma = 10^{-4}$)	1.210 $\pm$ 0.026	1.255 $\pm$ 0.024	0.226
REG-PAML ($\gamma = 10^{-3}$)	1.196 $\pm$ 0.019	1.183 $\pm$ 0.024	0.738
REG-PAML ($\gamma = 10^{-2}$)	1.194 $\pm$ 0.014	1.236 $\pm$ 0.011	0.312
PAML	1.471 $\pm$ 0.036	1.569 $\pm$ 0.031	0.038
AT-PAML	1.284 $\pm$ 0.005	1.382 $\pm$ 0.011	0.018

Table 5: Ablations of different methods for MAJOR Users (MAU) and Minor Users (MU) on MovieLens. The best results are highlighted in **bold** and the second-best results are in *italic*. Avg means average. P-value is drawn by two-tailed student’s t-test between the minor user MSEs and major user MSEs over the same methods.

6 Conclusion

In this paper, we propose a novel personalized adaptive meta-learning method to address the user overfitting problem in cold-start user preference prediction challenge with three key contributions: 1) We are the first to introduce a personalized adaptive learning rate based meta learning approach to improve the performance of MAML by focusing on both the major and minor users. 2) We build an approximated tree-based method to store different users’ features for fast calculation and a similarity-based method to find similar users as a reference to obtain accurate personalized learning rates for different users. 3) To reduce the memory usage, we propose a memory agnostic regularizer to reduce the space complexity and maintain the memorizing ability by learning. Experiments on MovieLens-1M, BookCrossing, and real-world production dataset reveal that our method outperforms the state-of-the-art methods dramatically for both the minor users and the major users.

7 Acknowledgements

This work was supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Joint Research Institute (JRI), Nanyang Technological University, Singapore.

The authors would like to thank Suming Yu, Zhenyu Shi, Rundong Wang, Xinrun Wang, Feifei Lin, Aye Phyu Phyu Aung, Hanzao Chen, Ziwen Jiang, Yi Cao, Yufei Feng for their helps. The authors would also thank the team of MELU for their codes.

References

- Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 139–154.
- Bharadhwaj, H. 2019. Meta-Learning for User Cold-Start Recommendation. In *IJCNN*, 1–8.
- Chen, F.; Luo, M.; Dong, Z.; Li, Z.; and He, X. 2018. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*.

- Dong, M.; Yuan, F.; Yao, L.; Xu, X.; and Zhu, L. 2020. MAMO: memory-augmented meta-optimization for cold-start recommendation. In *KDD*, 688–697.
- Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 1126–1135.
- Finn, C.; Yu, T.; Zhang, T.; Abbeel, P.; and Levine, S. 2017. One-Shot Visual Imitation Learning via Meta-Learning. In *CORL*, 357–368.
- Flennerhag, S.; Rusu, A. A.; Pascanu, R.; Yin, H.; and Hadsell, R. 2020. Meta-Learning with Warped Gradient Descent. In *ICLR*.
- Friedman, J. H.; Bentley, J. L.; and Finkel, R. A. 1976. An algorithm for finding best matches in logarithmic time. *TOMS* 3: 209–226.
- Gui, L.-Y.; Wang, Y.-X.; Ramanan, D.; and Moura, J. M. 2018. Few-shot human motion prediction via meta-learning. In *ECCV*, 432–450.
- Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *AAAI*, 1725–1731.
- Guo, Q.; Li, Z.; An, B.; Hui, P.; Huang, J.; Zhang, L.; and Zhao, M. 2019. Securing the deep fraud detector in large-scale e-commerce platform via adversarial machine learning approach. In *WWW*, 616–626.
- Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *TIIS* 5(4): 1–19.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS* 114(13): 3521–3526.
- Lee, H.; Im, J.; Jang, S.; Cho, H.; and Chung, S. 2019. MeLU: Meta-Learned user preference estimator for cold-Start recommendation. In *KDD*, 1073–1082.
- Lee, Y.; and Choi, S. 2018. Gradient-Based Meta-Learning with Learned Layerwise Metric and Subspace. In *ICML*, 2933–2942.
- Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-SGD: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.
- Luo, M.; Chen, F.; Cheng, P.; Dong, Z.; He, X.; Feng, J.; and Li, Z. 2020. MetaSelector: meta-learning for recommendation with user-level adaptive model selection. In *WWW*, 2507–2513.
- Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9: 2579–2605.
- Madotto, A.; Lin, Z.; Wu, C.-S.; and Fung, P. 2019. Personalizing dialogue agents via meta-learning. In *ACL*, 5454–5459.
- Muja, M.; and Lowe, D. 2013. FLANN-Fast Library for Approximate Nearest Neighbors User Manual. *University of British Columbia*.
- Muja, M.; and Lowe, D. G. 2014. Scalable nearest neighbor algorithms for high dimensional data. *PAMI* 36(11): 2227–2240.
- Pan, F.; Li, S.; Ao, X.; Tang, P.; and He, Q. 2019. Warm Up Cold-start Advertisements: Improving CTR predictions via learning to learn ID embeddings. In *SIGIR*, 695–704.
- Park, E.; and Oliva, J. B. 2019. Meta-curvature. In *NeurIPS*, 3309–3319.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.
- Qu, Y.; Cai, H.; Ren, K.; Zhang, W.; Yu, Y.; Wen, Y.; and Wang, J. 2016. Product-based neural networks for user response prediction. In *ICDM*, 1149–1154.
- Rajeswaran, A.; Finn, C.; Kakade, S. M.; and Levine, S. 2019. Meta-learning with implicit gradients. In *NeurIPS*, 113–124.
- Ravi, S.; and Larochelle, H. 2016. Optimization as a model for few-shot learning. In *ICLR*.
- Ren, K.; Qin, J.; Fang, Y.; Zhang, W.; Zheng, L.; Bian, W.; Zhou, G.; Xu, J.; Yu, Y.; Zhu, X.; et al. 2019. Lifelong Sequential Modeling with Personalized Memorization for User Response Prediction. In *SIGIR*, 565–574.
- Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-Learning with Latent Embedding Optimization. In *ICLR*.
- Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; and Liu, C. 2018. A survey on deep transfer learning. In *ICANN*, 270–279.
- Vanschoren, J. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*.
- Vartak, M.; Thiagarajan, A.; Miranda, C.; Bratman, J.; and Larochelle, H. 2017. A meta-learning perspective on cold-start recommendations for items. In *NeurIPS*, 6904–6914.
- Xu, Z.; van Hasselt, H. P.; and Silver, D. 2018. Meta-gradient reinforcement learning. In *NeurIPS*, 2396–2407.
- Yianilos, P. N. 1993. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, volume 93, 311–321.
- Zhao, M.; Li, Z.; An, B.; Lu, H.; Yang, Y.; and Chu, C. 2018. Impression Allocation for Combating Fraud in E-commerce Via Deep Reinforcement Learning with Action Norm Penalty. In *IJCAI*, 3940–3946.
- Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *WWW*, 22–32.

Capturing Delayed Feedback in Conversion Rate Prediction via Elapsed-Time Sampling

Jia-Qi Yang^{1*}, Xiang Li^{2†}, Shuguang Han², Tao Zhuang²
De-Chuan Zhan^{1†}, Xiaoyi Zeng², Bin Tong²

¹ State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

² Alibaba Group, Hangzhou, China

¹{yangjq,zhandc}@lamda.nju.edu.cn

²{leo.lx,shuguang.sh,zhuangtao.zt,yuanhan,tongbin.tb}@alibaba-inc.com

Abstract

Conversion rate (CVR) prediction is one of the most critical tasks for digital display advertising. Commercial systems often require to update models in an online learning manner to catch up with the evolving data distribution. However, conversions usually do not happen immediately after user clicks. This may result in inaccurate labeling, which is called delayed feedback problem. In previous studies, delayed feedback problem is handled either by waiting positive label for a long period of time, or by consuming the negative sample on its arrival and then insert a positive duplicate when conversion happens later. Indeed, there is a trade-off between waiting for more accurate labels and utilizing fresh data, which is not considered in existing works. To strike a balance in this trade-off, we propose Elapsed-Time Sampling Delayed Feedback Model (ES-DFM), which models the relationship between the *observed conversion* distribution and the *true conversion* distribution. Then we optimize the expectation of *true conversion* distribution via importance sampling under the elapsed-time sampling distribution. We further estimate the importance weight for each instance, which is used as the weight of loss function in CVR prediction. To demonstrate the effectiveness of ES-DFM, we conduct extensive experiments on a public data and a private industrial dataset. Experimental results confirm that our method consistently outperforms the previous state-of-the-art results.

Introduction

Digital display advertising has become the main business model for many online services, in which advertisers pay for placing ads on those platforms. Among the available payment options, paying per conversion (CPA) is usually the dominated mechanism as conversions can directly bring profits. In the CPA model, advertisers pay only if users performed certain predefined conversion actions with the advertisement. To effectively display ads, machine learning models have been widely adopted to forecast the conversion rate (CVR), which is widely investigated in both academia and

industry (Lee et al. 2012; Chapelle, Manavoglu, and Rosales 2014; Ma et al. 2018).

In order to capture the dynamic change of user needs, commercial systems often update learned models with up-to-date data within a short time, i.e., in an online training manner (Jugovac, Jannach, and Karimi 2018; Guo et al. 2019; Ktena et al. 2019). This further complicates the CVR prediction since conversions usually do not happen immediately after user clicks. The *Delayed Feedback* issue introduces a dilemma for streaming CVR prediction — on the one hand, we need to wait for a sufficient long time so that the observation information can approximately reflect the true conversion; on the other hand, we also tend to update the learned models without much delay for model-freshness.

Chapelle (2014) was among the early studies to address the delayed feedback problem. The proposed Delayed Feedback Model (DFM) optimizes CVR as a joint probability over the predicted CVR and the delayed time distribution. This joint probability is estimated in the observed time interval, which may be biased from the true conversion distribution. The biased CVR is probably more inaccurate due to the delayed feedback problem in online learning settings.

To achieve unbiased CVR estimation in delayed feedback problem, recent studies have explored the way of optimizing the expectation of true conversion distribution via importance sampling (Bishop 2007). Ktena et al. (2019) proposed the Fake Negative Weighted (FNW) approach, in which each arriving instance is firstly labeled as negative, and then corrected upon its conversion at a later time. Each fake negative instance may have a side-effect for the learned model until it is amended. This side-effect is amplified if the data distribution frequently changes. For example, in the beginning of a promotion event, user clicks may increase dramatically while most conversions come after a certain time. Such overwhelming fake negatives may harm the predictive model. Instead of blindly labeling each incoming example as a negative instance, Yasui et al. (2020) proposed a Feedback Shift Importance Weighting (FSIW) algorithm, in which the model waits for the real conversion in a certain time interval. However, FSIW does not allow the data correction even a conversion event took place afterward. We argue that positive examples are important for delayed feedback prediction as the positive examples are always more scarce than the negative examples. Moreover, FSIW may lack model-

*Jia-Qi Yang performed this work as an intern at Alibaba.

†De-Chuan Zhan and Xiang Li are the corresponding authors. This work was supported by NSFC (61773198, 6163000043, 61921006, 61751306).

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

freshness due to a long-time wait. Therefore, either updating the model in nearly real time (Ktena et al. 2019), or waiting a sufficiently long time for conversion (Yasui et al. 2020) may not be able to address the delayed feedback problem in the streaming CVR prediction.

For unbiased CVR estimation in the online settings, we propose to wait for a time interval which is modeled as a distribution. The readily available conversion information allows the model to trade-off the label correctness and online model-freshness, which are achieved in FSIW and FNW, respectively. Due to the introduction of the observed time distribution, delayed positive samples can be better handled than FNW via important sampling techniques. Especially in scenarios of promotion events, FNW may fail to have unbiased estimation due to that the distribution of positive samples in observed time may be dramatically different from the routines. On the other hand, FSIW is able to guarantee the label correctness but lacks of model-freshness. Furthermore, the model is not able to correct instance label even the delayed positive instance comes later. The introduction of time distribution in our proposal helps the model correct the label of an instance by degrading the weight of negative instance and upgrading the weight of positive instance.

In this work, we propose Elapsed-Time Sampling Delayed Feedback Model (ES-DFM), which models the relationship between the *observed conversion* distribution and the *true conversion* distribution. We optimize the expectation of *true conversion* distribution via importance sampling under the elapsed-time sampling distribution. We further estimate the importance weight for each instance, which is used as the weight of loss function in CVR prediction. To demonstrate the effectiveness of ES-DFM, we conduct extensive experiments on two widely-used datasets — a public ads conversion logs provided by Criteo, and a private data set provided by Taobao. Experimental results confirm that our method consistently outperforms the previously state-of-the-art results in most of the cases. Our main contribution can be summarized as following:

- To the best of our knowledge, we are the first to study the trade-off between waiting more accurate labels and exploiting fresher training data in the context of streaming CVR prediction.
- By explicitly modeling the elapsed time as a probability distribution, we achieve the unbiased estimation of true conversion distribution. Particularly, our model is shown to be robust even if the data distribution is different from the routines.
- We provide a set of rigorous experimental setups for streaming training and evaluation, which better aligns with industrial systems, and can be easily applied to real-world applications.

Related Work

Delayed Feedback Models

The mostly cited work that were addressing the delayed feedback problem came from Chapelle (2014), in which

the authors stated that such a problem is related to the survival time analysis (John D. Kalbfleisch 2002). The Delayed Feedback Model (DFM) assumed an exponential delay for conversion time distribution, and, based on that, proposed two models: one focusing on CVR prediction and the other on conversion delay prediction. Built on top of the DFM model, Yoshikawa and Imai (2018) further proposed a non-parametric delayed feedback model (NoDeF), in which the delay time was modeled without any parametric assumptions. One significant drawback of the above methods was that both of them only attempted to optimize the observed conversion information rather than the actual delayed conversion.

Importance Sampling

Using samples from one distribution to estimate the expectation with respect to another distribution can be achieved by importance sampling method. Ktena et al. (2019) proposed fake negative weighted method (FNW) to optimize the ground truth CVR prediction objective based on importance sampling. Under the assumption that all samples are initially labeled as negative, the delayed feedback problem can be resolved by FNW in expectation. However, in the streaming setting, every fake negative will affect the model negatively until it’s corresponding positive duplicate arrives. This negative effect can be amplified drastically under distribution change. Yasui et al. (2020) proposed a feedback shift importance weighting method (FSIW), where the importance weight is estimated with the aid of waiting time information. However, FSIW does not allow duplicated samples, thus cannot correct the mislabeled samples using the subsequent positive labels by inserting duplicates.

Delayed Bandits

The delayed feedback in the bandit algorithm has been researched (Joulani, György, and Szepesvári 2013; Mandel et al. 2015; Cesa-Bianchi, Gentile, and Mansour 2019). The aforementioned approaches often provide efficient and provably optimal algorithms for delayed feedback scenarios. However, such methods naturally wait for having received enough feedback before actually learning something, which may be quite unsuitable in a non-stationary environment. Vernade, György, and Mann (2020) defined a new stochastic bandit model and addressed the real-world modelling issues when dealing with non-stationary environments and delayed feedback. However, the objective in the bandit problem is to sequentially make decisions in order to minimize the cumulative regret, our goal is to predict the CVR in order to derive a bid price in ad auction.

Background

In this work, we focus on the CVR prediction task, which takes the user features x_u and the item features x_i as inputs, all the features are denoted by x , and aims to learn the probability that the user converts on the item. $y \in \{0, 1\}$ indicates the conversion label, where $y = 1$ means the conversion, otherwise $y = 0$. Ideally, the CVR model is trained

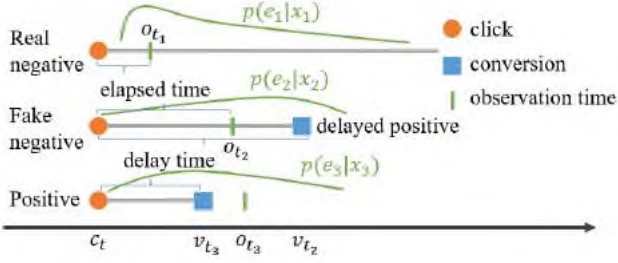


Figure 1: An illustration of different kind of time information for the delayed feedback tasks.

on top of the training data (x, y) drawn from the data distribution of ground-truth $p(x, y)$, thereby optimizing the ideal loss shown as follows:

$$\mathcal{L}_{ideal} = \mathbb{E}_{(x,y) \sim p(x,y)} \ell(y, f_\theta(x)) \quad (1)$$

where f is the CVR model function, and θ is the parameter. ℓ is the classification loss, and the widely-used cross entropy is adopted. However, due to the delayed feedback problem, the observed distribution of the training data $q(x, y)$ often deviates from the distribution of the ground-truth $p(x, y)$. Therefore, the ideal loss $\mathcal{L}_{ideal}$ is unavailable.

To formulate such a delayed feedback setting more precisely, we introduce three time points and corresponding time intervals in Figure 1. These three time points are the click time c_t when a user clicks an item, the conversion time v_t when a conversion action happens, and the observation time o_t when we extract the training samples. Then the time interval between c_t and o_t is denoted as the *elapsed time* e , and the time interval between c_t and the v_t is denoted as the *delayed feedback time* h . Therefore, the samples are labeled as $y = 1$ (positive) in the training data, when $e > h$, otherwise some positive samples are mislabeled as $y = 0$ (fake negative) when $e < h$.

Proposed Method

In order to realize flexible control on the waiting time, we assume the elapsed time is drawn from an elapsed time distribution $p(e|x)$. Then we developed a probabilistic model that combines the elapsed time distribution $p(e|x)$, the delay time distribution $p(h|x, y = 1)$ and the conversion rate $p(y = 1|x)$ into a unified framework. To achieve an unbiased estimation of the actual CVR prediction objective, we propose an importance weighting method corresponding to our elapsed sampling method. Then we provide a practical estimation of the importance weights, and give an analysis of the bias introduced by this estimation, which can guide us on designing an appropriate elapsed time distribution $p(e|x)$.

Elapsed-Time Sampling Delayed Feedback Model

To strike the balance between obtaining accurate feedback information and keeping model freshness, a reasonable waiting time (elapsed time) should be integrated into the modeling process. Moreover, the elapsed time e should be a distribution depend on x , i.e., $p(e|x)$. For example, users need

more time to consider when buying high-priced products, thus a long waiting time is required. When a click x_i arrives, an elapsed time e_i is drawn from $p(e|x_i)$. Then we wait the sample x_i for the time interval of e_i before assigning a label, and subsequently train on the data. By introducing the time distribution, we propose our Elapsed-Time Sampling Delayed Feedback Model (ES-DFM), which modeling the relationship between the *observed conversion* distribution $q(y|x)$ and the *true conversion* distribution $p(y|x)$, according to:

$$q(y = 0|x) = p(y = 0|x) + p(y = 1|x)p(h > e|x, y = 1) \quad (2)$$

$$q(y = 1|x) = p(y = 1|x)p(h \leq e|x, y = 1) \quad (3)$$

where

$$p(h > e|x, y = 1) = \int_0^\infty \left[p(e|x) \int_e^\infty p(h|x, y = 1) dh \right] de \quad (4)$$

$$p(h \leq e|x, y = 1) = \int_0^\infty \left[p(e|x) \int_0^e p(h|x, y = 1) dh \right] de \quad (5)$$

At the time of model training, some conversions that will occur eventually have not yet been observed, and previous methods like DFM and FSIW have ignored these conversions. We argue this is important for a delayed feedback task as the positive examples are way more scarce than the negative examples, and the positives may define the direction for model optimization. Therefore, in this work, as soon as the user engages with the ad, the data will be sent (duplicated if there is already a fake negative) to the model with a positive label. Then, $q(y|x)$ should be re-normalized as following:

$$q(y = 0) = \frac{p(y = 0) + p(y = 1)p(h > e|y = 1)}{1 + p(y = 1)p(h > e|y = 1)} \quad (6)$$

$$q(y = 1) = \frac{p(y = 1)}{1 + p(y = 1)p(h > e|y = 1)} \quad (7)$$

where the condition on x is omitted for conciseness, i.e., $q(y = 0) = q(y = 0|x)$, $p(y = 0) = p(y = 0|x)$, etc. Since we have inserted delayed positives, the total number of samples will increase by $p(y = 1)p(h > e|y = 1)$, so we should normalize by dividing $1 + p(y = 1)p(h > e|y = 1)$. The number of negatives will not change, so dividing Eq. (2) by this normalizing factor yields Eq. (6). The number of positives will increase by $p(y = 1)p(h \leq e|y = 1)$, so the numerator of $q(y = 1)$ is $p(y = 1)p(h \leq e|y = 1) + p(y = 1)p(h > e|y = 1)$. Using the fact that $p(h \leq e|y = 1) + p(h > e|y = 1) = 1$ yields Eq. (7).

Importance Weight of ES-DFM

To obtain unbiased CVR estimation in delayed feedback problem, we optimize the expectation of $p(y|x)$ via impor-

tance sampling (Bishop 2007). First, we provide the theoretical background of importance sampling, as follows:

$$\mathcal{L}_{ideal} = \mathbb{E}_{(x,y) \sim p(x,y)} \ell(y, f_\theta(x)) \quad (8)$$

$$= \int p(x) dx \int p(y|x) \ell(y, f_\theta(x)) dy \quad (9)$$

$$= \int p(x) dx \int q(y|x) \frac{p(y|x)}{q(y|x)} \ell(y, f_\theta(x)) dy \quad (10)$$

$$\approx \mathbb{E}_{(x,y) \sim q(x,y)} \frac{p(y|x)}{q(y|x)} \ell(y, f_\theta(x)) \quad (11)$$

$$= \mathcal{L}_{iw} \quad (12)$$

where f is the CVR model function, and θ is the parameter. ℓ is the classification loss, and the widely-used cross entropy is adopted. Notice that we assume $p(x) \approx q(x)$ to obtain (11) from (10), which is reasonable since the proportion of delayed positive is small, and this approximation is also used by Ktena et al. (2019). According to (11), we can optimize the ideal objective with a appropriate weight $w(x, y) = \frac{p(y|x)}{q(y|x)}$. Second, we further provide the importance weight under the proposed elapsed-sampling distribution. From Equation (6) and (7), we can obtain:

$$\frac{p(y=0|x)}{q(y=0|x)} = [1 + p_{dp}(x)] p_{rn}(x) \quad (13)$$

$$\frac{p(y=1|x)}{q(y=1|x)} = 1 + p_{dp}(x) \quad (14)$$

where

$$p_{dp} = p(y=1)p(h > e) \quad (15)$$

$$p_{rn} = \frac{p(y=0)}{p(y=0) + p(y=1)p(h > e)} \quad (16)$$

$p_{dp}(x)$ is the *delayed positive* probability, denote the probability that a sample is a duplicated positive; $p_{rn}(x)$ is the *real negative* probability, denote the probability that an observed negative is ground truth negative and will not convert.

Finally, considering Eq. (8) to Eq. (14), the importance weighed CVR loss function is:

$$\begin{aligned} \mathcal{L}_{iw}^n = & - \sum_{(x_i, y_i) \in \tilde{\mathcal{D}}} y_i [1 + p_{dp}(x_i)] \log(f_\theta(x_i)) \\ & + (1 - y_i) [1 + p_{dp}(x_i)] p_{rn}(x_i) \log(1 - f_\theta(x_i)) \end{aligned} \quad (17)$$

where $\tilde{\mathcal{D}}$ is the training data drawn from elapsed-time sampling distribution $q(x, y)$.

Estimation of Importance Weight (IW)

The challenge of resolving the delayed feedback problem through importance sampling is that we need to estimate the importance weights $w(x, y)$.

In this work, we decompose $w(x, y)$ into two parts: $p_{dp}(x)$ and $p_{rn}(x)$, according to Eq. (13) and Eq. (14). More precisely, we estimate these two probability with two binary classifiers. Namely, we train a classifier f_{dp} to predict the probability of being a *delayed positive* (Eq. (15)), and train a

classifier f_{rn} to predict the probability of being a *real negative* (Eq. (16)). The model architecture of $f_{dp}(x)$ and $f_{rn}(x)$ is the same as the CVR prediction model. To construct the training dataset, for each sample (x_i, y_i) , an elapsed time e is drawn from $p(e|x_i)$. Then, for the f_{dp} model, the delayed positives are labeled as 1, the others are labeled as 0; For the f_{rn} model, the observed positives are excluded, then the negatives are labeled as 1 and the delayed positives are labeled as 0. In practice, all these needed labels are available in a delayed data stream (for example, delayed by 30 days to ensure label correctness), and the data selection can be achieved by a mask on the loss function, thus we train the f_{rn} and f_{dp} models jointly with a shared network in streaming training.

Importance sampling methods usually suffer from high variance due to the division of two probabilities. Our method is less likely to introduce a large variance. The key is on how the importance weight $\frac{p(y|x)}{q(y|x)}$ is calculated. The high variance of importance sampling is mainly introduced by the large value of $\frac{p(y|x)}{q(y|x)}$ when $q(y|x) \ll p(y|x)$ at some (x, y) . However, we estimate the importance weight using the delayed positive rate p_{dp} and the real negative rate p_{rn} (in Eq. (17)), and these two values are probabilities and bounded within $[0, 1]$.

Bias Analysis of Estimated IW

The importance weighted loss function Eq. (17) is unbiased using ideal p_{dp} and p_{rn} . However, a bias may be introduced due to the estimated importance weights f_{dp} and f_{rn} . Through optimizing the loss function Eq. (17), and using the estimated f_{dp}, f_{rn} instead of ideal p_{dp}, p_{rn} , the predicted probability $f(x)$ converges to:

$$f(x) = \frac{p(y=1|x)}{p(y=1|x) + p_{neg}(x)f_{rn}(x)} \quad (18)$$

$$p_{neg}(x) = p(y=0|x) + p(y=1|x)p(h > e|x) \quad (19)$$

Proof sketch. Take partial derivative of Eq. (17) with respect to f , and set the derivative to zero. A detailed proof is given in the supplementary material[‡]. $\square$

From Eq. (18) and Eq. (19), we can draw the following observations, which can guide us to design appropriate elapsed-time sampling distribution $p(e|x)$:

- First, if f_{rn} is perfectly correct, we have $f_{rn} = p_{rn}$, then $f(x) = p(y=1)$, thus leading to no bias. However, in practice, f_{rn} is learned through historical data, bias always exists.
- Second, the bias is also related to $p(y=1|x)$ according to Eq. (18) and Eq. (19). Therefore, if the absolute value of conversion rate is large, the bias introduced by f_{rn} may be larger.
- Last, the sampling distribution $p(e|x)$ can be used to control the bias. If e is long, $p(h > e)$ will be smaller. Thus $p(y=0) + p(y=1)p(h > e)$ will be close to $p(y=0|x)$. f_{rn} will be more close to 1 since there are few fake negatives. Thus $p_{neg}(x)f_{rn}(x)$ is more close to $p(y=0|x)$.

Therefore, we can control the waiting time(elapsed time) distribution $p(e|x)$ to reduce bias, which is the core to realize the aforementioned trade-off and is the missing part of existing methods.

Experiments*

To evaluate the proposed model, we conduct a set of experiments to answer the following research questions:

RQ1 How does ES-DFM perform, compared to the state-of-the-art models for the streaming CVR prediction task?

RQ2 How do different choices of elapsed time affect the performance? What is the best elapsed time of the dataset?

RQ3 How do mislabeled samples affect importance weighting methods in streaming training?

RQ4 How does ES-DFM perform in online recommender systems?

Datasets

Public Dataset We use the Criteo dataset[†] used in Chapelle (2014) to evaluate the proposed method. This dataset is formed by Criteo live traffic data in a period of 60 days, which corresponds to conversions after a click has occurred. Each sample is described by a set of hashed categorical features and a few continuous features. It also includes the timestamps of the clicks and those of the conversions, if any. The statistics of Criteo dataset is shown in Table 1.

Taobao Dataset We collect 98×10^8 samples in a period of 14 days from the daily click and conversion logs in Taobao system, which consist of the user and item features with the labels (i.e., click or conversion) for the CVR task. The feature set of an item contains several categorical features and continuous features. The statistics of Taobao Dataset is shown in Table 1.

Dataset Preprocessing We divide both public and anonymous dataset into two parts evenly. We use the first part for model pre-training and achieve a well initialized CVR prediction model. We use the second part for streaming data simulation to evaluate compared methods.

Evaluation Metrics

We adopt three widely used metrics for the CVR prediction task (Ni et al. 2018; Zhou et al. 2019; Ktena et al. 2019; Yasui et al. 2020), which show a model’s performance from different perspectives. The first metric is area under ROC curve (AUC) which assesses the pairwise ranking performance of the classification results between the conversion and non-conversion samples. The second metric is area under the precision-recall curve (PR-AUC) which is more sensitive

than AUC in skewed data like CVR prediction task (Yasui et al. 2020). The last metric is negative log likelihood (NLL), which is sensitive to the absolute value of the CVR prediction (Chapelle 2014). In a CPA model, the predicted probabilities are important since they are directly used to compute the value of an impression.

Streaming Experimental Protocol

We have designed an experimental evaluation method for the streaming CVR prediction, which can fully verify the performance of different methods in the online learning settings. In this work, we divide the streaming dataset into multiple datasets according to the click timestamp, each of which contains one hour data. Following the online training manner of industrial systems, the model is trained on the t -th hour data and tested on the $t + 1$ -th hour data, then trained on the $t + 1$ -th hour data and tested on the $t + 2$ -th hour data, and so on and so forth. Note that, the training data is re-constructed with fake negatives, while evaluation data is the original data. Therefore, we report the weighted metrics of the evaluation dataset of different hours to verify the overall performance of different methods on streaming data.

Compared Methods

We compare our method with the state-of-the-art methods:

- **Pre-trained:** A CVR model without any finetuning.
- **Vanilla Finetune Model:** A model finetuned on top of the pre-trained model using the streaming data, which is the baseline method.
- **Delayed Feedback Model (DFM)**(Chapelle 2014): A model finetuned on top of the pre-trained model using delayed feedback loss.
- **Fake Negative Weighted (FNW)** (Ktena et al. 2019): A model finetuned on top of the pre-trained model using the fake negative weighted loss.
- **Fake Negative calibration(FNC)** (Ktena et al. 2019): A model finetuned on top of the pre-trained model using the fake negative calibration loss.
- **Feedback Shift Importance Weighting (FSIW)** (Yasui et al. 2020): The pre-trained model will be fine-tuned using the FSIW loss and pre-trained auxiliary model.
- **Elapsed-Time Sampling Delayed Feedback Model (ES-DFM):** Our proposed method which try to keep the model fresh while introducing low bias.

We also reported performance of an **Oracle*** model: A model finetuned using the ground truth label instead of observed label, assuming the conversion label is known at click time. This is the upper bound of possible improvements, where the delayed feedback problem does not exist. The asterisk* denotes that it’s not a baseline method.

Parameter Settings

For fair comparison, all hyper-parameters are tuned carefully for all compared methods. The feature engineering of the numerical features and the categorical features is the

*The code for reproducing our results on public dataset is available at https://github.com/ThyrixYang/es_dfm

[†]The original url provided by Criteo at <https://labs.criteo.com/2013/12/conversion-logs-dataset/> is broken, thus we provide a copy at https://github.com/ThyrixYang/es_dfm. The users should conform to the terms of use from Criteo to use this dataset.

Dataset	# Users	# Items	# categorical feature	# continuous feature	# conversions	# Samples	# average CVR	# log period
Criteo Dataset	-	5443	9	8	3619801	15898883	0.2269	60 days
Taobao Dataset	0.25 billion	0.8 billion	12	10	0.32 billion	9.8 billion	0.03273	14 days

Table 1: Statistics of Criteo and Taobao Dataset.

Method	Criteo Dataset						Taobao Dataset					
	AUC	PR-AUC	NLL	<i>R-AUC</i>	<i>R-PR-AUC</i>	<i>R-NLL</i>	AUC	PR-AUC	NLL	<i>R-AUC</i>	<i>R-PR-AUC</i>	<i>R-NLL</i>
Pre-trained	0.8307	0.6251	<u>0.4009</u>	-0.9212	-0.2058	<u>0.2139</u>	0.8731	0.6525	0.1156	-1.0374	-0.5217	-0.2419
Vanilla	<u>0.8376</u>	0.6288	0.4047	<u>0.0000</u>	0.0000	0.0000	0.8842	0.6645	0.1141	0.0000	0.0000	0.0000
Oracle*	0.8450	0.6469	0.3868	1.0000	1.0000	1.0000	0.8949	0.6875	0.1079	1.0000	1.0000	1.0000
DFM	0.8132	0.5784	1.2599	-3.2581	-2.7833	-47.645	0.8702	0.6471	0.1271	-1.3084	-0.7565	-2.0968
FSIW	0.8290	0.6189	0.4099	-1.1432	-0.5479	-0.2891	0.8735	0.6591	0.1149	-0.9971	-0.2348	-0.1290
FNC	0.8373	0.6267	0.4382	-0.0393	-0.1147	-1.8646	<u>0.8851</u>	0.6669	0.1142	<u>0.0841</u>	0.1043	-0.0161
FNW	0.8373	<u>0.6313</u>	0.4033	-0.0308	<u>0.1400</u>	0.0773	0.8845	<u>0.6672</u>	<u>0.1137</u>	0.0280	<u>0.1174</u>	<u>0.0645</u>
ES-DFM	0.8402*	0.6393*	0.3924*	0.3560	0.5799	0.6831	0.8895*	0.6762*	0.1112*	0.4953	0.5087	0.4677

Table 2: Performance comparisons of proposed model with baseline models on AUC, PR-AUC and NLL metrics. The bold value marks the best one in one column, while the underlined value corresponds to the best one among all baselines. Here, * indicates statistical significance improvement compared to the best baseline measured by t-test at p -value of 0.05. *R-AUC*, *R-PR-AUC* and *R-NLL* are relative metrics indicating the improvements within the delayed feedback gap.

same as the settings in the work (Chapelle 2014). Since we mainly discuss the delayed feedback issue in this paper, the model architecture is a simple MLP model with the hidden units fixed for all models with [256, 256, 128]. The activation functions are Leaky ReLU and every hidden layer is followed by a BatchNorm layer (Ioffe and Szegedy 2015) to accelerate learning. Adam (Kingma and Ba 2015) is used as the optimizer with the learning rate of 10^{-3} . L2 regularization strength is 10^{-6} . We describe the detailed setting of compared methods in the Supplementary Material [‡] due to the page limit.

Choice of $p(e|x)$

The sampling elapsed time distribution $p(e|x)$ can be designed based on expert knowledge and the aforementioned bias analysis. For example, users need more time to consider when buying high-priced products, thus a long waiting time is required. However, the public dataset is anonymized, where information like price-level is unavailable. To verify the effectiveness of introducing $p(e|x)$ in the streaming settings, we perform a simplified implementation of $p(e|x)$. More precisely, we set $p(e = c|x) = 1$ where c is a constant, which means $p(e|x)$ degenerates to a Dirac distribution. This brings us two following advantages. First, we can strike the balance between obtaining accurate feedback information and keeping model freshness with a single parameter c . Second, we conducted experiments with different c in the public dataset, and the experimental results show that choosing the best c can significantly improve performance. The c is also tuned on the private dataset and we report the best result which is achieved using $c = 1$.

[‡]Some experiment details and discussion are provided at https://github.com/ThyrixYang/es.dfm/blob/master/aaai21_sup.pdf

Standard Streaming Experiments: RQ1

From Table 2, we can see that our proposed method improves the performance significantly against all the baselines and achieves the state-of-the-art performance. Moreover, some further observations can be made. First, the performance of DFM and FSIW is worse than the vanilla baseline on both the public and Taobao Dataset. This is because DFM is difficult to converge, thus failing to achieve a good performance in streaming CVR prediction, and FSIW does not allow the data correction once a conversion took place afterwards, which is important for delayed feedback. Second, in most cases, FNC and FNW perform better than the vanilla baseline. Specially, FNW outperforms the baseline in both PR-AUC and NLL, which is consistent with the results reported in Ktena et al. (2019). Third, existing methods show little superior performance in terms of AUC, while our method outperform the best baseline by 0.26% and 0.44% AUC scores on the Criteo and Taobao Dataset, respectively. As reported in Zhou et al. (2018), DIN improves AUC scores by 1.13% and the improvement of online CTR is 10.0%, which means a small improvement in offline AUC is likely to lead to a significant increase in online CTR. In our practice, for cutting-edge CVR prediction models, even 0.1% of AUC improvement is substantial and achieves significant online promotion.

We further analyze the maximum benefit that can be achieved by resolving the delayed feedback problem. The maximum benefit is defined as the performance gap between the oracle model and baseline. Therefore, the goal of any method that tackling delayed feedback problem is to narrow this gap. We report three relative metrics within the performance gap, i.e, Relative-AUC(R-AUC), Relative-PR-AUC(R-PR-AUC) and Relative-NLL(R-NLL). As shown in Table 2, our method can narrow the delayed feedback

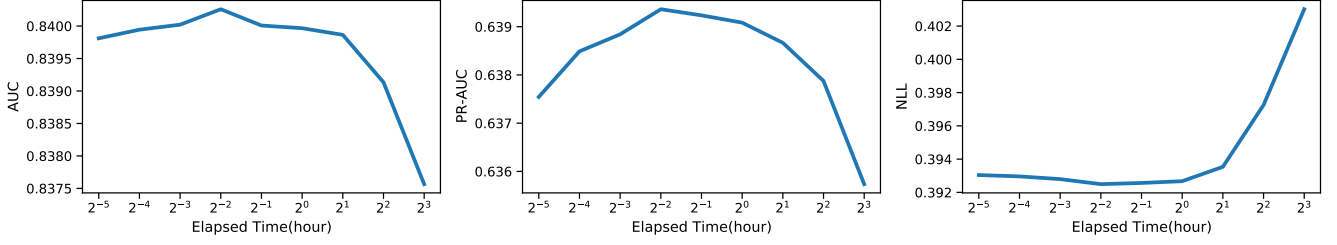


Figure 2: Experiments on the effect of elapsed time on performance. We control the elapsed time by a parameter c , which is the value on the x axis.

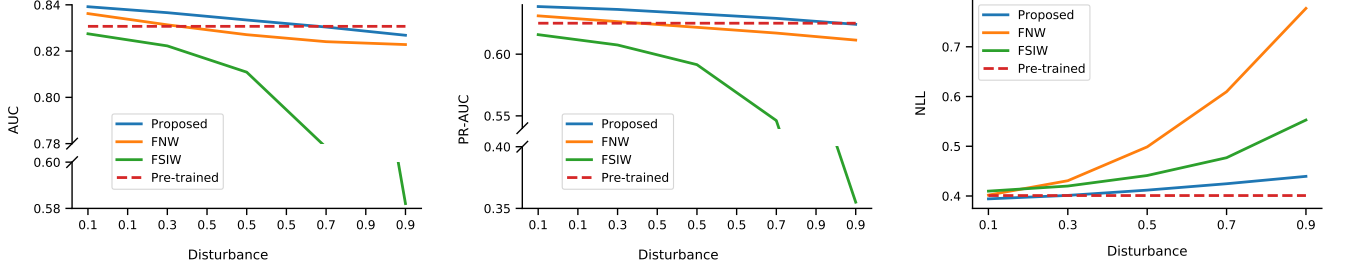


Figure 3: The experiment on resistance to disturbance. x axis is the disturbance strength which controls the portion of positive samples to be flipped.

gap significantly comparing to other methods, and the absolute improvement is larger when the delayed feedback gap is larger.

Influence of Elapsed Time: RQ2

To verify the performance of different choices of elapsed time, we have conducted experiments using different values of c on the Criteo dataset. As shown in Figure 2, the best c on the Criteo dataset is around 15 minutes, where about 35% conversions can be observed. Moreover, larger or smaller c will reduce the performance. The performance decreases slowly on smaller c , which indicates that the bias introduced by the importance weighting model is small. The performance decreases faster on larger c , which indicates that the data freshness matters more when c increase, and a c larger than 1 hour will significantly harm the performance.

Experiment on Robustness: RQ3

In delayed feedback setting, the same sample may be labeled as negative or positive. It is closely related to learning with noisy labels (Natarajan et al. 2013), where some of the labels are randomly flipped. We hypothesis that a method dealing with delayed feedback problem should not only correct incorrect labels, but also reduce the negative effect of the incorrect labels before they can be corrected or the correction fails (for example, if the weighting model deviate a lot, the bias will be large and correction will fail). Thus we conducted a robustness experiment. We randomly select d portion of all the positive samples in streaming dataset, then swap it's label (and click time and pay time) with a random selected negative one. Note that we do not disturb on the

pre-training dataset, so the initial CVR model and the pre-trained importance weighting models are not disturbed. We conducted experiments with different disturbance strength d , the results are shown in Figure 3. We can see that our method is more resistant to disturbance comparing to FNW and FSIW, and the performance gap is larger when disturbance increases (especially on NLL). We give an intuitive analysis about the weak robustness of FNW and FSIW in the Supplementary Material[‡].

Online Evaluation: RQ4

We conducted an A/B test in our online evaluation framework. We observed a steady performance improvement, AUC increases by 0.3% within a 7 days window compared with the best baseline, CVR increases by 0.7%, GMV (Gross Merchandise Volume) increases by 1.8%, where GMV is computed by the transaction number of items multiplied by the price of each item. The online A/B testing results align with our offline streaming evaluation and show the effectiveness of ES-DFM in industrial systems.

Conclusion

The trade-off between the label accuracy and model freshness in streaming training setting has never been considered, which is an active decision of the method rather than a passive feature in offline setting. In this paper, we propose elapsed-time distribution to balance the label accuracy and model freshness to address the delayed feedback problem in the streaming CVR prediction. We optimize the expectation of true conversion distribution via importance sampling under the elapsed-time sampling distribution. More-

over, we propose a rigorous streaming training and testing experimental protocol, which aligns with real industrial applications better. Finally, extensive experiments show the superiority of our approach.

References

- Bishop, C. M. 2007. *Pattern recognition and machine learning*. Springer.
- Cesa-Bianchi, N.; Gentile, C.; and Mansour, Y. 2019. Delay and cooperation in nonstochastic bandits. *The Journal of Machine Learning Research* 20(1): 613–650.
- Chapelle, O. 2014. Modeling delayed feedback in display advertising. In *KDD*, 1097–1105. ACM.
- Chapelle, O.; Manavoglu, E.; and Rosales, R. 2014. Simple and scalable response prediction for display advertising. *TIST* 5(4): 1–34.
- Guo, L.; Yin, H.; Wang, Q.; Chen, T.; Zhou, A.; and Quoc Viet Hung, N. 2019. Streaming session-based recommendation. In *KDD*, 1569–1577.
- Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 448–456. PMLR.
- John D. Kalbfleisch, R. L. P. 2002. *The Statistical Analysis of Failure Time Data*.
- Joulani, P.; György, A.; and Szepesvári, C. 2013. Online Learning under Delayed Feedback. In *ICML*, 1453–1461. JMLR.org.
- Jugovac, M.; Jannach, D.; and Karimi, M. 2018. Streamingrec: a framework for benchmarking stream-based news recommenders. In *Recsys*, 269–273.
- Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- Ktena, S. I.; Tejani, A.; Theis, L.; Myana, P. K.; Dilipkumar, D.; Huszár, F.; Yoo, S.; and Shi, W. 2019. Addressing delayed feedback for continuous training with neural networks in CTR prediction. In *RecSys*, 187–195. ACM.
- Lee, K.-c.; Orten, B.; Dasdan, A.; and Li, W. 2012. Estimating conversion rate in display advertising from past performance data. In *KDD*, 768–776.
- Ma, X.; Zhao, L.; Huang, G.; Wang, Z.; Hu, Z.; Zhu, X.; and Gai, K. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *SIGIR*, 1137–1140.
- Mandel, T.; Liu, Y.-E.; Brunskill, E.; and Popovic, Z. 2015. The Queue Method: Handling Delay, Heuristics, Prior Data, and Evaluation in Bandits. In *AAAI*, 2849–2856.
- Natarajan, N.; Dhillon, I. S.; Ravikumar, P.; and Tewari, A. 2013. Learning with Noisy Labels. In *NIPS*.
- Ni, Y.; Ou, D.; Liu, S.; Li, X.; Ou, W.; Zeng, A.; and Si, L. 2018. Perceive Your Users in Depth: Learning Universal User Representations from Multiple E-commerce Tasks. In *KDD*, 596–605.
- Vernade, C.; György, A.; and Mann, T. A. 2020. Non-Stationary Delayed Bandits with Intermediate Observations. In *ICML*. PMLR.
- Yasui, S.; Morishita, G.; Fujita, K.; and Shibata, M. 2020. A Feedback Shift Correction in Predicting Conversion Rates under Delayed Feedback. In *WWW '20: The Web Conference 2020*, 2740–2746. ACM / IW3C2.
- Yoshikawa, Y.; and Imai, Y. 2018. A Nonparametric Delayed Feedback Model for Conversion Rate Prediction .
- Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; and Gai, K. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI*, volume 33, 5941–5948.
- Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep interest network for click-through rate prediction. In *KDD*, 1059–1068. ACM.

Keypoint-Guided Fashion Detection

Shenhan Qian,¹ Dongze Lian,¹ Binqiang Zhao,² Jiawei Li,² Tong Liu,⁴ Bohui Zhu,² Hai Li,³ Shenghua Gao¹

¹ ShanghaiTech University ² Alibaba Group ³ Ant Group Group ⁴ Taobao
{qianshh, liandz, gaoshh}@shanghaitech.edu.cn

{binqiang.zhao, mingong.ljw, bohui.zbh}@alibaba-inc.com, yingmu@taobao.com, tianshu.lh@antgroup.com

Abstract

To recognize a garment in an image is challenging because the classification boundaries between garments are vague in the pattern but clear in the functionality. Since keypoints are defined explicitly from the perspective of functionality, it is reasonable to consider improving clothing detection with cues of keypoints. We design a keypoint-guided clothing detection method called KGDet. This method features aggregating features around keypoints as well as generating bounding boxes from predicted keypoints. To realize efficient feature aggregation from dense keypoints, we propose two solutions base on keypoint-encoding and keypoint-selection, respectively. Our method exhibits superior performance and efficiency on both the DeepFashion2 and FLD dataset.

Abstract is too short, to be improved.

Introduction

Recently, fashion image understanding has drawn a lot of attention (Liu et al. 2016a; Ge et al. 2019) due to its wide range of applications, especially in online shopping scenarios. A fundamental problem of fashion image analysis is clothing detection¹, which is to locate and recognize garments efficiently and accurately. It enables many other tasks, e.g., clothing attribute analysis, outfit matching, virtual try-on, etc.

There have been many works for general object detection (Ren et al. 2015; Redmon et al. 2016; Cai and Vasconcelos 2018). But directly adapting them to clothing detection might not be optimal because the difference between clothing categories can be subtle, *i.e.*, the classification boundaries between garments are vague in the pattern but clear in the functionality. As the example in Figure 1 shows, a vest and a sling have almost the same appearance except near the shoulders. That is to say, even for a human being, the key to correctly classifying a garment is to pay attention to local features (*e.g.*, keypoints around the shoulders). Therefore, we stress that the keypoint is a vital cue to determine the

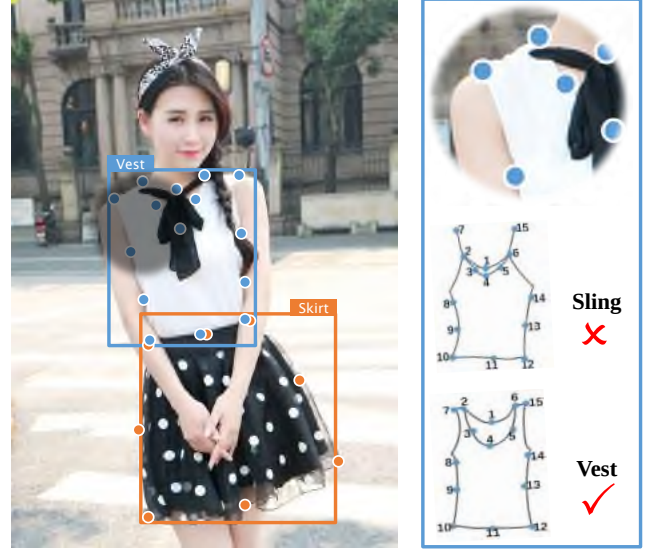


Figure 1: Local features of keypoints are useful for recognizing the category of a garment. In this figure, the keypoints on the collar shows that the garment is a vest rather than a sling.

category of a garment, which is ignored by existing methods (Liu et al. 2016a; Ge et al. 2019; Sidnev et al. 2019).

Then a question is raised: how can we incorporate keypoints cues into clothing detection? At first glance, we can bind clothing detection and keypoint estimation together in a parallel manner, as is done in Mask R-CNN (He et al. 2017), which extracts a dense region of features for each instance, then perform object detection and keypoint regression in separate branches. Such practice does combine the two tasks in a single model, but the connection between them is not fully utilized.

In this paper, we propose a keypoint-centered clothing detection method. Specifically, we design a unified framework that firstly estimates a garment’s keypoints, then explicitly aggregates local features around these points for subsequent refinement of clothing detection. In pursuit of high efficiency, we generate bounding boxes from the predicted keypoints rather than separately regressing them in another branch, which is different from most previous meth-

¹In (Cheng et al. 2020), fashion detection refers to a generalized concept, including landmark detection, fashion parsing, and item retrieval. While, in this paper, we do not strictly distinguish between ‘clothing detection’ and ‘fashion detection’.

ods (Ge et al. 2019; Sidnev et al. 2019) that are in a parallel-connection paradigm.

For our method, a key component is the design of our feature aggregation module. An ideal one should satisfy two requirements: i) as the definitions of keypoints vary across categories of clothes, and all the keypoints of all categories are estimated before classification, the module needs to select a subset of keypoints which are valid for a garment; ii) given the selected keypoints, aggregation of features around them is required. For the first requirement, we propose two solutions: i) multi-column keypoint-encoding based feature aggregation module that leverages a network to automatically do the information extraction; ii) reliable-keypoints-selection based feature aggregation module, which selects reliable keypoints according to their regressed confidence scores. As to the second requirement, we handle it with the deformable convolution (Dai et al. 2017), which extracts and fuses features from different positions guided by a learnable deformable kernel. Here, we use the offset of selected keypoints as the deformable kernel.

Under ideal circumstances, the more accurate the keypoints are estimated, the better the clothing detection performance is. Based on this hypothesis, we further propose a cascade strategy to refine the predicted keypoints repeatedly. Then the above feature aggregation modules serve as a bridge between two cascade stages. Experiments in ablation study show the effectiveness of both modules. Accordingly, a novel network, named KGDet, is proposed to simultaneously detect clothes and their keypoints efficiently in a bottom-up manner.

Our main contributions are as follows:

- we empirically show that keypoints help with locating and recognizing garments, and further propose a KGDet model which incorporates keypoints cues into clothing detection;
- we provide two modules to sparsely aggregate features around keypoints in pursuit of both performance and efficiency: i) multi-column keypoint-encoding based feature aggregation module; ii) reliable-keypoint-selection based feature aggregation module;
- our experiments validate the effectiveness of our method as well as the positive correlation between clothing detection and keypoint estimation.

Related Work

Fashion Image Understanding

Fashion image understanding has such a huge market value that many recent work has focused on it (Liu et al. 2016a; Ge et al. 2019). In the survey paper (Cheng et al. 2020), fashion is mainly categorized into the following four research topics: fashion analysis (Chen, Gallagher, and Girod 2012; Chen et al. 2015; Han et al. 2017; Hidayati et al. 2017; Caba Heilbron et al. 2019), fashion detection ² (Liu et al.

2016b; Yan et al. 2017; Ge et al. 2019), fashion synthesis (Yoo et al. 2016; Dong et al. 2019) and fashion recommendation (Song et al. 2019; Wang, Wu, and Zhong 2019; Vasileva et al. 2018). In this work, we mainly focus on fashion understanding and analysis. The early clothing analysis depends on hand-crafted features (Liu et al. 2012; Kiapour et al. 2014). *With the help of CNNs and large-scale fashion datasets, people utilize deep learning based models to extract features, and the next work pays attention to enhance the representation ability of deep learning models.* For instance, given the DeepFashion (Liu et al. 2016a) dataset, we can study the clothing recognition, landmark detection, retrieval, *etc.* However, having only one garment in an image, as well as the sparse and shared landmark definition limits its applications in the real world. To eliminate these drawbacks, DeepFashion2 dataset (Ge et al. 2019) is introduced, where each image may contain multiple garments, and fine-grained landmarks are defined for different clothing categories. Following this direction, our study focuses on multi-class and multi-instance clothing detection since it is a more realistic scenario to analyze fashion images, either for customers or businesses.

Clothing Detection and Fashion Landmark Detection

Fashion landmark detection is to estimate the keypoints (*e.g.*, collar, hemline and waistline) in clothes (Liu et al. 2016b; Yan et al. 2017; Ge et al. 2019; Li et al. 2019; Lee et al. 2019). In (Liu et al. 2016b), the authors introduce this task and collect a fashion landmark dataset (FLD). Further, a Deep Fashion Alignment (DFA) model is proposed. Next, Yan *et al.* (Yan et al. 2017) propose a hierarchical recurrent transformer network for unconstrained fashion landmark detection. Wang *et al.* (Wang et al. 2018) utilize a knowledge-guided fashion network to solve the fashion landmark detection problem, where a Bidirectional Convolutional Recurrent Neural Networks (BCRNNs) is used for message passing over grammar topologies. Later, (Ge et al. 2019) brings the difficulty of this task to a new level with finer keypoint annotations in the DeepFashion2 dataset. They also propose a Match R-CNN to combine clothing detection, landmark detection, and clothing retrieval in one method. Yu *et al.* (Yu et al. 2019) and Lee *et al.* (Lee et al. 2019) use layout-graph reasoning and a global-local embedding module for fashion landmark detection, respectively. All these methods rely heavily on clothing detection, but few of them consider the correlation between fashion landmarks and clothing detection. In this work, we emphasize that the keypoints annotations will indeed benefit clothing detection.

Our Approach

Our proposed keypoint-guided fashion detection framework is an extension of the anchor-free detector RepPoints (Yang et al. 2019), with keypoints incorporated into the process of clothing detection. In this section, we first present the global architecture of our KGDet network, then exhibit our concrete implementation of classification, keypoint regression, bounding box regression, and the supervision for each

²In this survey paper, fashion detection is a general description including landmark detection, fashion parsing, and item retrieval.

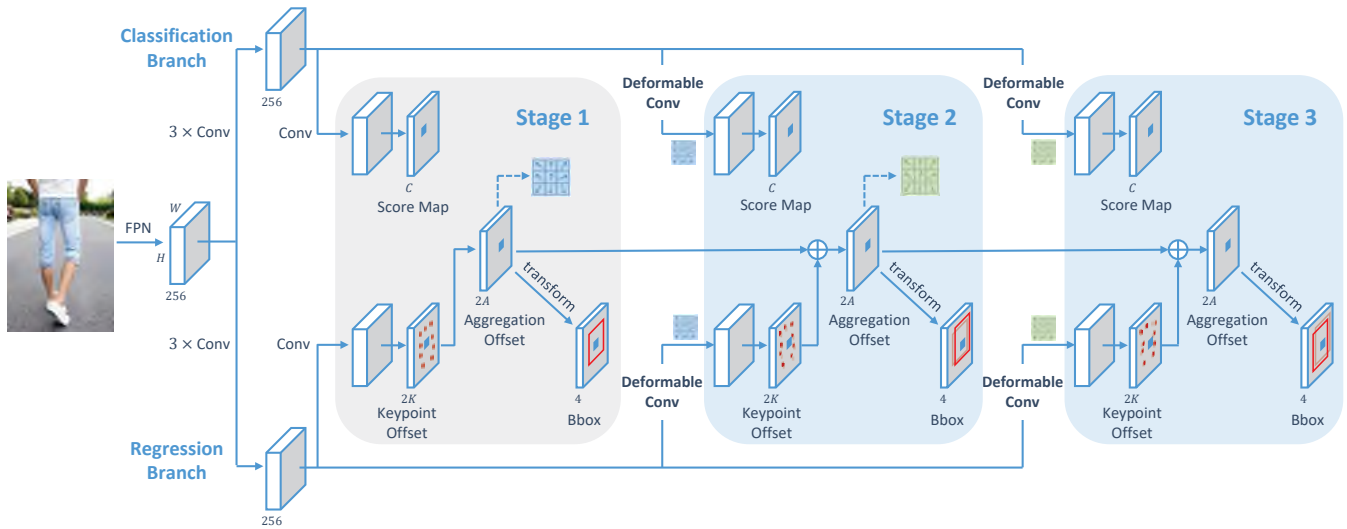


Figure 2: The architecture of the KGDet network, which consists of a initial stage and two refinement stages. Each stage contains two branches: a classification branch and a regression branch. The blue arrow refers to the convolution operation unless otherwise specified. For the labels of the shape of feature maps, C refers to the number of classes, K and A refer to the number of keypoints and aggregation points, respectively.

part. Afterward, we demonstrate our design of the cascading paradigm and the two feature aggregation modules. In the end, we discuss the details of the training and the inference of our model.

KGDet Network

As is shown in Figure 2, our KGDet network consists of an initial stage and two refinement stages. In each stage, there is a classification branch on the top and a regression branch on the bottom. Before feeding feature maps from the FPN into all three stages, we use three 3×3 convolution layers to decouple the feature maps for classification and those for regression. On the classification branch, we regress a score map for each clothing class. On the regression branch, we firstly regress all the keypoints, then generate another set of points (we call them aggregation points) by either keypoint-selection or keypoint-encoding. These points can be viewed as a representative subset or interpolation of all the predicted keypoints. Finally, we use the aggregation points to generate bounding boxes through an outer box transformation. The difference between the initial stage and the refinement stage is that the former one uses the features at the center point of each instance. In contrast, the latter one aggregates features from the aggregation points of its previous stage with the aid of deformable convolution (Dai et al. 2017). Here, the aggregation points serve as the deformation offset.

Classification

Just like other anchor-free detectors (Zhang et al. 2018; Tian et al. 2019; Yang et al. 2019), we represent an object by its center point. We predict a score map $\mathbf{C} \in \mathbb{R}^{H \times W \times C}$, where H and W are the height and width of the feature map, and C is the number of clothes categories. A high activation point

on a certain channel of the score map $\mathbf{C}$ indicates a candidate object center point of a certain class.

On the classification branch, we forward the feature maps for classification through a 3×3 convolution and a 1×1 convolution to generate the score map $\mathbf{C}$. Then, we use a focal loss (Lin et al. 2017) L_{cls} with $\gamma = 2.0$ and $\alpha = 0.25$ to train the classification branch. As to ground-truth assignment, we align the center point of each ground-truth bounding box onto the score map, and set the k -nearest grids of the point as positive samples (k is 9 by default).

Keypoint Regression On the regression branch, we feed the features for regression into a 3×3 convolution and a 1×1 convolution to predict keypoints $\mathbf{K} \in \mathbb{R}^{H \times W \times 2K}$, where H and W are the height and width of the feature map, and K is the total number of keypoints of all clothing categories. The keypoint offset vector at location (i, j) of $\mathbf{K}$ is

$$\Delta \mathbf{k}_{i,j} = [\Delta k_x^1, \Delta k_y^1, \dots, \Delta k_x^K, \Delta k_y^K]^T \in \mathbb{R}^{2K},$$

which encodes the offset values from the location (i, j) to all the K keypoints. Then we have the offset vector of the p -th keypoint

$$\Delta \mathbf{k}_{i,j}^p = [\Delta k_x^p, \Delta k_y^p]^T.$$

Finally, the predicted coordinate of the p -th keypoint at location (i, j) is

$$\mathbf{k}_{i,j}^p = [(\Delta k_x^p + i) \times s, (\Delta k_y^p + j) \times s]^T,$$

where s is the stride of the feature map.

In the training stage, we minimize the smooth l_1 distance between the predicted keypoint coordinate $\mathbf{k}_{i,j}$ and the ground-truth keypoint coordinate $\hat{\mathbf{k}}_{i,j}$ at locations of positive samples of classification. Note that for an instance of clothes, only a subset of keypoints are valid for its class.

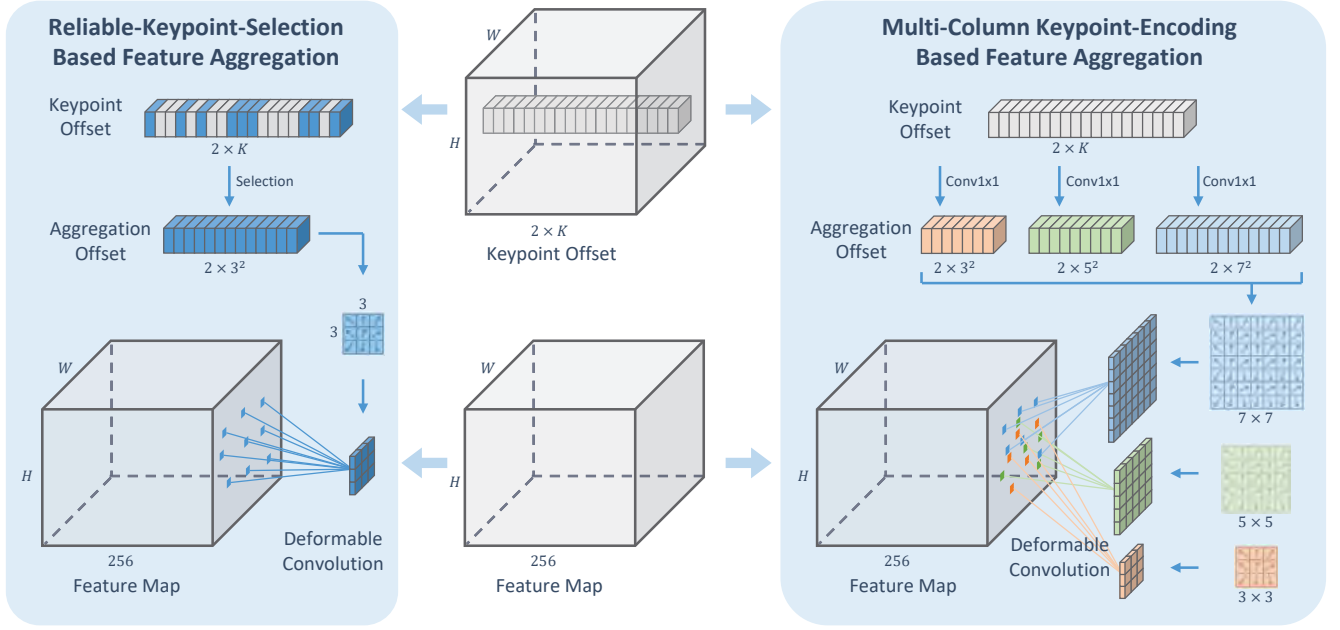


Figure 3: Keypoint-Guided Feature Aggregation.

And even for keypoints of the instance’s class, only part of them occur in an image. Therefore, we only count the loss of these valid keypoints that occur.

Box Regression Head The box regression head is built upon keypoint head because the contour structure of keypoints basically describes the bounding box of objects. Rather than predicting bounding box with an extra branch, we use the output of the keypoint head (*i.e.*, keypoints offset) as the input of the box regression head, where keypoints provides amounts of priors. We choose the keypoints with a high confidence score and extract their outer box, and an extra box deviation predictor is applied to adjust the error of this outer box. We denote the outer box generated from keypoints as $[x_k, y_k, w_k, h_k]$ (center point, width, and height), thus the output of box regression head $\mathcal{B} = [x_b, y_b, w_b, h_b]$ is

$$\begin{cases} x_b = x_k + \Delta x_k \\ y_b = y_k + \Delta y_k \end{cases} \text{ and } \begin{cases} w_b = w_k e^{\Delta w_k} \\ h_b = h_k e^{\Delta h_k} \end{cases}, \quad (1)$$

where $[\Delta x_k, \Delta y_k, \Delta w_k, \Delta h_k]$ are generated by box deviation predictor. Therefore, L_{reg} is defined with smooth l_1 loss between the output $\mathcal{B}$ of box regression head and ground-truth bounding box.

Keypoint-Guided Feature Aggregation

As discussed above, the keypoints are important cues for distinguishing the category of clothes, thus can improve the performance of clothes detection. To effectively extract the local features around keypoints, we propose two kinds of keypoint-guided feature aggregation modules to encode keypoints implicitly and explicitly: i) multi-column keypoint-encoding based feature aggregation module and ii) reliable-

keypoint-selection based feature aggregation module. The two feature aggregation modules are depicted in Figure ??.

Multi-Column Keypoint-Encoding Based Feature Aggregation Module As the keypoints for different clothes categories are different and the clothes’ category label is unknown, so all keypoints for all categories are estimated. To remove the redundant information (*e.g.*, keypoints not related to this category) and select more representative keypoints (*e.g.*, shape and position), we utilize a 1×1 convolution to encode the predicted offsets implicitly. As shown in Figure ?? (left), The input tensor predicted by keypoint head has k keypoints at each spatial position to represent ‘offsets’ $[\Delta x, \Delta y]$. Then it is encoded by a 1×1 convolution to get offsets tensors with different channels (*e.g.*, m_1, m_2, m_3). The encoded offsets will be learned automatically through back-propagation of network. After that, given features from FPN, a deformable convolution is employed to aggregate local features from these offsets. Such a multi-column keypoint-encoding module can extract local features around keypoints effectively for clothes detection.

Reliable-Keypoints-Selection Based Feature Aggregation Module Multi-column keypoint-encoding based feature aggregation can extract more representative keypoints offsets implicitly. Besides this method, we also propose a reliable-keypoints-selection based feature aggregation module. As shown in Figure ?? (right), the input tensor has k keypoints at each spatial position to represent ‘offsets’ and their confidence score $[\Delta x, \Delta y, s_k]$. Rather than encoding the implicitly with a 1×1 convolution, we directly select p keypoints with a high confidence score, and then a deformable convolution is used.

Cascade Strategy

Cascade strategy has been proven to be superior in object detection (Cai and Vasconcelos 2018) and such a strategy is suitable for our simultaneous clothes detection and keypoint estimation. The more accurate the keypoints are estimated, the better the clothes detection performance is. As shown in Figure 2, in the first stage, clothes boxes, fashion landmarks, and object scores are predicted. Using our feature aggregation modules (multi-column keypoint-encoding or reliable-keypoints-selection), the network will learn an enhanced feature map for clothes detection and keypoint estimation. Therefore, such feature aggregation strategies can be regarded as a bridge between two cascade stages. They are also plug-and-play and can be stacked repeatedly. As shown in our experiments, the cascade strategy helps the network learn more accurate keypoints, thus benefit clothes detection.

Network Training

Loss function Our network is composed of keypoint head, classification head and box regression head, thus the whole loss function is the combination of them.

$$L = L_{cls} + \lambda_1 L_{box} + \lambda_2 L_{kpt}, \quad (2)$$

where λ_1 and λ_2 are hyper-parameters to balance these different loss terms. We empirically set $\lambda_1 = 1$ and $\lambda_2 = 0.1$ to adapt the loss magnitude.

Implementations

Training Details We use ResNet-50 as our backbone, followed by the feature aggregation module. Three cascaded modules are used. For the hyper-parameters choice, we train our network with learning rate $1e^{-4}$, weight decay $1e^{-4}$, batch size 8 with 4 GPUs, and the Adam strategy is employed to optimize the whole network. The image with resolution 1333×800 is inputted into our network, which is deployed on NVIDIA P40 GPUs. We only use randomly horizontal flip as data augmentation.

Inference Details Given an image with the same resolution as in the training stage, we forward it through our trained network, where the clothes bounding box, fashion landmark and the object score are outputted. We use the same NMS operation as Mask R-CNN (He et al. 2017) for bounding box post-processing. The fashion landmark at the corresponding position is bundled with the bounding box to do the same post-processing operation, which means if the bounding box is removed, the corresponding fashion landmark at this position will also be moved out.

Experimental Setup

Datasets

We evaluate the proposed method on the DeepFashion2 (Ge et al. 2019) and Fashion Landmark Detection (FLD) (Liu et al. 2016b) dataset.

DeepFashion2 (Ge et al. 2019) is a large-scale fashion dataset containing four benchmarks (clothing detection, landmark estimation, segmentation and commercial-consumer clothes retrieval). There are 391K images for

training, 34K images for validation, and 67K images for test in the original paper (Ge et al. 2019). But, only 192K images for training, 32K images for validation and 63K images for test are released. Thus, we only use the released part of the dataset in our experiments and compare the corresponding performance on this part with other baselines. In this dataset, 13 categories of clothes and a total of 294 class-relevant landmarks are defined.

FLD (Liu et al. 2016b) contains a subset of images from the DeepFashion (Liu et al. 2016a) database with additional annotations for 8 landmarks of 3 main types of clothes. There are 83K images for training, 19K for validation and another 19K for test.

More detailed discussions about datasets can be found in supplementary materials.

Evaluation Metrics

The evaluation metric for clothing detection is AP_{box} , the mean value of the average precision under different thresholds of the Intersection over Union (IOU), which is consistent with DeepFashion2 (Ge et al. 2019). We also report $AP_{box}^{IoU=0.50}$ and $AP_{box}^{IoU=0.75}$.

To evaluate fashion landmark detection, we use AP_{kp} , the mean value of average precision under different thresholds of the Object Keypoint Similarity (OKS). We report $AP_{kp}^{OKS=0.50}$ and $AP_{kp}^{OKS=0.75}$ as well.

Baselines

We consider the following detectors as our baseline models.

Match R-CNN (Ge et al. 2019) is proposed by the authors of the DeepFashion2 (Ge et al. 2019) dataset. Since the dataset is only partially released, we report its performance trained with the publicly available part. The model is built upon *Mask R-CNN* (He et al. 2017) with extra multi-task branches which perform keypoint estimation, and clothing retrieval respectively.

RetinaNet (Lin et al. 2017) and *FCOS* (Tian et al. 2019) are both single-stage detectors. *RetinaNet* (Lin et al. 2017) regresses bounding boxes from prior anchors. While, *FCOS* (Lin et al. 2017) directly regresses bounding boxes, thus regarded as an anchor-free method. We keep the tricks in the original implementation of *FCOS* (Tian et al. 2019) including centerness regression, box normalization, center sampling when performing experiments.

RepPoints (Yang et al. 2019) is also an anchor-free detector. It features modeling an object with learned representative points on it. Given these representative points, it extracts features right from these points with the deformable convolution (Dai et al. 2017).

Evaluation Results:

Improvement, Correlation and Extension

In this section, we first show the influence of keypoint supervision to clothing detection. Then we explore the correlation between the performance of detection and keypoint estimation. Afterwards, we show effects of different designs of modules. Finally we train and test our KGDet model on

Table 1: **Clothing detection** performance of state-of-the-art detectors w/ or w/o the supervision of keypoints.

method	keypoint	backbone	AP	AP ₅₀	AP ₇₅	param.	FLOPs	time (ms)
Match R-CNN (Ge et al. 2019)		ResNet-50-FPN	63.8	78.9	74.5	-	-	-
RetinaNet (Lin et al. 2017)		ResNet-50-FPN	63.2	79.0	73.6	36.35 M	209.67 G	48.5
FCOS (Tian et al. 2019)		ResNet-50-FPN	64.1	80.2	74.7	31.87 M	197.21 G	38.3
RepPoints (Yang et al. 2019)		ResNet-50-FPN	63.3	79.7	74.0	36.61 M	189.79 G	55.2
RepPoints (Yang et al. 2019)	✓	ResNet-50-FPN	64.0	80.1	74.9	38.09 M	208.84 G	67.6
Ours (2-stage)	✓	ResNet-50-FPN	66.3	80.3	74.3	36.92 M	196.49 G	62.5
Ours (3-stage)	✓	ResNet-50-FPN	67.9	80.9	75.0	59.13 M	107.50 G	44.6
RetinaNet (Lin et al. 2017)		ResNet101-FPN	65.7	81.1	75.3	55.35 M	285.74 G	62.1
FCOS (Tian et al. 2019)		ResNet101-FPN	65.9	81.7	75.7	50.81 M	273.28 G	52.6
RepPoints (Yang et al. 2019)		ResNet101-FPN	65.4	80.8	75.2	55.60 M	265.86 G	69.0
RepPoints (Yang et al. 2019)	✓	ResNet-101-FPN	66.0	81.3	75.5	57.08 M	284.91 G	81.3
Ours (2-stage)	✓	ResNet-101-FPN	67.5	81.9	76.2	55.91 M	272.57 G	77.5
Ours (3-stage)	✓	ResNet-101-FPN	68.4	82.3	76.7	78.13 M	183.57 G	58.5

the FLD (Liu et al. 2016b) dataset to show the effectiveness of our design.

Can keypoints help improve clothing detection?

Before exploring whether keypoints can help improve clothing detection, we should first figure out how to integrate keypoints into a detection model. We provide two different solutions, both of which are extensions of the RepPoints (Yang et al. 2019) detector. As is show in Figure ??, one of the solutions is in a parallel paradigm, performing bounding box regression and keypoint regression in separate branches, like in many multi-task learning methods. The other is in a serial paradigm which first regresses the keypoints, then predicts the bounding boxes based on the regressed keypoints. Our KGDet model is a further augmentation of the serial solution with modifications including assignment method, cascading strategy, etc.

In Table 1, we compare the clothing detection performance of different models. Firstly, we can see that the two extensions of RepPoints (Yang et al. 2019) with additional keypoint supervision outperform the vanilla RepPoints (Yang et al. 2019) model. This validates that keypoints do help with clothing detection. Furthermore, our KGDet model outperforms all the baseline models³ with a large margin.

Here, we show some qualitative results and failure cases of fashion detection and landmark estimation in Figure 4. Our cascade keypoint-guided feature aggregation network can accurately detect clothes and its landmarks in various scenarios, such as occlusion, rotation and different sizes.

What is the correlation between keypoints estimation and clothing detection?

Although keypoint estimation is not our goal, it is an important intermediate task that boosts our detector’s performance. It is worth figuring out the correlation between the

performance of the two tasks in our model. For simplicity and efficiency of our model, we directly regress keypoints in the form of offset from an object’s center points. As is stated by previous works (Sun et al. 2018; Nie et al. 2019), it is hard to directly regress a large offset with the convolutional neural network. Therefore, we refine the predictions in a cascaded way. Meanwhile, we strengthen out model by extracting features from more points. In Table 2, we show the performance of clothing detection and keypoint estimation. We observe positive correlation between the performance of both tasks and the number of aggregation points, number of stages, as well as the size of the feature map. We also find that the *flip test* trick for keypoint estimation is beneficial to clothing detection too. Note that if we turn off the keypoint loss, then the detection AP drops for 1.2 point.

Table 2: Performance of **clothing detection** improves along with **keypoint estimation**. The more points we use to aggregate features, the better keypoint estimation and clothing detection performance. Enhancement of keypoints benefits detection. On the contrary, disabling the keypoint supervision results in performance drop of clothing detection.

#agg.pts.	#stages	backbone	loss _{kp}	flip test	AP ^{kp}	AP ^{box}
9	2	ResNet-50-FPN	✓		30.2	63.9
25	2	ResNet-50-FPN	✓		30.8	64.5
49	2	ResNet-50-FPN	✓		31.6	65.2
83	2	ResNet-50-FPN	✓		32.7	66.3
83	3	ResNet-50-FPN	✓		33.8	67.9
83	3	ResNet-101-FPN	✓		35.9	68.4
83	3	ResNet-50-FPN			0.0	66.7
83	3	ResNet-50-FPN	✓	✓	35.6	68.3

The Impact of Different Design Modules

We conduct the ablation study on a quarter of the training set of DeepFashion2. Our baseline model is made up of a

³The performance of Match R-CNN (Ge et al. 2019) is reported in <https://github.com/switchablenorms/DeepFashion2>.



Figure 4: The visualization results with backbone ResNet-50 on the validation set of DeepFashion2. Our cascade keypoint-guided feature aggregation network can accurately detect fashion and its landmarks on the various of scenarios (*e.g.*, occlusion, rotation and different sizes).

feature extractor and a one-stage head module, whose performance on DeepFashion2 dataset is shown in Table 3. The performance gain provided by a three-stage cascade is drastic for clothing detection, while marginal for landmark detection. It is reasonable because the number of selected keypoints are 9 by default, which is much less than the total number of keypoints (294) defined in DeepFashion2. Upon adapting the multi-column feature extraction strategy, more information is reserved, thus boosting the performance of both clothing detection and landmark detection.

Table 3: Comparison of the two proposed feature aggregation modules on the Deepfashion2 dataset. We use ResNet-50-FPN as the backbone, and aggregate features from 9 points961205.

agg. module	clothing detection			landmark estimation		
	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅
kp. encoding	65.8	77.9	72.1	31.7	65.4	25.3
kp. selection	61.3	72.5	64.7	29.5	61.9	22.3

Extension to the FLD dataset

To further evaluate the effectiveness of our method, we compare our KGDet model with the baseline models on the FLD

(Liu et al. 2016b) dataset. The results are shown in Table 4. Given that our KGDet takes the lead by the AP and AP₇₅ metrics, we conclude that our method features higher localization precision, which owes to the keypoint supervision.

Table 4: Performance of **clothing detection** on the FLD (Liu et al. 2016b) dataset. We use ResNet-50-FPN as the backbone for all the models below.

method	AP	AP ₅₀	AP ₇₅
RetinaNet (Lin et al. 2017)	69.3	90.4	77.8
FCOS (Tian et al. 2019)	68.7	89.9	78.0
RepPoints (Yang et al. 2019)	69.1	91.3	78.2
Ours	70.8	90.2	79.0

Conclusion

In this paper, we focus on the question that whether supervision of keypoints can help improve clothing detection. We provide two solutions to integrate keypoints into clothing detection, and show that the improvement does exist. Then we design the KGDet model as a extension of the two solutions to make the most of the potential of keypoints. We also show the positive correlation between keypoint estima-

tion and clothing detection. Our model outperforms recent popular detectors on the DeepFashion2 and FLD dataset.

References

- Caba Heilbron, F.; Pepik, B.; Barzelay, Z.; and Donoser, M. 2019. Clothing Recognition in the Wild using the Amazon Catalog. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.
- Cai, Z.; and Vasconcelos, N. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6154–6162.
- Chen, H.; Gallagher, A.; and Girrod, B. 2012. Describing clothing by semantic attributes. In *European conference on computer vision*, 609–623. Springer.
- Chen, Q.; Huang, J.; Feris, R.; Brown, L. M.; Dong, J.; and Yan, S. 2015. Deep domain adaptation for describing people based on fine-grained clothing attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5315–5324.
- Cheng, W.-H.; Song, S.; Chen, C.-Y.; Hidayati, S. C.; and Liu, J. 2020. Fashion Meets Computer Vision: A Survey. *arXiv preprint arXiv:2003.13988*.
- Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 764–773.
- Dong, H.; Liang, X.; Shen, X.; Wu, B.; Chen, B.-C.; and Yin, J. 2019. FW-GAN: Flow-navigated Warping GAN for Video Virtual Try-on. In *Proceedings of the IEEE International Conference on Computer Vision*, 1161–1170.
- Ge, Y.; Zhang, R.; Wang, X.; Tang, X.; and Luo, P. 2019. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5337–5345.
- Han, X.; Wu, Z.; Huang, P. X.; Zhang, X.; Zhu, M.; Li, Y.; Zhao, Y.; and Davis, L. S. 2017. Automatic spatially-aware fashion concept discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, 1463–1471.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.
- Hidayati, S. C.; You, C.-W.; Cheng, W.-H.; and Hua, K.-L. 2017. Learning and recognition of clothing genres from full-body images. *IEEE transactions on cybernetics* 48(5): 1647–1659.
- Kiapour, M. H.; Yamaguchi, K.; Berg, A. C.; and Berg, T. L. 2014. Hipster wars: Discovering elements of fashion styles. In *European conference on computer vision*, 472–488. Springer.
- Lee, S.; Oh, S.; Jung, C.; and Kim, C. 2019. A Global-local Embedding Module for Fashion Landmark Detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.
- Li, Y.; Tang, S.; Ye, Y.; and Ma, J. 2019. Spatial-Aware Non-Local Attention for Fashion Landmark Detection. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 820–825. IEEE.
- Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.
- Liu, S.; Song, Z.; Liu, G.; Xu, C.; Lu, H.; and Yan, S. 2012. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3330–3337. IEEE.
- Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; and Tang, X. 2016a. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1096–1104.
- Liu, Z.; Yan, S.; Luo, P.; Wang, X.; and Tang, X. 2016b. Fashion landmark detection in the wild. In *European Conference on Computer Vision*, 229–245. Springer.
- Nie, X.; Feng, J.; Zhang, J.; and Yan, S. 2019. Single-stage multi-person pose machines. In *Proceedings of the IEEE International Conference on Computer Vision*, 6951–6960.
- Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.
- Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.
- Sidnev, A.; Trushkov, A.; Kazakov, M.; Korolev, I.; and Sorokin, V. 2019. Deepmark: One-shot clothing detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.
- Song, X.; Han, X.; Li, Y.; Chen, J.; Xu, X.-S.; and Nie, L. 2019. GP-BPR: Personalized Compatibility Modeling for Clothing Matching. In *Proceedings of the 27th ACM International Conference on Multimedia*, 320–328.
- Sun, X.; Xiao, B.; Wei, F.; Liang, S.; and Wei, Y. 2018. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 529–545.
- Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 9627–9636.
- Vasileva, M. I.; Plummer, B. A.; Dusad, K.; Rajpal, S.; Kumar, R.; and Forsyth, D. 2018. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 390–405.
- Wang, W.; Xu, Y.; Shen, J.; and Zhu, S.-C. 2018. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, 4271–4280.

Wang, X.; Wu, B.; and Zhong, Y. 2019. Outfit Compatibility Prediction and Diagnosis with Multi-Layered Comparison Network. In *Proceedings of the 27th ACM International Conference on Multimedia*, 329–337.

Yan, S.; Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; and Tang, X. 2017. Unconstrained fashion landmark detection via hierarchical recurrent transformer networks. In *Proceedings of the 25th ACM international conference on Multimedia*, 172–180.

Yang, Z.; Liu, S.; Hu, H.; Wang, L.; and Lin, S. 2019. Repoints: Point set representation for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 9657–9666.

Yoo, D.; Kim, N.; Park, S.; Paek, A. S.; and Kweon, I. S. 2016. Pixel-level domain transfer. In *European Conference on Computer Vision*, 517–532. Springer.

Yu, W.; Liang, X.; Gong, K.; Jiang, C.; Xiao, N.; and Lin, L. 2019. Layout-graph reasoning for fashion landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2937–2945.

Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; and Li, S. Z. 2018. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4203–4212.

Gated Heterogeneous Graph Representation Learning for Shop Search in E-Commerce

Xichuan Niu¹, Bofang Li², Chenliang Li^{3†}, Rong Xiao², Haochuan Sun², Honggang Wang², Hongbo Deng², Zhenzhong Chen¹

¹School of Remote Sensing and Engineering, Wuhan University, China

²Alibaba Group, Hangzhou, China

³Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China

¹{niuxichuan, zzchen}@whu.edu.cn, ²{bofang.lbf, xiaorong.xr, haochuan.shc, honggang.whg, dhh167148}@alibaba-inc.com, ³cllee@whu.edu.cn

ABSTRACT

In e-commerce search, vectorized matching is the most important approach besides lexical matching, where learning vector representations for entities (e.g., query, item, shop) plays a crucial role. In this work, we focus on vectorized search matching model for shop search in Taobao¹. Unlike item search, shop search is faced with serious behavior sparsity and long-tail problem. To tackle this, we take the first step to transfer knowledge from item search, i.e., leveraging items purchased under a query and the shops they belong to. Moreover, we propose a novel **gated heterogeneous graph learning** model (named GHL) to derive vector representations for entities. Both first-order and second-order proximity of queries and shops are exploited to fully mine the heterogeneous relationships. And to relieve long-tail phenomenon, we devise an innovative gated neighbor aggregation scheme where each type of entities (i.e., hot ones and long-tail ones) can benefit from the heterogeneous graph in an automatic way. Finally, the whole framework is jointly trained in an end-to-end fashion. Offline evaluation results on real-world data of Taobao shop search platform demonstrate that the proposed model outperforms existing graph based methods, and online A/B tests show that it is highly effective and achieves significant CTR improvements.

KEYWORDS

Heterogeneous Graph, Gated Mechanism, E-commerce, Shop Search

ACM Reference Format:

Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Honggang Wang, Hongbo Deng, Zhenzhong Chen. 2020. Gated Heterogeneous Graph Representation Learning for Shop Search in E-Commerce. In *Proceedings of the 29th ACM International Conference on Information and Knowledge*

¹<http://taobao.com/>

† Chenliang Li is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-6859-9/20/10...\$15.00

<https://doi.org/10.1145/3340531.3412087>

Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3340531.3412087>

1 INTRODUCTION

Nowadays as online shopping is becoming increasingly indispensable, e-commerce search has become an essential part of people's life. Taobao, one of the world's largest e-commerce platforms, provides plentiful search services to help customers find desired items or shops. Many efforts have been made to improving user's e-commerce search experience both in academic and industrial community.

In this paper, we focus on the task of vectorized search matching in the context of shop search in Taobao, where user queries and shops will be represented as vectors in a latent embedding space. The goal of this task is to estimate click-through rate (CTR) which is calculated by some function (e.g., inner product) over query and shop vectors. One specific challenge is how to obtain a semantic yet robust query and shop representation, which is critical for retrieving satisfactory search results.

A commonly used method is to feed vector representations into multi-layer perceptrons (MLPs) to model the non-linear feature interactions for relevance estimation. Recently, some graph embedding algorithms are introduced into search CTR optimization in a two-stage manner [7, 9]. These methods take advantage of graph embedding pretraining techniques [2, 5] and the learned node representations are further injected into CTR prediction. A latest model GIN [4] uses the end-to-end joint training method of co-occurrence commodity graph learning and CTR prediction, which achieves significant CTR improvements.

Though these approaches attain remarkable performance gains, they cannot simply generalize to shop search problem. Two-stage methods cannot support a direct end-to-end optimization, which becomes the bottleneck of their expressive ability in specific CTR prediction task. Also, GIN doesn't take into account long-tail issue that is a serious phenomenon existing in shop search.

As a special scenario, shop search possesses several distinct characteristics: 1) user behaviors are quite sparser than item search, which is supported by the fact that daily page views of shop search only account for 1% of item search approximately; 2) complex heterogeneous relationships are involved in the context of shop search and the scale of the resultant heterogeneous graph is pretty

large; 3) behavior sparsity gives rise to long-tail problem, most of the exposures are popular shops or queries of high frequency.

To address the above challenges, we propose a gated heterogeneous graph representation learning framework (GHL) for shop search. Firstly, to alleviate the behavior sparsity problem, we propose to transfer knowledge from item search. The item transaction history (*i.e.*, items purchased under a query and shops they belong to) are fully leveraged to enable better query understanding and retrieve more relevant shops. Incorporating knowledge of item search also enriches semantic information of long-tail queries and shops. Then to mine heterogeneous neighbor relationships encoded in the interaction graph, we exploit both first-order and second-order proximity between queries and shops. Weighted neighbor sampling strategy is employed by considering both scalability and efficiency. Moreover, we design a novel gated attentive neighbor aggregation scheme to assure that different types of entities (*i.e.*, long-tail ones and hot ones) can benefit from the heterogeneous graph automatically. Finally, the graph-based representations of queries and shops are fed into a two-tower architecture for relevance estimation and the whole framework is jointly trained in an end-to-end fashion.

The main contributions of this work can be summarized as follows:

- (1) We propose to transfer knowledge from item search to relieve behavior sparsity problem existing in shop search.
- (2) We utilize both first-order and second-order proximity of queries and shops encoded in the heterogeneous interaction graph and a novel gated attentive neighbor aggregation scheme is applied to enhance representation learning.
- (3) Offline evaluation and online A/B tests verify the effectiveness of our proposed model.

2 THE PROPOSED APPROACH

In this section, we present the proposed method in detail, as shown in Figure 1. The overall architecture is a two-tower framework which is implemented as several MLPs with *LeakyReLU* as activation function. The most crucial part is how to build inputs for two-tower, *i.e.*, graph-based representations of queries and shops. The outputs of two-tower are fed into some relevance estimation function (*e.g.*, vector inner product) to get the final CTR prediction score.

2.1 Neighbor Construction

The underlying graph structure determines how much information can be assimilated from neighbors in order to strengthen the representation of node itself. Next we describe how we construct heterogeneous neighbors using both first-order and second-order proximity between queries and shops.

First-order proximity. We make use of click-through data from shop search directly. That is, the shops clicked under a query construct the query’s first-order heterogeneous shop neighbors and vice versa. Due to the behavior sparsity issue, this first-order proximity derived from click-through data is very sparse. Therefore, we need to leverage second-order query-shop relationships through transferring knowledge from item search.

Second-order proximity. Item search contains more abundant information since it is the main search service provided by most

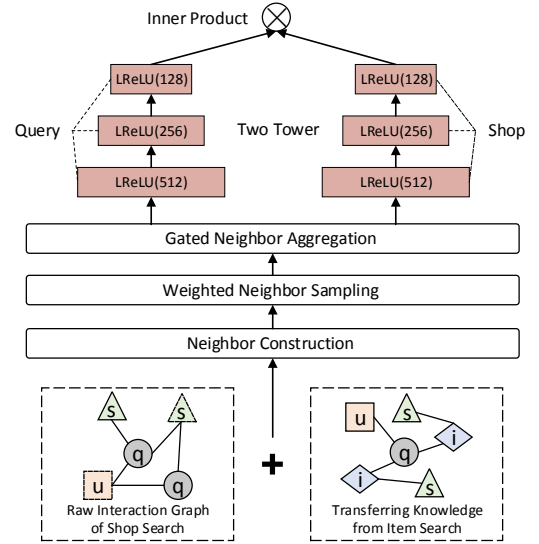


Figure 1: Framework of proposed GHL.

e-commerce platforms. Here, for an item purchased under a query and provided by a shop, the corresponding query and shop are formulated as each other’s second-order heterogeneous neighbor, respectively. Note that we did not utilize item click-through data since it might introduce much noise and purchase is a more reliable signal.

Weights assignment. After constructing heterogeneous neighbors, we further assign weight to each edge to indicate the relatedness between the query-shop pair. Combining information from both shop search and item search, we define the weight as the sum of numbers of the associated clicks and purchases.

2.2 Gated Attentive Neighbor Aggregation

In real-world e-commerce search, the size of heterogeneous neighbors of a query or a shop may vary significantly. To make computation more efficient, we adopt a **weighted neighbor sampling** strategy during training. The weights are calculated as stated in Section 2.1 and normalized over the heterogeneous neighbors of each node. Note that if some node has no neighbor, we instead use itself to fill the sampled set.

Next we describe how we aggregate information of sampled neighbors to enhance nodes (*i.e.*, queries and shops) representations. Take a piece of query q as example, let $\mathbf{h}_q$ be the embedding vector of q . Suppose we sample x heterogeneous neighbors (*i.e.*, shops) for q , the sampled set is denoted as $\mathcal{N}(q)$ and we have $|\mathcal{N}(q)| = x$. The neighbor aggregation is formulated as:

$$\mathbf{h}_{\mathcal{N}(q)} = \text{AGGREGATE}(\{\mathbf{h}_v, \forall v \in \mathcal{N}(q)\}) \quad (1)$$

where *AGGREGATE* is the aggregation function, such as averaging or max-pooling operation, $\mathbf{h}_{\mathcal{N}(q)}$ is the aggregated neighborhood representation of query q .

Here we propose to implement aggregation function with attention mechanism:

$$\mathbf{h}_{N(q)} = \sum_{v \in N(q)} \alpha_{q,v} \mathbf{h}_v \quad (2)$$

where $\alpha_{q,v}$ is the attention weight, which is calculated with the guidance of query representation $\mathbf{h}_q$:

$$\alpha_{q,v} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_q^\top [\mathbf{h}_q \| \mathbf{h}_v]))}{\sum_{k \in N(q)} \exp(\text{LeakyReLU}(\mathbf{a}_q^\top [\mathbf{h}_q \| \mathbf{h}_k]))} \quad (3)$$

where $\|$ denotes vector concatenation, $\mathbf{a}_q$ is attention parameter for query, and LeakyReLU is the activation function.

We can simply take $\mathbf{h}_{N(q)}$ as final representation of query q to establish a shop-to-shop framework. However, we need to discriminate the importance of aggregated neighborhood representation for different types of queries. For hot queries, the embedding vectors appear to be more critical since they are associated with too many neighbors that may hold massive noise. For long-tail queries, the heterogeneous neighbors usually are more decisive as we do not have sufficient prior knowledge of themselves.

Hence, inspired by GRU [1], we design a novel gated aggregation scheme to fuse the initial embedding vectors and corresponding neighborhood representations:

$$\mathbf{z} = \sigma(\mathbf{W}_z \cdot [\mathbf{h}_q \| \mathbf{h}_{N(q)} \| \mathbf{h}_q^{pv}] + \mathbf{b}_z) \quad (4)$$

$$\mathbf{r} = \sigma(\mathbf{W}_r \cdot [\mathbf{h}_q \| \mathbf{h}_{N(q)} \| \mathbf{h}_q^{pv}] + \mathbf{b}_r) \quad (5)$$

$$\hat{\mathbf{h}}_q = \text{LeakyReLU}(\mathbf{W} \cdot [(\mathbf{r} * \mathbf{h}_q) \| \mathbf{h}_{N(q)}] + \mathbf{b}) \quad (6)$$

$$\tilde{\mathbf{h}}_q = (1 - \mathbf{z}) * \mathbf{h}_q + \mathbf{z} * \hat{\mathbf{h}}_q \quad (7)$$

where $\mathbf{z}$ and $\mathbf{r}$ stand for update gate and reset gate respectively. In Eq. (4)-(5), $\mathbf{h}_q^{pv}$ represents **query pv (page view) features** which are used for distinguishing hot queries and long-tail queries. In practice, we embed discretized query page views and concatenate the resultant embeddings over different granularities to form $\mathbf{h}_q^{pv}$. As will be shown in experiments, these additional features boost the model performance explicitly.

Analogously, we can derive the final representation of shop s , denoted as $\tilde{\mathbf{h}}_s$. The gated attentive neighbor aggregation for shop is identical except that the network parameters are distinct from the query side.

2.3 End-to-End Joint Training

After obtaining $\tilde{\mathbf{h}}_q$ and $\tilde{\mathbf{h}}_s$, we feed them into two-tower architecture to get the predicted click-through rate p_{qs} as shown in Figure 1. The outputs of the last layer of two-tower are carried out via a vector inner product and the result is sent into *sigmoid* function to meet the binary constraint.

The objective function is defined as cross entropy loss:

$$\mathcal{L} = - \sum_{(q,s)} y_{qs} \log(p_{qs}) + (1 - y_{qs}) \log(1 - p_{qs}) \quad (8)$$

where y_{qs} is the ground truth label (*i.e.*, clicked or not clicked) of pair $\langle q, s \rangle$.

In the training process, we leverage mini-batch training technique and the details of end-to-end joint training are presented in Algorithm 1.

Algorithm 1: Mini-batch Implementation of GHL

```

1 Initialize model parameters  $\Theta$  randomly;
2 for iteration in 1, 2, ... do
3   Pick a mini-batch of  $B$  query-shop pairs  $\langle q, s \rangle$  and the
     corresponding pv features and labels  $y_{qs}$ ;
4   Sample by weight ( $B \times 2x$ ) heterogeneous neighbors
      $N(q), N(s)$ ;
5   Forward propagation to get  $p_{qs}$  according to Eq. (2)-(7);
6   Compute gradients  $\nabla \mathcal{L}(\Theta)$  according to Eq. (8);
7   Update model:  $\Theta = \Theta - \epsilon \nabla \mathcal{L}(\Theta)$ ;
8 end
```

3 EXPERIMENTS

3.1 Experimental Setup

Dataset. We evaluate our proposed model based on a large-scale real-world dataset collected from Taobao users' search log with a period of 10 days. The dataset is comprised of 20 million queries, 7 million shops and 3.3 billion interactions, leading to data sparsity rate of 10^{-5} . The graph data contains 30 million nodes and 200 million edges, of which 70 percent are transferred from item search. We choose the first nine days as training set and leave tenth day for testing, which is called *Normal* test set. Moreover, we filter out a *Long Tail* test set where the queries or shops only appear once in training set.

Competitors. We conduct experiments with several competitors widely used in industrial community. (1) **Base**: the baseline two-tower model for large-scale search matching task, where we feed into two-tower network the query and shop initial embedding vectors and keep numbers of neurons of each layer as same as the proposed GHL for fair comparison. (2) **PinSage** [8]: the state-of-the-art GNN-based web-scale recommender system with GraphSage [3] as the backbone model. (3) **GAT** [6]: the self attention based GNN model. (4) **GHL**: our proposed method. Besides, we also report the performance of two variants of our model: **GHL-avg** replaces attentive aggregator (*i.e.*, Eq. (2)) with average pooling and **GHL-pv** removes the pv (page view) features during gated fusion.

Evaluation Metrics. We adopt three commonly used performance metrics for offline evaluation: *Area Under the receiver operating characteristic Curve (AUC)*, *Group AUC (GAUC)*, *Hit Ratio (HR@1)*. *GAUC* is different from *AUC* as it measures the discrepancy between predictions and labels under each query and sum over all queries using the frequency as weight. Higher metric values demonstrate better search matching performance. Note that in our scenario a 0.001 increment means significant improvement.

3.2 Offline Evaluation

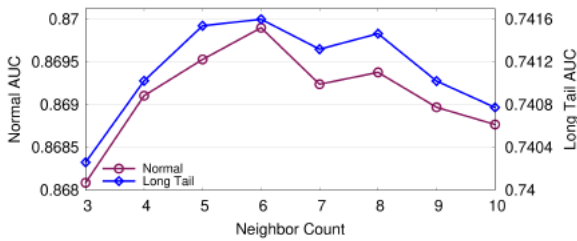
The experimental results of different methods on two test datasets (*i.e.*, *Normal*, *Long Tail*) are presented in Table 1. As can be seen, GHL significantly outperforms all compared algorithms under two

Table 1: Overall performance of GHL and competitors.

Model	Normal			Long Tail		
	AUC	GAUC	HR@1	AUC	GAUC	HR@1
Base	0.8650	0.8528	0.7631	0.7195	0.6491	0.8794
PinSage	0.8657	0.8522	0.7637	0.7232	0.6534	0.8813
GAT	0.8593	0.8456	0.7519	0.7146	0.6448	0.8772
GHL-avg	0.8682	0.8545	0.7649	0.7382	0.6678	0.8883
GHL-pv	0.8669	0.8534	0.7640	0.7325	0.6613	0.8827
GHL	0.8699	0.8560	0.7661	0.7416	0.6709	0.8926

scenarios, especially on *Long Tail* test set. Specifically, the relative performance gains of *AUC*, *GAUC*, *HR@1* over the best baseline are 0.49%, 0.38%, 0.31%; 2.54%, 2.68%, 1.28% for *Normal* and *Long Tail* test set respectively. PinSage and GAT perform worse than GHL because they cannot fully tap the potential of heterogeneous relationships between queries and shops, and they do not take long-tail phenomenon into consideration. Moreover, GHL-avg attains inferior results to attentive aggregator and the decline of GHL-pv proves the contribution of incorporating pv features.

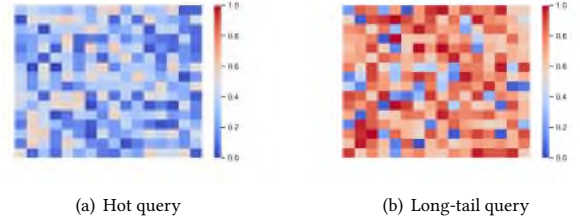
Neighbor count. We now analyze the effects of number of sampling neighbors. We vary neighbor count from 3 to 10 while keeping the other parameters fixed. The *AUC* values on both *Normal* and *Long Tail* datasets are depicted in Figure 2. We observe that the optimal results are obtained when neighbor count is 6 and either sampling more or less neighbors will lead to performance degradation. It is reasonable since a smaller neighbor count cannot provide enough information while a larger one may introduce too much noise.

**Figure 2: Comparison of different neighbor counts.**

Case study. We further conduct some queries case study to visualize the update gate values (*i.e.*, z in Eq. (4)) in Figure 3. Figure 3(a) is a case of a hot query (*perfume*) whose representation relies more on its initial embedding vector while Figure 3(b) is another case where the long-tail query (*scarifier*) benefits more from heterogeneous neighbors. This verifies the correctness of gated fusion strategy in order to discriminate different types of queries.

3.3 Online A/B Test

We design an online A/B test (*i.e.*, bucket test) to further evaluate the performance of GHL. Table 2 shows online CTR improvements over Base model during 3 consecutive days. The long-tail shops are

**Figure 3: Visualization of update gate values, which are re-formed as matrices to ease display.****Table 2: Online CTR improvements of A/B test.**

Time	T	T+1	T+2
whole bucket	2.67%	0.52%	3.70%
long-tail shops	19.71%	18.41%	24.34%

chosen on the basis of less than 10 exposures in training set. The results demonstrate the effectiveness of GHL on the shop search CTR prediction task, especially in solving long-tail problem.

4 CONCLUSION

In this paper, we introduce a gated heterogeneous graph representation learning framework for shop search in e-commerce. A serious problem, *i.e.*, behavior sparsity, is alleviated through transferring knowledge from item search. Both first-order and second-order proximity of queries and shops contained in the heterogeneous graph are mined to enhance representations. A novel gated attentive neighbor aggregation strategy is leveraged such that different types of entities can benefit from heterogeneous relationships in an automatic manner. Experiments on offline and online real-world dataset demonstrate the effectiveness of our proposed model, especially for long-tail queries and shops.

REFERENCES

- [1] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*. 1724–1734.
- [2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.
- [3] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.
- [4] Feng Li, Zhenrui Chen, Pengjie Wang, Yi Ren, Di Zhang, and Xiaoyu Zhu. 2019. Graph Intention Network for Click-through Rate Prediction in Sponsored Search. In *SIGIR*. 961–964.
- [5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*. 701–710.
- [6] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [7] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD*. 839–848.
- [8] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*. 974–983.
- [9] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. In *WWW*. 2390–2400.

EdgeDIPN: a Unified Deep Intent Prediction Network Deployed at the Edge

Long Guo, Lifeng Hua, Rongfei Jia

Fei Fang, Binqiang Zhao

Alibaba Inc

{leo.gl,issac.hlf,rongfei.jrf,mingyi.ff,binqiang.zhao}@alibaba-inc.com

Bin Cui

School of EECS & Key Laboratory of High Confidence

Software Technologies (MOE), Peking University

bin.cui@pku.edu.cn

ABSTRACT

With the rapid growth of e-commerce in recent years, e-commerce platforms are becoming a primary place for people to find, compare and ultimately purchase products. To improve online shopping experience for consumers and increase sales for sellers, it is important to understand user intent accurately and be notified of its change timely. In this way, the right information could be offered to the right person at the right time. To achieve this goal, we propose a unified deep intent prediction network, named EdgeDIPN, which is deployed at the edge, i.e., mobile device, and able to monitor multiple user intent with different granularity simultaneously in real-time. We propose to train EdgeDIPN with multi-task learning, by which EdgeDIPN can share representations between different tasks for better performance and saving edge resources in the meantime. In particular, we propose a novel task-specific attention mechanism which enables different tasks to pick out the most relevant features from different data sources. To extract the shared representations more effectively, we utilize two kinds of attention mechanisms, where the multi-level attention mechanism tries to identify the important actions within each data source and the inter-view attention mechanism learns the interactions between different data sources. In the experiments conducted on a large-scale industrial dataset, EdgeDIPN significantly outperforms the baseline solutions. Moreover, EdgeDIPN has been deployed in the operational system of Alibaba. Online A/B testing results in several business scenarios reveal the potential of monitoring user intent in real-time. To the best of our knowledge, EdgeDIPN is the first full-fledged real-time user intent understanding center deployed at the edge and serving hundreds of millions of users in a large-scale e-commerce platform.

PVLDB Reference Format:

Long Guo, Lifeng Hua, Rongfei Jia, Fei Fang, Binqiang Zhao, and Bin Cui.

EdgeDIPN: a Unified Deep Intent Prediction Network Deployed at the Edge. PVLDB, 14(3): 320-328, 2021.

doi:10.14778/3430915.3430922

1 INTRODUCTION

E-commerce platforms are becoming a primary place for people to find, compare and ultimately purchase products. One of the fundamental questions that arises in e-commerce is to understand

user intent, which allows for providing better services for both sellers and customers. In reality, a user's intent is changing all the time when surfing the e-commerce app. For example, a user's intent may change from browsing to buying when she finds some goods of interest. It is important to be notified of the change of user intent timely, which can provide users with brand new proactive experiences by offering the the right information at the right time. However, previous work only deals with the offline user intent prediction [7, 18, 23, 24, 27] and cannot monitor user intent in real-time. The reasons are twofold. First, previous work cannot predict users' real-time intent with a high accuracy limited by the representation capability of traditional coarse-grained behavior data. Second, it is difficult to monitor user intent in real-time in traditional cloud-based computing architecture due to the high communication cost.

In order to overcome the aforementioned difficulty when monitoring user intent in real-time, we propose to deploy our user intent understanding system at the edge, i.e., mobile devices, following the idea of edge computing. There are several advantages to deploy the system at the edge. First, it offers us the opportunity to unlock the potential of the vast untapped data created by the connected devices. In our system, we collect a new type of user interactive behavior data, which we call touch-interactive behavior and contributes a lot when predicting user's real-time intent. Second, it can reduce the communication cost between the cloud and the devices significantly. During the Alibaba 2019 Double 11 Shopping Festival, EdgeDIPN serves more than 0.5 billion customers without suffering from the traditional peak traffic problem of the platform on that day. Third, it can greatly increase the response speed of EdgeDIPN by moving the model to the data instead of the data to the model. The response time of EdgeDIPN is between 20 ~ 50 ms on different devices, which is immune to the influence of the network traffic and can achieve the real-time requirement. Last but not least, it enables us to create new business scenarios in the e-commerce platform based on the real-time user intent.

However, monitoring user intent in real-time at the edge is challenging. First, it is necessary to build a computationally lightweight and memory efficient system, because edge resources are typically resource-constrained. Second, the touch-interactive behavior is more fine-grained and contains less semantic information compared with traditional browse-interactive behavior. Therefore, it is challenging to extract useful features from these data to improve the prediction performance. To solve these challenges, we propose a unified deep intent prediction network, named EdgeDIPN, which is able to monitor multiple user intent with different granularity simultaneously in real-time. To save edge resources, we propose

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 14, No. 3 ISSN 2150-8097.
doi:10.14778/3430915.3430922

to train EdgeDIPN with multi-task learning. In more details, we extract a universal shared representation based on users' behavior data from several views, based on which multiple user intent with different granularity could be derived. In this way, edge resources can be reduced dramatically, because we do not need to build a model for each intent separately. We improve the performance of multi-task learning from two aspects. First, we utilize two kinds of attention mechanisms to extract the shared representation more effectively, where the multi-level attention mechanism tries to identify the important actions within each data source and the inter-view attention mechanism learns the interactions between different data sources. Second, we propose a novel task-specific attention mechanism which enables different tasks to pick out the most relevant features from different data sources. The contribution of the paper can be summarized as follows:

- We propose a unified deep intent prediction network, named EdgeDIPN, which is deployed at the edge and able to monitor multiple user intent with different granularity simultaneously in real-time.
- We propose to train EdgeDIPN with multi-task learning by sharing representations among several intent, which can save edge resources dramatically. In addition, several attention mechanisms are proposed to improve the performance.
- We evaluate the performance of EdgeDIPN by conducting extensive experiments on a large-scale industrial dataset. The results show the superiority of EdgeDIPN in predicting user intent. In particular, EdgeDIPN has been deployed in the operational system of Alibaba. Online A/B testing results show the benefits of monitoring user intent in real-time.

2 PROBLEM STATEMENT

2.1 Dataset

We build two types of user interactive behavior dataset which depict users from different views, i.e., the novel touch-interactive behavior collected from the edge and the traditional browse-interactive behavior. In particular, the touch-interactive behavior is further divided into the swipe and tap interactive behavior. The touch-interactive behavior records a user's fine-grained behavior when surfing in the app and thus contains rich user behavior patterns.

The swipe-interactive behavior includes four types of basic actions, i.e., *Open Page*, *Leave Page*, *Swipe* and *Tap*. A user's swipe-interactive track is a time sequence of actions, consisting of these four basic types of actions. Each action has a timestamp and a page index to identify when and where the action occurs. In addition, the positional coordinates of the action on the touch screen are also recorded (i.e., we record the tap position for the *Tap* action and the start and end positions for the *Swipe* action.). The duration presents how long the action lasts. For each action at a data point, we extract 14 features. Among these features, the time duration of a swipe, time gap between two actions and positional coordinates of actions (i.e., tap position X/Y, swipe start position X/Y, swipe end position X/Y and swipe length on X/Y) are continuous variables. Page indices, action indices and swipe directions (i.e., left/right and up/down) are categorical variables. We conduct discretization on all the raw features to ensure unified inputs for EdgeDIPN. The discretization of the continuous variables is described as follows:

- **Position.** The positional coordinates of actions are discretized according to the resolution of the touch screen. We divide the width and length of the screen into 17 and 25 uniform segments corresponding to X and Y, respectively, for one-hot vectors encoding.
- **Swipe Length.** The length of a swipe is encoded into a one-hot vector, the length of which is as twice as the length of the one-hot vectors of position encoding. The reason of applying twice length is that, for a swipe track, we consider the direction of the swipe.
- **Time Gap and Duration.** We apply a step function to encode time gaps between actions and swipe duration as follow:

$$y = \begin{cases} x/f_s, & x < f_b \\ x/f_b + 9, & f_b \leq x < 10 \times f_b \\ 19, & x \geq 10 \times f_b \end{cases}$$

where $\{f_s = 100, f_b = 1000\}$ are used for time gap, and $\{f_s = 25, f_b = 250\}$ are used for time duration.

The tap-interactive behavior records the information associated with the tap actions. A user's tap-interactive track is a time sequence of tap actions. Each action has a timestamp and page index to identify when and where the action occurs. There is also an event id to identify whether a user taps on a page or a button. If a button is tapped, the button name is also recorded. We extract 3 raw features (i.e., event index, page index and button index) for each tap action, all of which are categorical variables.

The browse-interactive behavior represents the typical behavior users conduct on products. It includes five types of actions, i.e., *Browse*, *Search*, *collect*, *Add to cart* and *Purchase*. A user's browse-interactive track is a time sequence of these actions. We extract 6 raw features for each action (i.e., type index, top category index, leaf category index, page index, page stay time and timestamp), where the type index represents the type of an action.

2.2 Problem Formulation

EdgeDIPN is used to monitor multiple user intent with different granularity in real-time, which enables us to send the right information to the right user at the right time. The typical predicted user intent is listed as follows.

- **Real-time Purchasing Intent:** whether a user would buy some goods within one hour.
- **Long-time Purchasing Intent:** whether a user would buy some goods within the current day.
- **Real-time Cart Intent:** whether a user would add some goods to cart within one hour.
- **Retention Intent:** whether a user would visit the app the next day.

Besides the above global intent, EdgeDIPN also predicts some fine-grained intent such as category preference. Note that a user's category preference is measured by the probability of whether the user would buy some goods belonging to the category. We also set the category preference as real-time and long-time preference, which means the probability of the order being placed within one hour and one week, respectively. In addition, the category is further divided into the top category and leaf category, following the settings of the category label system in Taobao app.

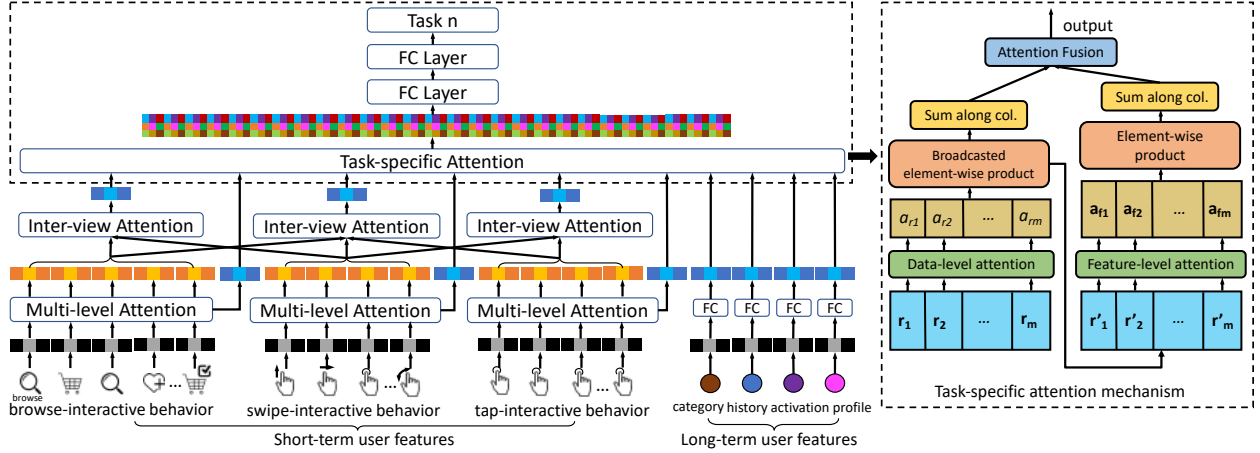


Figure 1: The model architecture of EdgeDIPN.

In our work, each intent is treated as a separate task and we adopt the binary classification for each task. Note that we do not use multi-label classification for the category preference, because the softmax operation would make the model too big to fit at the edge due to the large number of categories to be predicted.

3 MODEL ARCHITECTURE

We propose a unified deep intent prediction network deployed at the edge, named EdgeDIPN, to predict multiple user intent with different granularity simultaneously, as shown in Figure 1.

3.1 Embedding Layer

In the discretization process, the raw values of every feature are encoded into one-hot vectors. Then the one-hot vectors are used as the inputs of EdgeDIPN. As the inputs are high dimensional binary vectors with limited representation capacity, we use the embedding layer to transform them into low dimensional dense representations. The embedding operation follows the table lookup mechanism. In more details, each feature is corresponding to one embedding matrix. For example, the embedding matrix of the *Button Index* feature in the tap-interactive behavior is represented as $E_{button} = [e_1; e_2; \dots; e_{n_b}] \in \mathbb{R}^{n_e \times n_b}$, where $e_i \in \mathbb{R}^{n_e}$ represents an embedding vector with dimension n_e (i.e., 10), and n_b (i.e., 500) represents the number of buttons that a user can tap. The embedding vector of the *Button Index* feature can then be obtained as $e_{button} = E_{button} \cdot b_{button} \in \mathbb{R}^{n_e}$, where $b_{button} \in \mathbb{R}^{n_b}$ is the one-hot vector of the *Button Index* feature. In addition, to capture the temporal order information in users' behavior sequence used by the following multi-level attention mechanism, we add the time feature to each action defined as the elapsed time (i.e., seconds) between the moment the action happens and the current moment. However, the model size would become too large after the embedding operation if we directly use the original elapsed time as the input feature. We notice that the impact of the time feature to the current intent decays as the elapsed time increases. Therefore, we propose a new discretization method which would not influence the model performance and can save model space significantly at

the same time. Specifically, we split the time feature into multiple granularity, i.e., the continuous time feature in range of $[2^k, 2^{(k+1)})$ is mapped to the discrete feature $(k+1)$ while $[0, 1)$ is mapped to 0. At last, for each action, all the embedded features are concatenated into a vector and fed into a fully-connected layer for reshape. Specifically, each behavior sequence S can be encoded as a matrix E , i.e., $E = \{e_1, e_2, \dots, e_n\} \in \mathbb{R}^{n \times d_e}$, where d_e is the dimension of e_i and n is the number of actions contained in S . Moreover, as shown in Figure 1, except for the behavior sequences which can capture the short-term user features, we also collect some statistics for long-term user features.

3.2 Multi-level Attention Layer

The user interactive behaviors are all time sequence of actions. A traditional method is to use RNN to model the long-term dependencies between actions. The adoption of RNN can eliminate the need for extensive feature engineering. However, there are two limitations which hinder us from adopting RNN in EdgeDIPN. First, edge resources are typically resource-constrained and thus places a high demand on the memory cost and inference time of the model. Under this context, RNN is not suitable for use at the edge due to the complex data dependencies and limited parallelism. Second, RNN cannot achieve the best performance because of the unique characteristics of the touch-interactive behavior sequence with plenty of noisy actions operated by users inadvertently.

Therefore, we propose to use the attention mechanism to identify the important actions in the behavior sequence and reduce the interference from the noisy actions. Specifically, we use a softmax layer to determine the weights of different behavior actions. In this case, those actions contributing more to the intent understanding are given larger weights. The weights $a = \{a_1, a_2, \dots, a_n\}$ are computed as follows:

$$a_t = \text{softmax}(h_t), h_t = w_{s2} \tanh(W_{s1} e_t), \quad (1)$$

where e_t is embedding vector of the t -th action, $W_{s1} \in \mathbb{R}^{d_e \times d_e}$, $w_{s2} \in \mathbb{R}^{d_e}$, and a_t is the weight calculated for e_t . Then we can sum up the behavior sequence E according to the calculated weights to get a representation vector $r \in \mathbb{R}^{d_e}$, i.e., $r = \sum_{i=1}^n a_i e_i$.

However, there is a limitation in the representation vector $\mathbf{r}$, which usually focuses on a specific component of the sequence, like a special set of actions. In reality, there can be multiple components in a sequence that together form the overall preferences of a user. More importantly, EdgeDIPN is proposed to predict multiple user intent and different intent may correspond to different component of the behavior sequence. Therefore, we perform multiple hops of attention and get multiple representation vectors that focus on different component of the behavior sequence, following the idea of the sentence embedding task in [15]. Suppose we want h different components to be extracted from the sequence, we just need to extend $\mathbf{w}_{s2}$ into a matrix $\mathbf{W}_{s2} \in \mathbb{R}^{h \times d_e}$. Formally, we have

$$\mathbf{A} = \text{softmax}(\mathbf{W}_{s2} \tanh(\mathbf{W}_{s1} \mathbf{E}^T)), \mathbf{R} = \mathbf{A} \mathbf{E}, \quad (2)$$

where the $\text{softmax}()$ is performed along the second dimension of its input. As a result, the weight vector $\mathbf{a} \in \mathbb{R}^n$ becomes a weight matrix $\mathbf{A} \in \mathbb{R}^{h \times n}$, and the representation vector $\mathbf{r} \in \mathbb{R}^{d_e}$ becomes an embedding matrix $\mathbf{R} \in \mathbb{R}^{h \times d_e}$. After the multi-level attention layer, for each behavior sequence, we can get two outputs, i.e., the representation vector $\mathbf{r}_f \in \mathbb{R}^{d_e}$ by flattening and reshaping $\mathbf{R}$, and the reweighed sequence by applying the element-wise product on $\mathbf{E}$ and $\mathbf{R}$ as follows:

$$\mathbf{r}_m = \text{reduce\_mean}(\mathbf{R}), \mathbf{V} = \mathbf{E} \odot \mathbf{r}_m. \quad (3)$$

The reweighed sequences are then sent to the following layer.

3.3 Inter-view Attention Layer

To better fuse the views extracted from different behavior sequences, we adopt the inter-view attention mechanism [9] that learns the inter-view relations between different views, as shown in Figure 1. The inter-view attention takes two views as inputs, and emphasizes the important interactions between the two views using attention scores. In particular, for each action in one view, we calculate its relevance with all the actions in the other view. In this way, the asynchronous interactions between actions in these two sequences can be captured effectively. The inter-view attention $\text{IA}(\mathbf{V}_s, \mathbf{V}_b)$ is formulated as follows, taking the swipe-interactive view $\mathbf{V}_s$ and the browse-interactive view $\mathbf{V}_b$ for example:

$$\begin{aligned} \text{IA}(\mathbf{V}_s, \mathbf{V}_b) &= \mathbf{A}_s(\mathbf{V}_b, \mathbf{V}_s, \mathbf{V}_s) \odot \mathbf{A}_b(\mathbf{V}_s, \mathbf{V}_b, \mathbf{V}_b), \\ \mathbf{A}_s(\mathbf{V}_b, \mathbf{V}_s, \mathbf{V}_s) &= \text{softmax}\left(\frac{\mathbf{V}_b \mathbf{V}_s^T}{\sqrt{2d}}\right) \mathbf{V}_s, \\ \mathbf{A}_b(\mathbf{V}_s, \mathbf{V}_b, \mathbf{V}_b) &= \text{softmax}\left(\frac{\mathbf{V}_s \mathbf{V}_b^T}{\sqrt{2d}}\right) \mathbf{V}_b, \end{aligned} \quad (4)$$

where $\mathbf{A}_s \in \mathbb{R}^{n \times d_e}$ and $\mathbf{A}_b \in \mathbb{R}^{n \times d_e}$ are attentive representations of $\mathbf{V}_s$ and $\mathbf{V}_b$, respectively, and the element-wise product $\odot$ is used to model the interactions between $\mathbf{A}_s$ and $\mathbf{A}_b$. Note that $\text{IA}(\mathbf{V}_s, \mathbf{V}_b)$ is a symmetric operation and returns the representation $\mathbf{R}_{sb}$ between $\mathbf{V}_s$ and $\mathbf{V}_b$. We can calculate $\mathbf{R}_{st}$ and $\mathbf{R}_{tb}$ following the same procedure. At last, we can get three representations $\mathbf{r}_{sb} \in \mathbb{R}^{d_e}$, $\mathbf{r}_{st} \in \mathbb{R}^{d_e}$ and $\mathbf{r}_{tb} \in \mathbb{R}^{d_e}$ by applying the reduce_mean operation.

3.4 Multi-task Layer

In this work, we need to monitor multiple user intent simultaneously. Considering the limited computational and memory resources at the edge, it is not reasonable to build a model for each

intent separately. Therefore, we propose to train EdgeDIPN with multi-task learning, by which a universal shared representation could be learned and used to derive multiple intent simultaneously. In this way, edge resources can be reduced dramatically.

Prior approaches to simultaneously learning multiple tasks commonly concatenate the learned representations from several data sources and send the concatenated representation to the following task-specific output layers [2, 8, 18]. However, we find that different data representations have different contributions in different tasks. Therefore, the performance of each task is highly dependent on an appropriate choice of weighting for each representation. Searching for an optimal weighting is prohibitively expensive and difficult to resolve with manual tuning. Therefore, we propose a novel task-specific attention mechanism to learn the weights for each task automatically and enable each task to pick out the most relevant features from different data representations.

The task-specific attention consists of two components, i.e., data-level attention and feature-level attention, as shown in Figure 1. These two components allow each task the selection of effective data representations and latent features, respectively.

Data-level Attention. The data-level attention is used to learn the weights for each data representation given a task. Suppose we have m representations extracted from different data sources shown as the blue rectangles in Figure 1, i.e., $\mathbf{D} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m\}$, the weight vector $\mathbf{a}_r$ for the given task on different data representations is:

$$\begin{aligned} \mathbf{a}_r &= \{a_{r1}, a_{r2}, \dots, a_{rm}\}, \\ a_{ri} &= \frac{\exp(e(\mathbf{r}_i))}{\sum_{j=1}^m \exp(e(\mathbf{r}_j))}, \\ e(\mathbf{r}_i) &= \mathbf{w}_{r2} \tanh(\mathbf{W}_{r1} \mathbf{r}_i + b_{r1}), \end{aligned} \quad (5)$$

where $\mathbf{W}_{r1} \in \mathbb{R}^{d_e \times d_e}$, $\mathbf{w}_{r2} \in \mathbb{R}^{d_e}$, and a_{ri} is the weight calculated for the data representation $\mathbf{r}_i$. After getting the weight vector $\mathbf{a}_r \in \mathbb{R}^m$, we extract a weighed data representation matrix $\mathbf{M}$ by applying the broadcasted element-wise product on $\mathbf{D}$ and $\mathbf{a}_r$.

Feature-level Attention. After getting the weighed data representation $\mathbf{M}$, we go one step further and let each task to pick out the relevant features from $\mathbf{M}$. The intuition behind this idea is that different features in the same data representation may have different impact on different tasks. The feature-level attention is a natural extension of the data-level attention at the feature level. Specifically, instead of computing a single scalar weight for each weighed data representation $\mathbf{r}'_i$, the feature-level attention computes a feature-wise weight vector for $\mathbf{r}'_i$, formalized as follows:

$$\begin{aligned} \mathbf{a}_f &= \{\mathbf{a}_{f1}, \mathbf{a}_{f2}, \dots, \mathbf{a}_{fm}\}, \\ \mathbf{a}_{fi} &= \frac{\exp(e(\mathbf{r}'_i))}{\sum_{j=1}^m \exp(e(\mathbf{r}'_j))}, \\ e(\mathbf{r}'_i) &= \mathbf{W}_{f2} \tanh(\mathbf{W}_{f1} \mathbf{r}'_i + b_{f1}), \end{aligned} \quad (6)$$

where $\mathbf{W}_{f1} \in \mathbb{R}^{d_e \times d_e}$, the vector $\mathbf{w}_{r2} \in \mathbb{R}^{d_e}$ in Eq. 5 is replaced with the matrix $\mathbf{W}_{f2} \in \mathbb{R}^{d_e \times d_e}$ in Eq. 6, and the $\text{softmax}()$ is performed along the second dimension of its input. As a result, we can get a feature-wise weight matrix $\mathbf{a}_f \in \mathbb{R}^{m \times d_e}$. Then the new feature-wise weighed data representation $\mathbf{M}'$ is obtained by applying the element-wise product on $\mathbf{M}$ and $\mathbf{a}_f$, i.e., $\mathbf{M}' = \mathbf{M} \odot \mathbf{a}_f$.

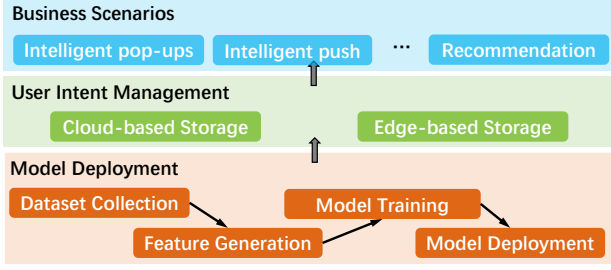


Figure 2: System overview under edge computing.

Attention Fusion. This component is used to combine the two weighted data representation M and M' , as shown in Figure 1. The combination is accomplished by a dimension-wise fusion gate:

$$\begin{aligned} F &= \text{sigmoid}(WM + W'M' + b), \\ O &= F \odot M + (1 - F) \odot M', \end{aligned} \quad (7)$$

where $W, W' \in \mathbb{R}^{d_e \times d_e}$ and $b \in \mathbb{R}^{d_e}$.

For each task, the extracted data representation O is then fed into two fully connected layers to further learn the combination of task-specific features automatically. Given T binary classification tasks, the loss function of task t_i is:

$$\mathcal{L}_i = -\frac{1}{N} \sum_{(x,y) \in \mathcal{D}} (y \log p_i(x) + (1-y) \log(1-p_i(x))) \quad (8)$$

where $\mathcal{D}$ is the training set with size D , x is the input of the network and y is the label, and $p_i(x)$ represents the predicted probability of sample x for task t_i . The global loss is:

$$\mathcal{L} = \sum_{i=1}^T \mathcal{L}_i + \lambda \|(\mathbf{A}\mathbf{A}^T - \mathbf{I})\|_F^2 \quad (9)$$

where the term $\|(\mathbf{A}\mathbf{A}^T - \mathbf{I})\|_F^2$ is used to punish redundancy between different vectors in the weight matrix $\mathbf{A}$ mentioned in Section 3.2, and thus the multi-level attention mechanism can focus on different component of the behavior sequence.

4 SYSTEM OVERVIEW

The structure of our user intent prediction system is illustrated in Figure 2. For model deployment, the procedure is as follows. First, user behavior data is collected from the mobile app usage logs to construct the dataset. Second, features are generated to build the training samples. Third, EdgeDIPN is trained in the cloud based on the samples. Finally, the trained model is deployed to each user’s mobile device. In addition, we build a platform where the model deployment procedure is automatically conducted. As a result, we have built a user intent understanding system in several apps owned by Alibaba for more than a year, such as Taobao, Tmall, Xianyu and Freshippo with little manual work. For user intent management, EdgeDIPN automatically predicts multiple user intent using the real-time features collected from the mobile device at a predefined time interval. The predicted user intent is stored in two forms, i.e., the cloud-based storage where only the predicted intent is sent to the cloud and the edge-based storage where the intent is directly stored at the edge, and used to support the above business scenarios.

5 EXPERIMENTS

5.1 Experimental Setup

Dataset Statistics. We conduct the experiments on a large-scale industrial dataset collected from Taobao. The dataset contains normal users’ daily interaction information when using our app. We collect 800,000 users’ behaviors for two weeks. For each user, we randomly truncate about 400 groups of samples on average. In total, we obtain 300 million groups of samples. Each group contains the user profile feature, the user statistic feature, and the three behavior sequences (the default length is 256 and padding 0 for short ones). The labels of the multiple tasks are then tagged for each group based on the timestamp of the last action. We use samples in the first 13 days for training and samples in the last day for evaluation.

Compared Methods. We compare EdgeDIPN with the following baseline methods. In particular, we conduct experiments to verify the effect of each component in EdgeDIPN. In the following, we introduce the compared methods briefly.

- **GBDT**, which is the competitive gradient boosting model widely used in the industrial environment.
- **EdgeDIPN-tradition**, which does not use the touch interactive behavior data. Our goal is to see the benefit of using the new touch-interactive behavior.
- **EdgeDIPN-single**, which does not use multi-task learning and train a separate model for each intent.
- **EdgeDIPN-v0**, which replaces the multi-level attention mechanism with the GRU architecture and removes the inter-view attention and task-specific attention mechanism from EdgeDIPN. This method is used to verify the effect of each component in EdgeDIPN.
- **EdgeDIPN-fastgrnn**, which replaces the GRU architecture in EdgeDIPN-v0 with a more advanced RNN architecture named Fastgrnn [13] with smaller model size and lower prediction costs suited to be used at the edge. In addition, we also test the low rank version of Fastgrnn which can further reduce the mode size, denoted as EdgeDIPN-fastgrnnlr.
- **EdgeDIPN-v1**, which replaces GRU in EdgeDIPN-v0 with the multi-level attention mechanism.
- **EdgeDIPN-v2**, which adds the inter-view attention mechanism to EdgeDIPN-v1.
- **EdgeDIPN-v3**, which adds the data-level attention mechanism to EdgeDIPN-v2.
- **EdgeDIPN-v4**, which adds the feature-level attention mechanism to EdgeDIPN-v3 but the feature-level attention mechanism picks out relevant features from the original data representations $\mathbf{D}$ instead of $\mathbf{M}$.
- **EdgeDIPN**, which is the complete model.

Evaluation Metrics. The metric used in our experiments is Area Under the Curve (AUC), which is insensitive to class imbalance and suitable to our experiments. Since EdgeDIPN is used to predict multiple intent simultaneously, we define a metric name for each intent, as shown in Table 1.

Experimental Details. The dimension of embedding features is 10 and the dimension of the reshaped embedding vector is set to $d_e = 64$. The parameters h and λ corresponding to the multi-level attention mechanism are set to 10 and 0.01, respectively. Layers of MLP is set by $128 \times 64 \times 2$. In our experiments, we test 8 binary

Table 1: The metric name of each user intent.

User Intent	Metric Name
Real-time Purchasing Intent	RPAUC
Long-time Purchasing intent	LPAUC
Real-time Cart Intent	RCAUC
Retention Intent	RAUC
Real-time Top Category Preference	RTCAUC
Long-time Top Category Preference	LTCAUC
Real-time Leaf Category Preference	RLCAUC
Long-time Leaf Category Preference	LLCAUC

classification tasks, i.e., $T = 8$. EdgeDIPN is trained with SGD, using the Adam optimizer [12] with initial hyper-parameters of $\epsilon = 10^{-3}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set the mini-batch size to be 1024 and apply exponential decay, in which the learning rate starts at 0.001 and decay rate is set to 0.9. We train EdgeDIPN using a distributed TensorFlow with 1 parameter server and 200 workers.

5.2 Experimental Results

Results of different models. Table 2 shows the performance of the evaluated models. We have the following observations. (1) EdgeDIPN outperforms the baseline methods EdgeDIPN-tradition on all the tasks by a significant margin between 0.9% and 5.3% in terms of AUC. The improvement of EdgeDIPN over EdgeDIPN-tradition reveals the value of adopting the touch-interactive behavior to depict users from different views. (2) The multi-level attention mechanism is effective in identifying the important actions and reduce the interference from the noisy actions. We can see that EdgeDIPN-v1 performs better than EdgeDIPN-v0, EdgeDIPN-fastgrnn and EdgeDIPN-fastgrnnlr for all the tasks. (3) The interview attention mechanism plays an important role in EdgeDIPN. As shown, EdgeDIPN-v2 outperforms EdgeDIPN-v1 for all the tasks. (4) With the help of multi-task learning, EdgeDIPN can increase AUC by about 0.85% on average compared with EdgeDIPN-single. This shows the superiority of adopting multi-task learning in EdgeDIPN, which can not only save edge resources dramatically but also improve the performance greatly. In particular, the comparison of EdgeDIPN, EdgeDIPN-v4, EdgeDIPN-v3 with EdgeDIPN-v2 demonstrate the effectiveness of our proposed task-specific attention mechanism. The reason is that different data representations as well as their features play different role in different tasks. By adopting the task-specific attention mechanism, we can let each task choose the most relevant data representations and features, and thus reduce the mutual influence between tasks. We can also see that EdgeDIPN outperforms EdgeDIPN-v4 for all the tasks. This is because the feature-level attention cannot consider the relationship between data representations in EdgeDIPN-v4, which can be solved by adopting the data-level attention first in EdgeDIPN.

Impact of sequence length (Scalability). In this section, we evaluate the impact of different sequence lengths on the performance of the models. The results are shown in Table 3. We evaluate four different models, where Att, RNN, FG and FG-lr denote the model with the multi-level attention mechanism, GRU, Fast-grnn and Fastgrnnlr structure, respectively. As shown, the prediction

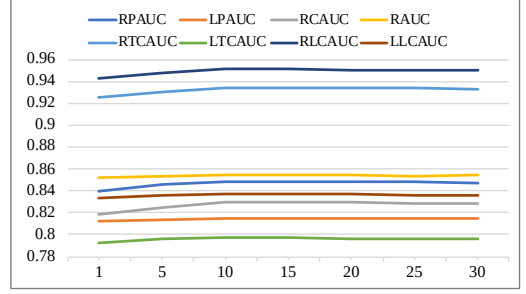


Figure 3: Impact of multiple components.

performance¹ of each model increases when the sequence length increases. This is because more useful features can be extracted from longer sequences. However, longer sequences result in larger memory cost and longer inference time, which creates a heavy burden for the deployment at the edge. Compared to RNN, Att scales better when the sequence length increases. As shown, when the sequence length is 512, Att can reduce the model size and inference time by 43.9% and 56.2%, respectively. Although FG and FG-lr can reduce the model size and inference time significantly compared with RNN, they perform much worse in terms of AUC in our scenario. Compared with FG and FG-lr, Att not only performs much better in terms of AUC, but also has comparable and even better performance in model size and inference time, which further demonstrates its effectiveness.

Impact of multiple components. In this section, we evaluate the impact of the parameter h in the multi-level attention mechanism. We vary h from 1 to 30 for each task and the results are shown in Figure 3. We can see the best performance appears after h reaches 10. This demonstrates that there can be multiple components in a sequence that together form the overall preferences of a user. However, it is not always true that a larger h results in better performance. As shown, there is no more benefit when h is larger than 10. Therefore, an appropriate h should be selected.

5.3 Online A/B Testing

5.3.1 Intelligent Pop-ups. Pop-ups are one of the most powerful tools on e-commerce platforms which can reach most of the traffic the platforms receive. If used intelligently, they can play an important role in turning the visitors into sales. In this section, we introduce a novel pop-up strategy based on the real-time category preference predicted by EdgeDIPN in online traffic of Taobao. One important task in Taobao is to promote the conversions of the inactive customers. Since these customers are hard to place an order, we design a special promotion strategy, introduced as follows. We first use the pop-ups to pop some product from the candidate products to the inactive customers. If one customer taps the pop-ups, she would be directed into a shopping page where the products are specially designed for the inactive customers and can promote the conversions effectively. Therefore, one of the key tasks in this strategy is to improve the click-through rate of the pop-ups.

¹Due to the page space limitation, we only show the AUC for the real-time purchasing intent. The performance is similar for other intent.

Table 2: Comparison of different models.

Model	RPAUC	LPAUC	RCAUC	RAUC	RTCAUC	LTCAUC	RLCAUC	LLCAUC
GBDT	0.7911	0.7747	0.8057	0.8421	0.8933	0.7826	0.9102	0.8203
EdgeDIPN-tradition	0.7960	0.7856	0.8159	0.8457	0.9069	0.7883	0.9267	0.8263
EdgeDIPN-single	0.8396	0.8068	0.8208	0.8512	0.9241	0.7889	0.9407	0.8286
EdgeDIPN-fastgrnn	0.8367	0.8049	0.8181	0.8496	0.9153	0.7834	0.9291	0.8214
EdgeDIPN-fastgrnnlr	0.8342	0.8037	0.8160	0.8490	0.9105	0.7807	0.9262	0.8185
EdgeDIPN-v0	0.8413	0.8093	0.8191	0.8519	0.9208	0.7895	0.9397	0.8278
EdgeDIPN-v1	0.8435	0.8097	0.8221	0.8522	0.9231	0.7901	0.9418	0.8283
EdgeDIPN-v2	0.8458	0.8106	0.8239	0.8527	0.9266	0.7909	0.9433	0.8292
EdgeDIPN-v3	0.8471	0.8126	0.8271	0.8534	0.9317	0.7947	0.9471	0.8341
EdgeDIPN-v4	0.8482	0.8138	0.8291	0.8538	0.9335	0.7955	0.9499	0.8357
EdgeDIPN	0.8489	0.8152	0.8298	0.8543	0.9346	0.7969	0.9514	0.8368

Table 3: Impact of sequence length.

Sequence Length	64	128	256	512
RPAUC (RNN)	0.8395	0.8418	0.8463	0.8499
RPAUC (FG)	0.8371	0.8384	0.8419	0.8432
RPAUC (FG-lr)	0.8353	0.8359	0.8396	0.8389
RPAUC (Att)	0.8413	0.8457	0.8489	0.8508
Model Size (RNN)	1.38MB	1.73MB	2.42MB	3.83MB
Model Size (FG)	1.06MB	1.38MB	1.69MB	2.45MB
Model Size (FG-lr)	0.88MB	1.19MB	1.53MB	2.21MB
Model Size (Att)	1.03MB	1.30MB	1.57MB	2.15MB
Infer Time (RNN)	19.5ms	32.9ms	59.8ms	111.6ms
Infer Time (FG)	15.6ms	24.5ms	41.3ms	81.7ms
Infer Time (FG-lr)	14.2ms	23.6ms	35.5ms	59.8ms
Infer Time (Att)	13.7ms	22.6ms	31.2ms	48.9ms

To improve the CTR of the pop-ups, one common method is to pop the hot products to the customers, which is usually a good strategy. A more effective method is to pop the products based on customers' preference. However, the inactive customers have very few history behaviors and thus traditional recommendation methods are not suited to be used in this scenario. The advantage of our EdgeDIPN is that we can utilize users' rich real-time behaviors to predict users' intent. Therefore, we propose to pop the products based on customers' real-time leaf category preference.

We set four pop-ups strategies as follows.

- **Hot Strategy** where the popped product is selected based on the purchase frequency of the products.
- **Category-1 Strategy** where the leaf category ranking first in the category preference is selected and then the hot product belonging to this category is selected.
- **Category-2 and Category-3 Strategy** are used to compare with **Category-1**, where the leaf category ranking second and third are selected, respectively.

The online A/B testing results are shown in Figure 4, where the x-axis and y-axis represents the date time and the CTR, respectively. It is notable that the category-1 strategy contributes up to **50.1%** CTR promotion in average compared with the hot strategy in the online

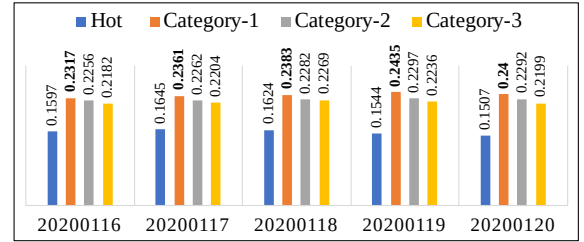


Figure 4: The results of different pop-up strategies.

traffic. The reason is that EdgeDIPN can understand a user's real-time preference by utilizing the rich real-time behaviors conducted by the user, which helps the pop-ups to select more appropriate product. In addition, the category-1 strategy increases the CTR by **4.4%** and **7.3%** compared with the category-2 and category-3 strategy, respectively. This indicates the effectiveness of EdgeDIPN in ranking user category preference.

5.3.2 Intelligent Push. Push notifications are another powerful tool to boost the user engagement on e-commerce platforms. Compared with the pop-ups, push notifications cause less disturbance to users and thus can be used at any moment when a user is visiting the app. In this section, we introduce a novel push notification strategy for the soon-to-expire coupons based on the real-time purchasing intent predicted by EdgeDIPN in online traffic of Taobao.

We choose two different types of coupons (denoted as C_1 and C_2) and set two push strategies to compare the performance.

- **Random Strategy** where the system chooses the moment randomly to push the notification of the soon-to-expire coupons to a user when she is visiting.
- **Model Strategy** where the system uses the score given by EdgeDIPN and a fixed threshold to decide the push moment.

The users selected to get this notification will be pushed a message in Taobao's mobile application. We use the CTR improvement I_C as the evaluation metric, defined as follows:

$$I_C = \frac{CTR_{model} - CTR_{rand}}{CTR_{rand}}, \quad (10)$$

where CTR_{model} and CTR_{rand} are the CTR of the push notifications with model and random strategy, respectively.

Table 4: The results of different push notification strategies.

	Num. of Users	I_{C_1}	I_{C_2}
Random Strategy	18.3M	/	/
Model Strategy	18.3M	+9.02%	+5.32%

We hypothesize that the users with a low purchasing intent may not be interest in these coupons and would be disturbed by these push notifications. On the contrary, the users with a high purchasing intent may be attracted by these coupons and click the related push notifications. Therefore, we set the threshold $t_l = 0.3$ which has the best online performance. The users whose real-time purchasing intent score given by EdgeDIPN is large than t_l are selected to get this push notification in the model strategy bucket.

As shown in Table 4, the model strategy contributes up to **9.02%** and **5.32%** CTR promotion compared with random strategy in the two different types of coupons in the large scale online traffic. The reason is that EdgeDIPN can help the system to understand users' real-time purchasing intent and push the coupon notification to the right person at the right time. Compared with random strategy, a reasonable model strategy relied on EdgeDIPN would result in a significant CTR improvement.

6 RELATED WORK

6.1 Multi-task Learning

Multi-task learning has been used successfully across various applications of machine learning. Ruder [20] presents an overview of multi-task learning in deep learning, where multi-task learning is typically done with either hard or soft parameter sharing of hidden layers. The hard parameter sharing method is the most commonly used multi-task learning approach, which shares the hidden layers between all tasks and keeps several task-specific output layers [4, 5, 8, 18]. The shared-bottom model structure suffers from optimization conflicts caused by task differences. To overcome this issue, Ma et al. [16] propose the MMoE method by sharing the expert submodels across all tasks with a gating network trained to optimize each task. However, MMoE is originally designed for DNNs where the shared-bottom model can be easily partitioned into small expert submodels to reduce the parameter size. This is not suited for EdgeDIPN with a complicated bottom model, which would result in a model with a large size and cannot be deployed at the edge. In soft parameter sharing, each task has its own model with its own parameters where the distance between the parameters is regularized [6, 17, 25]. However, compared to the hard parameter sharing methods, the soft parameter sharing methods have much more task-specific parameters and thus are not suited to be adopted at the edge. Recently, some work [3, 21] try to optimize multi-task learning from the perspective of loss functions. Previous methods usually use a weighed sum of losses, where the loss weights are uniform or manually tuned, and is hard to achieve the best performance. To overcome this issue, Kendall et al. [3] propose to automatically learn the weights by using the homoscedastic task uncertainty, while Sener et al. [21] cast multi-task learning as multi-objective optimization and aim to find a Pareto optimal solution. These methods are orthogonal to our work and can be

adopted in EdgeDIPN seamlessly. Compared with previous work, EdgeDIPN proposes a task-specific attention mechanism to let each task pick out the most relevant features from the extracted data representations and thus reduce the conflict between different tasks. Moreover, the task-specific attention mechanism is light-weight and very friendly to be used at the edge.

6.2 Mobile User State Prediction

Another related work is user state prediction based on the mobile phone usage history. In these studies, the usage history is used as features for machine learning models and the targeted user state varies for each study, such as emotion state [26], user trait [22], and mental health [1]. However, none of these studies implement an on-device inference for a real-time prediction [14]. Current on-device applications are mainly image recognition [10] and natural language processing such as smart reply [11]. Recently, Ochiai et al. [19] implement a MLP model on Android smartphone for the smartphone operation troubleshooting recommendation. We are the first to monitor multiple user intent on a large-scale e-commerce platform at the edge. In [9], we make a preliminary attempt at predicting user's real-time purchasing intent with a model named DIPN. EdgeDIPN distinguishes from DIPN in several aspects. First, EdgeDIPN is a full-fledged real-time user intent understanding center and demonstrates its effectiveness in several business scenarios while DIPN just focuses on the purchasing intent prediction. Second, EdgeDIPN replaces the RNN component with the multi-level attention mechanism, which can not only improve prediction performance but also save edge resources. Last but not least, EdgeDIPN proposes a novel task-specific attention mechanism for the multi-task learning.

7 CONCLUSION

In this paper, we propose a unified deep intent prediction network, named EdgeDIPN, which is deployed at the edge and able to monitor multiple user intent with different granularity simultaneously in real-time. To save edge resources, we propose to train EdgeDIPN with multi-task learning by sharing representations among several intent. In addition, several attention mechanisms are proposed to improve the performance of EdgeDIPN. Experimental results on a large-scale industrial dataset shows the superiority of EdgeDIPN. In particular, EdgeDIPN has been deployed in the operational system of Alibaba. Online A/B testing on several business scenarios shows the benefits of monitoring users' intent in real-time.

Note that EdgeDIPN is a universal and scalable architecture which extracts an effective user embedding from various user behaviors and thus new user intent can be easily integrated. In addition, EdgeDIPN is the first full-fledged on-device attempt at the domain of e-commerce, which can offer the opportunity to create more novel business scenarios in the e-commerce platforms.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61702016 and 61832001, the National Key Research and Development Program of China (No.2018YFB1004403), Beijing Academy of Artificial Intelligence (BAAI), and Okawa Research Grant.

REFERENCES

- [1] B. Cao, L. Zheng, C. Zhang, P. S. Yu, A. Piscitello, J. Zulueta, O. Ajilore, K. Ryan, and A. D. Leow. Deepmood: Modeling mobile phone typing dynamics for mood detection. In *KDD*, page 747–755, 2017.
- [2] Y. Chen, Y. Ma, X. Mao, and Q. Li. Multi-task learning for abstractive and extractive summarization. *Data Science and Engineering*, 4(1):14–23, 2019.
- [3] R. Cipolla, Y. Gal, and A. Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, June 2018.
- [4] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008.
- [5] L. Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: an overview. In *ICASSP*, 2013.
- [6] L. Duong, T. Cohn, S. Bird, and P. Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL-IJCNLP*, pages 845–850, 2015.
- [7] Z. Gharibshah, X. Zhu, A. Hainline, and M. Conway. Deep learning for user interest and response prediction in online display advertising. *Data Science and Engineering*, 5(1):12–26, 2020.
- [8] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.
- [9] L. Guo, L. Hua, R. Jia, B. Zhao, X. Wang, and B. Cui. Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In *KDD*, page 1984–1992, 2019.
- [10] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. V. Gool. Ai benchmark: Running deep neural networks on android smartphones. In *ECCV Workshops*, 2018.
- [11] A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukacs, M. Ganea, P. Young, and et al. Smart reply: Automated response suggestion for email. In *KDD*, page 955–964, 2016.
- [12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [13] A. Kusupati, M. Singh, K. Bhatia, A. Kumar, P. Jain, and M. Varma. Fastgrnn: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network. In *NIPS*, pages 9017–9028, 2018.
- [14] S. Li, D. Zhai, P. Du, and T. Han. Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled iot networks. *Science China Information Sciences*, 62(2):29307, 2018.
- [15] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. In *ICLR*, 2017.
- [16] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*, page 1930–1939, 2018.
- [17] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. *CoRR*, abs/1604.03539, 2016.
- [18] Y. Ni, D. Ou, S. Liu, X. Li, W. Ou, A. Zeng, and L. Si. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *KDD*, pages 596–605, 2018.
- [19] K. Ochiai, K. Senkawa, N. Yamamoto, Y. Tanaka, and Y. Fukazawa. Real-time on-device troubleshooting recommendation for smartphones. In *KDD*, page 2783–2791, 2019.
- [20] S. Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.
- [21] O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. *CoRR*, abs/1810.04650, 2018.
- [22] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti. Predicting user traits from a snapshot of apps installed on a smartphone. *SIGMOBILE Mob. Comput. Commun. Rev.*, 18(2):1–8, 2014.
- [23] A. Toth, L. Tan, G. D. Fabbriozzi, and A. Datta. Predicting shopping behavior with mixture of rnns. In *ACM SIGIR Forum*, 2017.
- [24] R. Xu, J. Du, Z. Zhao, Y. He, Q. Gao, and L. Gui. Inferring user profiles in social media by joint modeling of text and networks. *Science China Information Sciences*, 62(11):219104, 2019.
- [25] Y. Yang and T. M. Hospedales. Trace norm regularised deep multi-task learning. *CoRR*, abs/1606.04038, 2016.
- [26] X. Zhang, W. Li, X. Chen, and S. Lu. Moodexplorer: Towards compound emotion detection via smartphone sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4), 2018.
- [27] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai. Deep interest network for click-through rate prediction. In *KDD*, pages 1059–1068, 2018.

A General Method For Automatic Discovery of Powerful Interactions In Click-Through Rate Prediction

Ze Meng^{1,*†}, Jinnian Zhang^{2,*†}, Yumeng Li³, Jiancheng Li³, Tanchao Zhu³, Lifeng Sun^{4,5,†}

¹Beijing Key Lab of Networked Multimedia, Dept. of Computer Science and Technology, Tsinghua University, China

²Dept. of Electrical and Computer Engineering, University of Wisconsin Madison, USA

³Alibaba Group, China

⁴Key Laboratory of Pervasive Computing, Tsinghua University, Ministry of Education, China

⁵BNRist, Dept. of Computer Science and Technology, Tsinghua University, China

[†]mengz18@tsinghua.org.cn, jinnian.zhang@wisc.edu, sunlf@tsinghua.edu.cn

Abstract

Modeling powerful interactions is a critical challenge in Click-through rate (CTR) prediction, which is one of the most typical machine learning tasks in personalized advertising and recommender systems. Although developing hand-crafted interactions is effective for a small number of datasets, it generally requires laborious and tedious architecture engineering for extensive scenarios. In recent years, several neural architecture search (NAS) methods have been proposed for designing interactions automatically. However, existing methods only explore limited types and connections of operators for interaction generation, leading to low generalization ability. To address these problems, we propose a more general automated method for building powerful interactions named AutoPI. The main contributions of this paper are as follows: AutoPI adopts a more general search space in which the computational graph is generalized from existing network connections, and the interactive operators in the edges of the graph are extracted from representative hand-crafted works. It allows searching for various powerful feature interactions to produce higher AUC and lower Logloss in a wide variety of applications. Besides, AutoPI utilizes a gradient-based search strategy for exploration with a significantly low computational cost. Experimentally, we evaluate AutoPI on a diverse suite of benchmark datasets, demonstrating the generalizability and efficiency of AutoPI over hand-crafted architectures and state-of-the-art NAS algorithms.

CCS Concepts

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**; • **Mathematics of computing** → *Nonconvex optimization*.

Keywords

Click-through Rate Prediction, Gradient-based Neural Architecture Search, Feature Interaction, Interaction Ensemble

1 Introduction

Predicting the user’s clicking probability on a given item is a crucial task in a recommender system. Essentially, the key challenge in CTR prediction is how to model powerful feature interactions effectively. The classical methods employ a single type of interactive operator to model the fixed-order feature interactions in an explicit

manner. For example, Factorization Machines (FM) based methods [25, 30] learn 2nd-order cross features using inner product. And its variant, High-order Factorization Machine (HOFM) [1], adopted Analysis of Variance (ANOVA) to approximate high-order cross features, while only marginal improvement was observed despite using more parameters. Deep learning introduces a more effective operator, Single-layer Perceptron (SLP) [9], which can be stacked to form Multi-layer Perceptron (MLP) to generate high-order feature interactions in an implicit manner. However, the representation ability of the cross features generated from a single type of operator is limited. As an enhancement to MLP, incorporating explicit lower-order feature interactions in deep learning models achieves better performance, namely, multi-interaction ensemble, such as Wide&Deep [2], DeepFM [7] and xDeepFM [14].

However, manually developing the interactions for generalizing across diverse scenarios is computationally intensive and is a black box optimization without an optimality guarantee. Instead of fixing the order of interactions, learning the order of interactions or adaptively identifying important groups of features within a predefined order attract increasing attention. Specifically, by using the multi-head self-attention [27], the relevant features can be grouped to form meaningful arbitrary-order cross features. AFN [3] learns adaptive-order cross features via a logarithmic transformation layer. And more advanced NAS methods, AutoFIS [18] automatically identify important cross features from all possible 2nd-order and 3rd-order combinations of features with inner product. As a generalization of AutoFIS, AutoGroup [17] can generate substantial interactions of any order in an automatic manner. AutoCross [20] explicitly search for useful high-order cross features in a tree-based search space by an efficient beam search method. However, the above methods generate the interactions by using only a single type of operator for all identified groups of features. In order to enable the automatic search for the best connections of operators for generating the interaction from each group, AutoFeature [12] encodes the subnetwork for each candidate group of features within a predefined order as a string. Note that each subnetwork can be considered as a direct acyclic graph (DAG) of operators. Then a tree of Naive Bayes classifiers is used for optimization. AutoCTR [26] defines three basic blocks from operators SLP, FM, and dot product (DP), and utilize the evolutionary algorithm to obtain the best connections of these blocks. Similarly, in AutoRec [29], five different blocks are created based on several operators such as SLP, FM, outer product, self-attention, etc. Three strategies (random,

*Both authors contributed equally to this research.

[†]Corresponding authors.

greedy, and Bayesian) are used to optimize the number and type of blocks for ensemble, and the hyperparameters in these blocks.

Nevertheless, these works only involve simple operators and limited possible connection types for building the interactions, leading to narrow search space. For generalization to a large number of scenes, we study a more general paradigm of discovering powerful interactions. We search the interactions of CTR prediction according to three key factors: (1) the order of interaction in each tower, (2) the types and connections of operators used to construct interactions, (3) the multi-tower structure for the ensemble of interactions. Specifically, we review various representative literature to extract and modularize significant operators to form the operation space and design the tailored cell-based computational graph containing more possible connections of operators. Each node in the graph represents a latent representation (e.g., intermediate cross features), and the edge is an operator from the operation space. Notably, we unify the concept of the order related to explicit and implicit interactions following [14]. For example, DeepFM contains two types of interactive operators, FM and SLP. FM explicitly produces 2nd-order cross features, while MLP implicitly generates 4th-order cross features because it has three SLPs. Thus, our method discovers powerful interactions by integrating multiple interactions, where each interaction consists of different numbers, types, and connections of operators. However, our search space exponentially increases with the aforementioned three key factors of the interactions, requiring an efficient search strategy. Inspired by DARTS [19], we describe an improved gradient-based method for exploration to improve efficiency. In detail, we introduce architecture parameters to relax the discrete set of operators and node connections, and utilize the Bi-level optimization algorithm to optimize the architecture and weights by gradient descent. Moreover, two different training techniques are proposed to alleviate a well-known problem in the weight-sharing mechanism, w.r.t. the performance gap [31].

To summarize, the main contributions of this paper can be highlighted as follows:

- Designing a general and efficient search space to generate powerful interactions, which consists of the operation space and the computational graph. The operation space is constructed via extracting and modularizing the representative interactive operators in existing literature, and we design a tailored cell-based computational graph for finding the best connections of the operators.
- A gradient-based search strategy is applied for exploration in our broad search space to improve efficiency. We introduce the architecture parameters for relaxing the discrete search space and use a Bi-level optimization algorithm to optimize the architecture and weights iteratively.
- Two training techniques are utilized for alleviating the performance gap between the continuous architecture encoding and the derived architecture.
- Extensive experiments demonstrate the generalization and efficiency of AutoPI on datasets in different scales. Comparing with SOTA methods, on average, our method increases the AUC by 0.46% and 0.70% on the public and private datasets, respectively. Furthermore, the gradient-based

search strategy can reduce 90% of the search overhead compared to the NAS baselines.

Our work is organized as follows. In Section 2, we summarize the related work on NAS and CTR prediction. Section 3 formally defines our problem. Section 4 presents the details of our method. In Section 5, we present the experimental results and detailed analysis. Finally, we conclude this paper and point out the future work in Section 6.

2 Related Work

Click-Through Rate Prediction. Traditional models aim to formulate the explicit fixed-order interactions for CTR prediction, such as LR, FM, Attention Factorization Machine (AFM), HOFM, and CrossNet [22]. Different from HOFM, the number of parameters in CrossNet only increases linearly with the input dimension.

Compared to polynomial interactions in FM, CrossNet, etc., complicated non-linear features created through neural networks [3, 8, 22, 27] can be more informative and thus improve the prediction performance. PNN [22] builds the inter-field interactions by using the inner product or outer product and feeds it to an MLP. NFM [8] has a similar architecture as PNN, but uses a Bi-Interaction layer to model the inter-field interactions. Instead of interactions generated by a single model, multi-interaction ensembles are known as the boost of performance by exploiting the advantages of different models. In Wide&Deep [2], it combines LR and Deep Neural Network (DNN), which outperforms each individual model. DeepFM [7] comprises DNN and FM, and other ensemble models include Deep&Cross [28] and xDeepFM [14]. These models have been verified to be effective on many public datasets. In recent years, some hand-crafted models try to learn the order of feature interactions. For example, AutoInt [27] makes use of the latest techniques, attention and residual networks, to generate non-linear features. AFN [3] learns the powers (orders) of each feature in interactions by a logarithmic transformation layer.

Neural Architecture Search. The primary motivation for NAS is to automate the laborious process of designing neural networks in different tasks. There are basically four existing frameworks for NAS [6]: evolution-algorithm based NAS [23], reinforcement-learning based NAS [21], Bayesian optimization based NAS [6], and gradient-based NAS [24]. Among various NAS methods, DARTS [19] and its variants [4, 5, 13, 15, 32] have become the most popular algorithms due to remarkable efficiency improvement in time cost. The core of DARTS is to utilize continuous relaxation of the architecture representation and apply Bi-level optimization to iteratively update the architecture and its weights by using gradient descent. The effectiveness of DARTS has been proved in both Computer Vision (CV) and Neural Language Processing (NLP) domains. Besides existing literature on NAS methods are focused on the tasks in CV and NLP, there is an increasing interest in applying NAS to recommender systems in recent years, such as AutoCTR, AutoFeature and AutoRec, etc. These methods are demonstrated to yield higher performance than human-crafted models.

3 Problem Definition

We summarize the general framework of CTR prediction in Figure 1. The deep CTR prediction models have three fundamental stages from a bottom-up perspective: (i) input transformation, (ii) modeling diverse feature interactions, and (iii) multi-interaction ensemble.

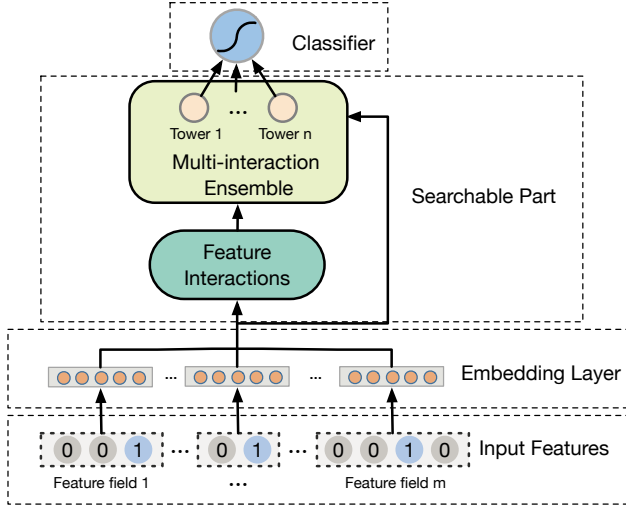


Figure 1: The CTR model consists of the fixed part (embedding layer and classifier) and the searchable part (feature interactions and multi-interaction ensemble).

We formally introduce the definitions of three subproblems in CTR prediction as follows:

DEFINITION 1. (Input Transformation) We represent the raw feature vector as $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_m]$, where m is the number of total fields, $\mathbf{x}_i$ is a one-hot feature representation of the i -th field, which is categorical (e.g., gender=male, name=Alan, age=25). Since the feature representations are very sparse and high-dimensional, we employ embedding layers to convert the sparse feature into a low dimensional and dense real-value vector $\mathbf{e}_i = \mathbf{V}_i \mathbf{x}_i$, where $\mathbf{V}_i$ is an embedding matrix for field i . Then, the output of the embedding layer will be a feature matrix $\mathbf{E} \in \mathbb{R}^{m \times k}$ by stacking multiple embedding vectors $[\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m]$, where k is the embedding size of each field. To avoid interference between the deep and shallow model [3], we adopt dual-embedding¹ (i.e., two independent embedding layers $\mathbf{V}^{\text{low}}$ and $\mathbf{V}^{\text{high}}$) to build lower-order and higher-order feature interactions, respectively.

DEFINITION 2. (Feature Interactions) The key problem is to determine which types and orders of interactions should be built to form meaningful cross features. We first redefine the order of interactions from the convention following [14]. Mathematically, Given an input feature matrix $\mathbf{E}$, a p -order cross feature is defined as:

$$\mathbf{E}^p = o^{(p-1)}(o^{(p-2)}(\dots(o^{(1)}(\mathbf{E}))\dots)). \quad (1)$$

Namely, the p -order cross feature $\mathbf{E}^p$ is generated by the composition of $p-1$ operators $o^{(1)}(\cdot), o^{(2)}(\cdot), \dots, o^{(p-1)}(\cdot)$ in order, such as FM and SLP. Traditionally, this is designed by domain experts based on their knowledge. In this paper, we tackle this problem with a novel method, i.e. NAS.

DEFINITION 3. (Multi-interaction Ensemble) In hand-crafted CTR prediction models, one of the most critical designs is the multi-interaction ensemble, which combines various orders and types of interactions to form a multi-tower structure. For final prediction, the cross features of all towers are concatenated and then delivered

to an SLP as follows:

$$\hat{y} = \sigma(\mathbf{w}^\top (\mathbf{e}^{(1)} \oplus \mathbf{e}^{(2)} \oplus \dots \oplus \mathbf{e}^{(n)}) + b), \quad (2)$$

where $\mathbf{w}$ is a column projection vector which linearly combines concatenated features, and b is the bias. Symbol $\oplus$ is the concatenation operation, and $\sigma(x) = 1/(1 + e^{-x})$ transforms the logits to users' clicking probabilities. Note that $\mathbf{e}^{(i)}$ is the flattened cross features of $\mathbf{E}^{(i)}$ from the i -th tower, and n is a hyperparameter that indicates the total number of towers. The *Logloss* is adopted for training the model, which is defined as follows:

$$\text{Logloss} = -\frac{1}{N} \sum_{i=1}^N (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)), \quad (3)$$

where y_i and $\hat{y}_i$ are the ground truth of user clicks and predicted CTR, respectively, and N is the number of training samples.

4 AutoPI

4.1 Method Overview

The work flow of AutoPI is shown in Figure 2. Roughly speaking, our method includes three parts: (i) search space (Section 4.2), (ii) search strategy (Section 4.3) and (iii) performance evaluation (Section 4.4). Specifically, we adopt the cell-based search space,

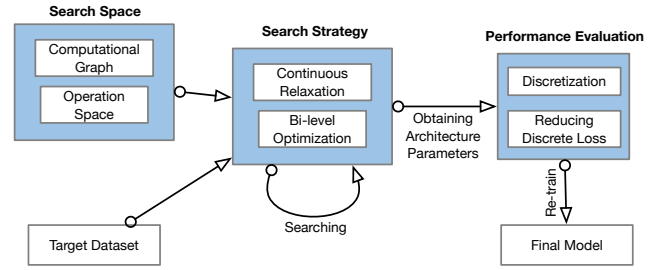


Figure 2: Method overview of AutoPI.

which consists of the computational graph (Section 4.2.1) and the operation space (Section 4.2.2). The efficient gradient-based search strategy is utilized to search for the best connections of operators. Its core is a continuous relaxation (Section 4.3.1) scheme for our discrete search space, leading to a differentiable learning objective for the joint optimization of the architecture and its weights (Section 4.3.2). We also apply the Batch Normalization [11] technique, because of its capability of dealing with numerical instability and improving fairness among operators (Section 4.3.3). After the search process, the discretization technique (Section 4.4.1) is used to obtain the discrete architecture from continuous architecture encoding. For alleviating the performance gap before and after discretization, temperature anneal and noisy skip-connection can be utilized in the search process (Section 4.4.2). The final procedure is training the model weights with the powerful architecture from scratch.

4.2 Search Space Design

We summarize the interactions of hand-crafted models as three critical factors: (1) The orders of interactions, w.r.t, how many operators should be combined to form an interaction. (2) Operators used to construct interactions, w.r.t, which operators should be selected and how operators are connected to build an interaction. (3) Multi-interaction ensemble, w.r.t, which interactions should be

¹For ease of reading, we omit the dual-embedding in Figure 1.

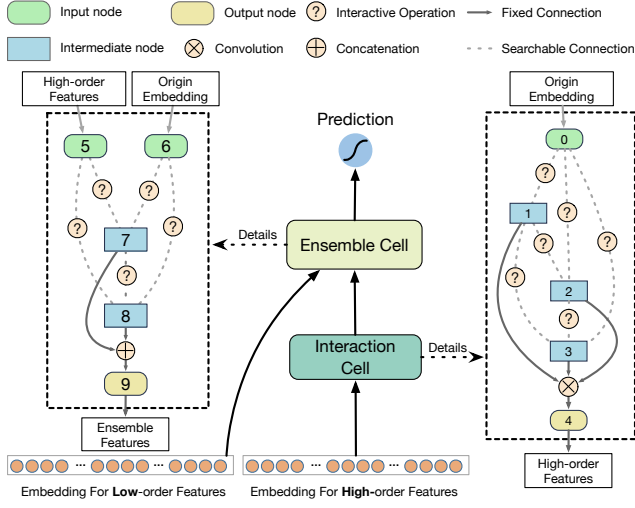


Figure 3: AutoPI decomposes the searchable part into an interaction cell and an ensemble cell. Each cell is represented as a DAG with several nodes, where each dashed line defines a set of candidate interactive operators.

integrated for final prediction. Therefore, our search space consists of the operation space and tailored cell-based computational graph. Each node in the graph represents for a latent representation (i.e., an arbitrary-order cross feature), and the edge is an operator from the operation space. For better generalization to various applications, the search space should contain sufficient nodes in the computational graph, while potentially encompassing distinct interactive operators in the operation space. And the multi-tower structure is contained in the graph for the multi-interaction ensemble to improve the performance. Note that the connections of operators in the multi-tower structure are also automatically optimized. As a result, we extract diverse interactive operators from representative hand-crafted works to form the operation space. Moreover, two types of computational cells are designed as building blocks of the graph, including an interaction cell and an ensemble cell.

4.2.1 Computational Graph. We first introduce two types of computational cells in this section. As shown in Figure 3, each cell is a DAG consisting of an ordered sequence of nodes, including the input node(s), N intermediate nodes and the output node, where N is a hyperparameter, indicating the maximum order in interaction cell and the number of towers in ensemble cell. Each node $\mathbf{X}^{(i)} \in \mathbb{R}^{m \times k}$ is a latent representation (i.e., a feature matrix), and each directed edge (i, j) is associated with an interactive operator $o^{(i,j)}$ that transforms $\mathbf{X}^{(i)}$. And each intermediate node is computed based on all of its predecessors

$$\mathbf{X}^{(j)} = \sum_{i < j} o^{(i,j)}(\mathbf{X}^{(i)}). \quad (4)$$

The task of learning the cell, therefore, reduces to choosing the best operators for its edges.

We have two types of cells to serve two main functions: (1) *the interaction cell formulates the higher-order feature interactions*. Specifically, the interaction cell has an input node (node 0) that is defined

as the input embedding produced from one of the dual-embedding layers, three intermediate nodes (node 1, 2, 3) representing the intermediate cross features, and an output node (node 4) that fuses the feature matrices of all intermediate nodes via a combination operation (i.e., convolution). (2) *The ensemble cell formulates the ensemble of lower-order and higher-order interactions*. Different from the interaction cell, the ensemble cell has two input nodes, where node 5 is the matrix of the higher-order cross features generated from the interaction cell, and node 6 is the input embedding matrix produced by another of dual-embedding layers. Note that the intermediate nodes (node 7, 8) in the ensemble cell function as the output from two towers. The output node 9 is a concatenation of all intermediate nodes. Since the number of intermediate nodes is configurable, the number of towers in the ensemble cell can be adapted to the scenarios. It is notable that although the cells are separate DAGs, instead of optimizing the cells sequentially, a joint optimization can also be applied.

In summary, we explicitly search for higher-order and lower-order interactions, and integrate them by a multi-tower structure. According to our experiments, sharing embedding layers between deep and shallow models leads to a bad performance. Therefore, We adopt the dual-embedding layers to avoid the gradient interference between different cells [3].

4.2.2 Operation Space. To search for various interactions, diverse interactive operators should be included in our search space. We extract and modulate interactive operators with the following considerations [26]:

- **Functionality:** Operators in our search space should complement and accommodate each other for applications in extensive scenarios. For example, complex operators (e.g., SLP, convolution layer, etc.) have strong learning ability, modeling more complex cross features. In contrast, simple operators (e.g., LR, SENET Layer, etc.) can learn general representation with a low risk of overfitting.
- **Operator granularity:** The fine-grained operators (e.g., FM, SLP, etc.) have substantial combinational flexibility than coarse-grained (e.g., AFM, MLP, etc.). Because the NAS method is sensitive to computational overhead, efficient operation space requires operators with lower time-space complexity. Thus fine-grained operators are preferable.
- **Dimension alignment:** In our computation graph, we need to maintain the dimension of the output feature $o(\mathbf{X}) \in \mathbb{R}^{m \times k}$ as the same as the input feature $\mathbf{X} \in \mathbb{R}^{m \times k}$.

According to the above principles, we extract and modify the operators from representative NAS and CTR prediction literature, such as LR, inner product, hadamard product, outer product, cross layer [28], none [19], skip-connection, self-attention [27], FM [25], logarithmic transformation layer [3], SENET layer [10], 2d convolution [16], 1d convolution, etc. After extensive experiments and careful comparison, we finally determine the following operators to form the operation space:

- **Skip-connection** : takes an input feature matrix $\mathbf{X} \in \mathbb{R}^{m \times k}$ and outputs the same feature matrix $\mathbf{X}' = \mathbf{X} \in \mathbb{R}^{m \times k}$, which is an identity mapping and does not increase the order of interactions.

- *SENET Layer* [10]: takes an input feature matrix $\mathbf{X} \in \mathbb{R}^{m \times k}$ and produces a weight vector $\mathbf{a} = [a_1, a_2, \dots, a_m]$, and then rescales $\mathbf{X}$ with vector $\mathbf{a}$ to obtain a weighted feature matrix $\mathbf{X}' = [a_1 \cdot \mathbf{x}_1, a_2 \cdot \mathbf{x}_2, \dots, a_m \cdot \mathbf{x}_m] \in \mathbb{R}^{m \times k}$.
- *Self-attention* [27]: takes an input feature matrix $\mathbf{X} \in \mathbb{R}^{m \times k}$ and outputs cross features $\mathbf{X}' \in \mathbb{R}^{m \times k}$ via the key-value attention mechanism.
- *FM*: takes an input feature matrix $\mathbf{X} \in \mathbb{R}^{m \times k}$ and produces an inner product vector $\mathbf{p} = \{(\mathbf{x}_j \cdot \mathbf{x}_i)\}_{(i,j) \in R_x}$, where $R_x = \{(i, j)\}_{i \in \{1, \dots, m\}, j \in \{1, \dots, m\}, i < j\}$, and then outputs the feature matrix $\mathbf{X}' = \mathbf{p} \cdot \mathbf{W} \in \mathbb{R}^{m \times k}$ via a linear transformation for dimension alignment.
- *Single-layer Perceptron*: takes an input flattened feature vector $\mathbf{x} \in \mathbb{R}^{mk}$ and outputs a feature matrix $\mathbf{X}' = \mathbf{x} \cdot \mathbf{W} \in \mathbb{R}^{m \times k}$ via a linear transformation.
- *1d Convolution*: takes an input feature matrix $\mathbf{X} \in \mathbb{R}^{m \times k}$ and outputs a feature matrix $\mathbf{X}' \in \mathbb{R}^{m \times k}$ via convolving with m kernel matrices $\{\mathbf{C}^{(i)}\}_{i \in \{1, \dots, m\}} \in \mathbb{R}^{m \times 1 \times 1}$.

Note that each operator is followed by a nonlinear activation function (e.g. $\text{ReLU}(\cdot)$), excluding *Skip-connection*.

4.3 Search Strategy

The gradient-based search strategy is adopted in our method for computational efficiency. We first introduce architecture parameters for continuous relaxation on discrete search space. By doing so, the architecture can be optimized by the gradient descent method.

4.3.1 Continuous Relaxation. Let $\mathcal{O}$ be a set of candidate interactive operators, where each operator is denoted by a function $o(\cdot)$ to be applied to $\mathbf{X}^{(i)}$. Within a cell, the goal is to choose the best operator from $\mathcal{O}$ to connect each pair of nodes. Let a pair of nodes be (i, j) , where $i < j$, the key of continuous relaxation is to convert the combinatorial optimization problem to find the best weights of all operators between (i, j) after formulating the information propagated from i to j as a weighted sum over $|\mathcal{O}|$ operators, namely,

$$f^{(i,j)}(\mathbf{X}^{(i)}) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)} / \tau)}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)} / \tau)} o(\mathbf{X}^{(i)}) \quad (5)$$

where the weights of operators are parameterized by a vector $\alpha^{(i,j)}$ with dimension $|\mathcal{O}|$, and $\tau > 0$ is a parameter called temperature for controlling the importance discrepancy among all operators. Besides operation-level parameters α , we also introduce edge-level parameters β for choosing important node pairs in an interaction cell. And the Equation 4 can be rewrite as:

$$\mathbf{X}^{(j)} = \sum_{i < j} \frac{\exp(\beta^{(i,j)} / \tau)}{\sum_{i' < j} \exp(\beta^{(i',j)} / \tau)} \cdot f^{(i,j)}(\mathbf{X}^{(i)}). \quad (6)$$

As a result, we omit the *none* operation widely used in most gradient-based NAS methods [4, 13, 15, 19, 32], and normalize $\mathbf{X}^{(j)}$ for numerical stability. The task of architecture search then reduces to learning two set of continuous variables $\alpha = \{\alpha^{(i,j)}\}_{i < j}$, $\beta = \{\beta^{(i,j)}\}_{i < j}$.

4.3.2 Bi-level Optimization. Following [19], we use the Bi-level algorithm to alternatively learn the architecture parameters α, β and the weights w of all operators in the architecture. Specifically, the goal for searching powerful interactions is to find α^*, β^* that minimizes the validation loss $\mathcal{L}_{val}(w^*, \alpha, \beta)$, where the weights w^* are

obtained by minimizing the training loss $w^* = \arg\min_w \mathcal{L}_{train}(w, \alpha, \beta)$. This implies a bilevel optimization problem with α, β as the upper-level variable and w as the lower-level variable. Evaluating the architecture gradient exactly can be prohibitive due to the expensive inner optimization. Therefore, the one-step approximation [19] is used and the approximate architecture gradient yields

$$\nabla_{\alpha, \beta} \mathcal{L}_{val}(w', \alpha, \beta) - \xi \nabla_{\alpha, \beta, w}^2 \mathcal{L}_{train}(w, \alpha, \beta) \nabla_w \mathcal{L}_{val}(w', \alpha, \beta) \quad (7)$$

where $w' = w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha, \beta)$ denotes the weights for one-step forward model. We use the first-order approximation following [32], w.r.t $\xi = 0$, to reduce the computational complexity in the search process.

4.3.3 Batch Normalization. In Equation 5, the node j is the weighted sum of outputs of all operators in the operation space given i as the input. The contribution of a feature interaction $o(\mathbf{X}^{(i)})$ can be measured by $\text{softmax}(\alpha_o^{(i,j)})$. However, two problems may arise in fairness: (1) Our operation space contains both complicated and simple operators. In Bi-level optimization, the simple operators generally converge faster than complex operators, and then the stability of converged simple operators yield continuously better performance than unstable complex operators, which promotes larger weights $\text{softmax}(\alpha_o^{(i,j)})$ on simple operators. Since each edge reserves only one operator with the largest weight, more simple operators are prone to be selected. Therefore, the optimization on architecture parameters α, β should consider the comparison fairness of converged and under-converged operators. (2) Since the range of $o(\mathbf{X}^{(i)})$ varies among different operators and $o(\mathbf{X}^{(i)})$ is jointly optimized with $\text{softmax}(\alpha_o^{(i,j)})$, the coupling of their scale will lead to that $\text{softmax}(\alpha_o^{(i,j)})$ can hardly represent the relative importance of $o(\mathbf{X}^{(i)})$ [18].

It is well known that BN [11] can effectively solve the above problems by moving the output of each operator to a standard normal distribution, which is expressed as

$$\text{BN}(o(\mathbf{X})) = \frac{o(\mathbf{X}) - \mu_{\mathcal{B}}(o(\mathbf{X}))}{\sqrt{\sigma_{\mathcal{B}}^2(o(\mathbf{X})) + \epsilon}} \quad (8)$$

where $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ are the mean, standard deviation vector of $o(\mathbf{X})$ on the field dimension in mini-batch $\mathcal{B}$ and ϵ is a constant for numerical stability. Note that there is no need to optimize to distribution after normalization. Therefore, we set the scale and shift parameters in BN to be 1 and 0, respectively.

4.4 Performance Evaluation

The objective of our method is to find powerful interactions that can achieve high predictive performance on a target data distribution. We obtain architecture parameters with the best performance on the validation set during the search process, and then convert the continuous architecture encoding to a discrete architecture. The discrete architecture will be trained from scratch. However, a well-known problem in weight-sharing NAS methods is the performance gap [31] between the continuous architecture encoding and the derived discrete architecture, which is called discrete loss. In this section, we first introduce the discretization technique, and then illustrate two optional training techniques for reducing the performance gap.

Table 1: Statistics of the datasets.

Dataset	#instance	#fields	#features
Criteo	45,840,617	39	2,086,936
Avazu	40,428,967	22	1,544,250
Movielens	2,006,859	3	90,445
Frappe	288,609	10	5382
S1-S5	~25,000,000*5	28*5	~600,000*5

4.4.1 Discretization. For each node in the converged computational graph, we retain the top- k strongest operators (from distinct nodes) among all candidate operators collected from all previous nodes. The strength of an operator is defined as $\frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})}$. $\frac{\exp(\beta^{(i,j)})}{\sum_{k < j} \exp(\beta^{(k,j)})}$. We use $k = 2$ for interaction cell and ensemble cell following existing works on CV domain [19, 32].

4.4.2 Reducing Discrete Loss. Two training techniques can be optionally utilized to reduce the performance gap in our method.

(1) *Temperature Anneal.* [19] alleviate the performance gap via annealing the temperature parameter in softmax. By doing so, in each node pair, all operators have nearly the same strength defined above in order to train the weights w in the early search stage. As the temperature decreases, operators and edges begin to compete with each other to increase the discrepancy. We anneal the softmax temperature on every training epoch as follow formulation:

$$\tau = \frac{1 + \log(T)}{1 + \log(t)} \quad (9)$$

where T is the number of training epochs, and $t \in [1, \dots, T]$ is an index of the current epoch.

(2) *Noisy Skip-connection.* [4] proved that *Skip-connection* usually overperform other operators with softmax, resulting in a degenerate model where the skip connections contribute less. After injecting noise into the *Skip-connection* operator, the bias towards *Skip-connection* will be weakened. In this paper, we inject the Gaussian noise $\tilde{\mathbf{X}} \sim \mathcal{N}(\mu, \sigma)$ into our *Skip-connection* operator, i.e., $o_{\text{skip}}(\mathbf{X}) = \mathbf{X} + \tilde{\mathbf{X}}$, where $\mu = 0$ and $\sigma = \lambda \cdot \text{std}(\mathbf{X})$. Note that λ is a small positive coefficient.

5 Experiments and Results

To comprehensively evaluate our method, we design plenty of experiments to answer the following research questions:

RQ1: How do the architectures identified via AutoPI perform compared with the SOTA hand-crafted models and available NAS algorithms on datasets with different scales?

RQ2: How efficient is AutoPI compared with the SOTA hand-crafted models and available NAS algorithms?

RQ3: How effective are training techniques in alleviating the performance gap between the networks before and after discretization?

RQ4: How to understand the computational graph and operation space in our search space?

5.1 Experiment setup

Datasets. We conduct experiments on four public datasets with different statistical characteristics and five private commercial datasets generated from different E-commerce scenarios. The setting of public datasets follows previous works [3, 8, 14], including Criteo², Avazu³, Movielens⁴ and Frappe⁵. We randomly split the instances by 8:1:1 for training, validation and testing, respectively. The commercial datasets contain users' browsing and click records generated from real-world traffic logs on an E-commerce website. It is collected from five different scenes (S1-S5) in a discount activity. Each record has its corresponding user, item, context features, and a label indicating a click or not. Data of 3 days are used for training, and the following two days are used for validation and testing, respectively. The statistics of all datasets is shown in Table 1.

Evaluation metrics. We adopt two metrics for performance evaluation: AUC (Area Under the ROC curve) and Logloss (cross-entropy). Due to a large number of datasets, Δ_{AUC} and $\Delta_{Logloss}$ are also calculated to indicate averaged performance gain compared to a given benchmark over a group of datasets, which help to reflect the generalization ability of the algorithms. Note that an increase in AUC or decrease in Logloss at .001-level is known to be a significant improvement for the CTR prediction task [2, 3, 27].

Comparison methods. We compare **AutoPI-D** (AutoPI with Bi-level optimization in DARTS) with five classes of the existing approaches: (i) first-order approaches that is a weighted sum of raw features, including **LR**; (ii) FM-based methods that consider second-order cross features, including **FM** [25] and **AFM** [30]; (iii) advanced approaches that model higher-order feature interactions, including **CrossNet** [28], **HOEM** [1], **NFM** [8], **PNN** [22], **CIN** [14], **AutoInt** [27] and **AFN** [3]; (iv) ensemble models that are integrated with a tower of DNN, including **Deep&Cross** [28], **Wide&Deep** [2], **DeepFM** [7], **xDeepFM** [14], **AutoInt+** [27] and **AFN+** [3]. (v) NAS algorithms that automatically search the effective architecture for a target dataset, including **AutoRec** [29] and **AutoPI-R** (AutoPI with random search strategy). Note that no available codes have been found for other NAS methods, such as AutoCTR. We select the open-sourced **AutoRec** as our baseline. And we denote **AutoRec-R**, **AutoRec-G**, **AutoRec-B** as the random, greedy, Bayesian version of **AutoRec**, respectively.

Implementation details. We implement AutoPI using Pytorch⁶. Both interaction and ensemble cells consist of $N = 4$ intermediate nodes. We set the embedding size as $k = 10$ and adopt temperature anneal during the search process. Half of the training set is used for optimizing weights w in operators, and the other half is for architecture parameters α, β . The maximum number of training epochs for Criteo, Avazu, Commerical Dataset, Frappe, Movielens are 5, 5, 5, 50 and 100, respectively, and the corresponding batch size is 4096. We use momentum SGD to optimize the weights w and use Adam as the optimizer for architecture parameters α, β , with the configuration of [32]. To avoid overfitting, we perform early-stopping according to the Logloss on the validation set. Furthermore, we implement hand-crafted models according to the details of [3], where

²<http://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/>

³<https://www.kaggle.com/c/avazu-ctr-prediction>

⁴<https://grouplens.org/datasets/movielens/>

⁵<http://baltrunas.info/research-menu/frappe>

⁶The code is available here: <https://github.com/thu-media/AutoPI>.

Table 2: Effectiveness Comparison of Different Algorithms on Public Benchmarks.

Model Class	Model	Criteo		Avazu		Movielens		Frappe		$\Delta_{AUC} \uparrow$	$\Delta_{Logloss} \downarrow$
		AUC	Logloss	AUC	Logloss	AUC	Logloss	AUC	Logloss		
First-Order	LR	0.7858	0.4636	0.7313	0.4065	0.9215	0.3080	0.9329	0.2860	-3.01%	+0.0535
Second-Order	FM	0.7933	0.4574	0.7496	0.3740	0.9388	0.2797	0.9641	0.2143	-1.16%	+0.0188
	AFM	0.7953	0.4554	0.7454	0.3766	0.9295	0.2836	0.9639	0.2294	-1.45%	+0.0237
High-Order	CrossNet	0.7915	0.4585	0.7498	0.3756	0.9323	0.2929	0.9393	0.2835	-1.98%	+0.0400
	HOFM	0.7960	0.4551	0.7516	0.3756	0.9410	0.3088	0.9709	0.2141	-0.81%	+0.0259
	NFM	0.7968	0.4537	0.7531	0.3761	0.9441	0.3004	0.9727	0.2079	-0.63%	+0.0220
	PNN	0.8026	0.4509	0.7526	0.3737	0.9469	0.2792	0.9735	0.2012	-0.41%	+0.0137
	CIN	0.8042	0.4472	0.7533	0.3756	0.9494	0.2600	0.9704	0.2342	-0.37%	+0.0170
	AutoInt	0.8062	0.4456	0.7465	0.3790	0.9418	0.2762	0.9701	0.2034	-0.69%	+0.0135
	AFN	0.8061	0.4458	0.7512	0.3731	0.9477	0.2753	0.9759	0.1784	-0.28%	+0.0056
Ensembled	Deep&Cross	0.8059	0.4463	0.7550	0.3721	0.9419	0.2791	0.9402	0.2808	-1.23%	+0.0320
	Wide&Deep	0.8062	0.4453	0.7529	0.3744	0.9381	0.3310	0.9728	0.2038	-0.55%	+0.0261
	DeepFM	0.8025	0.4501	0.7535	0.3742	0.9424	0.3131	0.9719	0.2108	-0.54%	+0.0245
	xDeepFM	0.8070	0.4443	0.7535	0.3737	0.9448	0.2717	0.9738	0.2098	-0.32%	+0.0123
	AutoInt+	0.8075	0.4438	0.7477	0.3776	0.9417	0.2764	0.9661	0.2102	-0.73%	+0.0145
	AFN+	0.8083	0.4437	0.7555	0.3718	0.9500	0.2585	0.9783	0.1762	-	-
NAS	AutoRec-R	0.8104	0.4412	0.7499	0.3746	0.9510	0.2573	0.9775	0.1688	-0.08%	-0.0021
	AutoRec-G	0.8100	0.4417	0.7467	0.3756	0.9496	0.2579	0.9771	0.1778	-0.22%	+0.0007
	AutoRec-B	0.8097	0.4418	0.7488	0.3766	0.9539	0.2443	0.9783	0.1610	-0.03%	-0.0066
	AutoPI-R	0.8105	0.4410	0.7598	0.3730	0.9555	0.2598	0.9820	0.1487	+0.39%	-0.0069
	AutoPI-D	0.8102	0.4420	0.7605	0.3691	0.9572	0.2660	0.9827	0.1426	+0.46%	-0.0076

we use the same neural network structure (i.e., 3-layers MLP, 400-400-400) for all approaches that involve DNN. We implement the NAS baselines by following [29]. For each empirical result, we run the experiments with different random seeds for three times and report the average value.

5.2 Overall Performance (RQ1)

We set the powerful hand-crafted model, AFN+, as the benchmark for calculating Δ_{AUC} and $\Delta_{Logloss}$ on all datasets.

Public Datasets. As illustrated in Table 1, the public datasets have distinct heterogeneous feature spaces, w.r.t, different number of instances, fields and features. Table 2 summarizes the performance of all methods on four public datasets.

We first present three important observations when comparing various hand-crafted models. First, the methods that utilize higher-order feature interactions generally outperform those based on lower-order cross features, which means higher-order feature interactions are informative. Second, AFN and AutoInt consistently outperform FMs and HOFMs on all public datasets, which verifies that learning adaptive-order cross features can bring better predictive performance than modeling fixed-order feature interactions. Third, the ensemble models have a significant performance improvement than individual models, which demonstrates that combining different types of interactions is more effective than a single one.

Moreover, the NAS baseline, AutoRec, produces the competitive performance comparing with the AFN+, and outperforms all other

hand-crafted models. This illustrates that using NAS can successfully find effective feature interactions and alleviate laborious and tedious architecture engineering. The search space in AutoRec supports the ensemble of at most 6 interactive blocks, where each block can be selected from one of the 5 types of configurable operators: MLP, FM, CrossNet, self-attention and element-wise interaction. Although the number of blocks and hyperparameters in the operators can be jointly optimized by different algorithms including the effective Bayesian method, the search space of AutoRec only incorporates limited efficient manual designs. AutoPI-R, exploring our search space by a simple search strategy, can search for more powerful interactions than AutoRec, which means our search space is more flexible and general than AutoRec. We further compare our method (AutoPI-D) with AutoPI-R. AutoPI-D yields a better or competing performance than AutoPI-R over all public datasets. This demonstrates that the improved gradient-based search strategy is more effective than the random method.

Real Scenarios. As illustrated in Table 1, the commercial datasets have close statistical properties but diverse semantics. We collect real-world traffic logs of five different scenes (S1-S5) from an online retailer platform. Each scene is a specific page on the E-commerce website to satisfy all kinds of shopping needs of users. The five scenes we collected contain food, makeup, etc. Each scene has different UI pages, target user groups, and data distribution. Due to page limitations, we select three representative hand-crafted models for real-scenario evaluation, including xDeepFM, AutoInt+ and AFN+.

Table 3: Effectiveness Comparison of Different Algorithms on Real Scenarios.

Model	S1		S2		S3		S4		S5		Δ_{AUC}	$\Delta_{Logloss}$
	AUC	Logloss	AUC	Logloss	AUC	Logloss	AUC	Logloss	AUC	Logloss	$\uparrow$	$\downarrow$
xDeepFM	0.7973	0.1678	0.7179	0.3319	0.7481	0.1462	0.7457	0.1449	0.8206	0.2970	-0.67%	+0.0030
AutoInt+	0.8033	0.1639	0.7173	0.3319	0.7606	0.1396	0.7400	0.1459	0.8214	0.2949	-0.41%	+0.0006
AFN+	0.8084	0.1618	0.7263	0.3291	0.7610	0.1414	0.7464	0.1448	0.8208	0.2954	-	-
AutoRec-R	0.8112	0.1634	0.7230	0.3300	0.7637	0.1413	0.7502	0.1445	0.8248	0.2930	+0.20%	-0.0001
AutoRec-G	0.8082	0.1637	0.7214	0.3307	0.7638	0.1414	0.7509	0.1445	0.8248	0.2929	+0.12%	+0.0001
AutoRec-B	0.8072	0.1636	0.7235	0.3298	0.7630	0.1407	0.7512	0.1442	0.8250	0.2930	+0.14%	-0.0001
AutoPI-R	0.8128	0.1647	0.7308	0.3302	0.7656	0.1392	0.7545	0.1434	0.8238	0.2948	+0.50%	-0.0001
AutoPI-D	0.8142	0.1638	0.7316	0.3287	0.7678	0.1387	0.7590	0.1431	0.8256	0.2970	+0.70%	-0.0002

Table 4: Comparison Of Architecture Efficiency Among SOTA Methods (Excluding Embedding Layers Of The Same Size).

Model	# Params (Million)				Flops (Million)				Search cost (GPU Hours)			
	Criteo	Avazu	Mvl	Frappe	Criteo	Avazu	Mvl	Frappe	Criteo	Avazu	Mvl	Frappe
xDeepFM	7.33	4.13	0.82	2.01	31.24	11.54	1.16	3.55	-	-	-	-
AutoInt+	0.49	0.42	0.34	0.37	27.79	6.75	0.89	1.86	-	-	-	-
AFN+	6.86	5.56	3.86	3.09	14.18	11.32	7.73	6.22	-	-	-	-
AutoRec-R	1.47	0.01	0.04	0.03	2.82	0.01	0.07	0.05	47.50	22.20	1.70	1.30
AutoRec-G	0.58	0.01	0.30	0.04	1.17	0.01	0.61	0.07	45.70	18.80	2.10	1.60
AutoRec-B	0.85	0.01	1.21	0.02	1.70	0.02	2.50	0.05	41.10	20.20	20.50	2.40
AutoPI-R	1.60	0.14	0.01	0.02	3.46	0.32	0.02	0.07	50.50	19.40	2.00	1.20
AutoPI-D	1.18	0.10	0.01	0.05	2.51	0.25	0.02	0.12	4.20	2.20	0.40	0.10

The comparison on the commercial datasets of these algorithms is shown in Table 3. From the results, we observe that the NAS methods outperform all hand-crafted models over all commercial datasets. This illustrates that the NAS methods are more effective than hand-crafted models on extensive real scenarios. Most importantly, our method increases the average AUC Δ_{AUC} by 0.70%, which is large than AutoRec by 0.50%, which further verify the generalization of our proposed AutoPI.

5.3 Efficiency Analysis (RQ2)

In real-world scenarios, both the effectiveness and efficiency of a method are critical. Many SOTA methods can achieve good accuracy but fail to be deployed in commercial recommender systems due to their computational inefficiency, especially the additional time-consuming search process in NAS methods. In this section, we study the efficiency of AutoPI on public datasets using one NVIDIA P100 GPU. We present the search time of the NAS algorithms and the complexity (parameters size and Flops) of the identified best architectures in Table 4. Benefiting from the gradient-based search strategy and weight-sharing mechanism, our method (AutoPI-D) requires $\sim \times 10$ less search time than other NAS algorithms (AutoRec and AutoPI-R). In addition, the complexity of the architectures identified via our method are close to AutoRec, but much lower

than hand-crafted models⁷. Therefore, AutoPI can achieve better performance efficiently, which makes it easier to be deployed in industrial applications.

Furthermore, we present the ablation study of operators with large parameters in Table 5 to reduce the complexity of our architectures. In our operation space, MLP and FM have a large number of parameters, and we run four experiments on Avazu and Frappe with different configurations of operation space: (1) omitting MLP, (2) omitting FM, (3) omitting MLP and FM, (4) non-omitting. As illustrated in Table 5, the flexibility of AutoPI is validated. Therefore, by omitting operators with large parameters, the trade-off between performance and computational cost can be achieved.

5.4 Training Techniques Study (RQ3)

In this section, we evaluate the effectiveness of two training techniques on Avazu and Frappe by ablation study and the results are illustrated in Table 6. In detail, we run four experiments with different training techniques: (1) noisy skip-connection (Noisy): injecting the noise ($\lambda = 0.03$) into the skip-connection with the fixed softmax temperature ($\tau = 1$), (2) temperature anneal (Anneal): annealing the softmax temperature with the original skip-connection

⁷Although AutoInt+ has lower parameters than our method, the Flops of AutoInt+ is much larger than our method because of the key-value attention mechanism.

Table 5: Ablation study of interactive operators.

Omitted Operation	Avazu				Frappe				Δ_{AUC}	$\Delta_{Logloss}$
	AUC	Logloss	Params (M)	Flops (M)	AUC	Logloss	Params (M)	Flops (M)	$\uparrow$	$\downarrow$
MLP	0.7580	0.3695	0.05	0.27	0.9790	0.1777	0.02	0.07	-0.62%	+0.0166
FM	0.7587	0.3695	0.10	0.24	0.9820	0.1399	0.05	0.11	-0.13%	-0.0024
MLP&FM	0.7594	0.3708	0.01	0.17	0.9685	0.2146	0.01	0.03	-0.77%	+0.0357
None	0.7605	0.3715	0.10	0.25	0.9827	0.1426	0.05	0.12	-	-

Table 6: Effectiveness Comparison of Training Techniques.

Techniques	Avazu		Frappe		Δ_{AUC}	$\Delta_{Logloss}$
	AUC	Logloss	AUC	Logloss	$\uparrow$	$\downarrow$
Origin	0.7606	0.3688	0.9794	0.1588	-	-
Noisy	0.7610	0.3682	0.9809	0.1522	+0.10%	-0.0036
Anneal	0.7605	0.3715	0.9827	0.1553	+0.17%	-0.0004
Noisy+Anneal	0.7605	0.3684	0.9804	0.1556	+0.09%	-0.0018

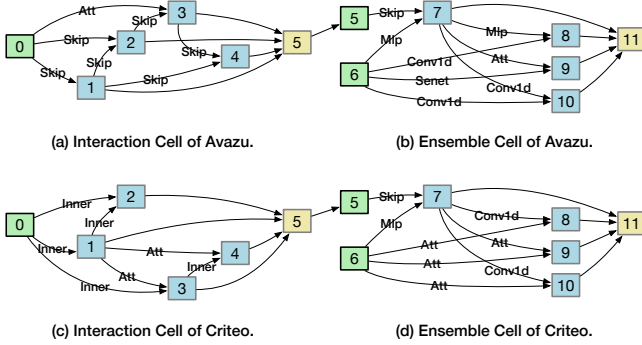


Figure 4: Two architectures found by AutoPI.

during the search process, (3) noisy skip-connection and temperature anneal (Noisy+Anneal): using both training techniques in the search process, (4) without noisy skip-connection and temperature anneal (Origin): searching the architecture without both training techniques. We use AUC and Logloss as metrics to evaluate the effectiveness of two types of training techniques. Besides, we set the *Origin* as the benchmark for calculating Δ_{AUC} and $\Delta_{Logloss}$. As illustrated in Table 6, *Noisy* reduces the averaged Logloss $\Delta_{Logloss}$ by a large margin (0.0036), while *Anneal* increases the averaged AUC Δ_{AUC} by 0.0017. This demonstrates that both injecting noise into skip-connection and annealing the softmax temperature can reduce the discrete loss over datasets with different scales. The former can reduce the averaged Logloss significantly, while the latter can increase the averaged AUC remarkably. Note that the combination of these two techniques (*Noisy+Anneal*) does not show superior performance than individual ones. We will conduct further investigation on this phenomenon in the future.

5.5 Case Study (RQ4)

We run experiments of Avazu and Criteo multiple times with different random seeds. Since the obtained architectures identified by AutoPI are similar in each task, two representative architectures

for Avazu and Criteo respectively are shown in Figure 4 for more detailed analysis of our search space. We analyse the architecture searched via AutoPI from three perspectives: (1) The order of interactions. The interaction cell in Avazu consists of a self-attention (Att) and seven skip-connections, building a second-order feature interaction, of which order is lower than Criteo. This demonstrates the Avazu relies more on lower-order cross features which is consistent with the observation in [3]. (2) The type of operators. In our operation space, there exist various operators with different complexity. We observe that compared to the architecture searched for Avazu, more complex operators are selected by AutoPI for Criteo, which means our method can automatically generate larger networks for more complicated datasets. (3) The integrated towers. An interesting observation is that the ensemble cell of Criteo has two repeated connections (i.e., $6, 7 \rightarrow 8$ and $6, 7 \rightarrow 10$), which can be simplified to a three-tower ensemble (i.e., integrating intermediate nodes 7, 8, 9). This illustrates that Criteo only needs a small number of towers to be integrated because of its higher-order interactions. On the contrary, Avazu needs more towers to be integrated because it prefers lower-order interactions.

6 Discussion

In this paper, we propose a general method AutoPI to automatically search for both effective and efficient architectures for CTR prediction in different scenarios with different scales. Extensive experiments on a diverse suite of benchmark datasets demonstrate the model searched by our method outperforms both the hand-crafted models and other NAS methods in terms of AUC and Logloss. Coupling with the gradient-based search strategy, AutoPI is more computationally efficient than all available NAS methods.

To reduce the performance gap between the continuous architecture encoding and discrete architecture, both temperature anneal and noisy skip-connection are verified to be effective. Finally, in the case study, some interesting observations regarding to the architecture properties for large-scale and small-scale datasets are summarized, which can provide insights for researchers on how to design better CTR models.

Acknowledgments

This work was supported by NSFC under Grant No. 61936011 and No. 615210002, Beijing Key Lab of Networked Multimedia. Research reported in this publication was also supported by the National Library of Medicine of the National Institutes of Health, USA, under award number R01LM013151. We thank Tianchi Huang and Xin Yao for fruitful discussions. Finally, life is fine.

References

- [1] Mathieu Blondel, Akinori Fujino, Naonori Ueda, and Masakazu Ishihata. 2016. Higher-order factorization machines. *arXiv preprint arXiv:1607.07195* (2016).
- [2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishikesh Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ipsir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.
- [3] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3609–3616.
- [4] Xiangxiang Chu, Bo Zhang, and Xudong Li. 2020. Noisy differentiable architecture search. *arXiv preprint arXiv:2005.03566* (2020).
- [5] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. 2020. Fair darts: Eliminating unfair advantages in differentiable architecture search. In *European Conference on Computer Vision*. Springer, 465–480.
- [6] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. 2019. Neural architecture search: A survey. *J. Mach. Learn. Res.* 20, 55 (2019), 1–21.
- [7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
- [8] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.
- [9] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feed-forward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.
- [10] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 169–177.
- [11] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.
- [12] Farhan Khawar, Xu Hang, Ruiming Tang, Bin Liu, Zhenguo Li, and Xiuqiang He. 2020. AutoFeature: Searching for Feature Interactions and Their Architectures for Click-through Rate Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 625–634.
- [13] Yonggang Li, Yafeng Zhou, Yongtao Wang, and Zhi Tang. 2020. PD-DARTS: Progressive Discretization Differentiable Architecture Search. In *International Conference on Pattern Recognition and Artificial Intelligence*. Springer, 306–311.
- [14] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.
- [15] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. 2019. Darts+: Improved differentiable architecture search with early stopping. *arXiv preprint arXiv:1909.06035* (2019).
- [16] Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. 2019. Feature generation by convolutional neural network for click-through rate prediction. In *The World Wide Web Conference*. 1119–1129.
- [17] Bin Liu, Niannan Xue, Huifeng Guo, Ruiming Tang, Stefanos Zafeiriou, Xiuqiang He, and Zhenguo Li. 2020. AutoGroup: Automatic feature grouping for modelling explicit high-order feature interactions in CTR prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 199–208.
- [18] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2636–2645.
- [19] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).
- [20] Yuanfei Luo, Mengshuo Wang, Hao Zhou, Quanming Yao, Wei-Wei Tu, Yuqiang Chen, Wenyuan Dai, and Qiang Yang. 2019. Autocross: Automatic feature crossing for tabular data in real-world applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1936–1945.
- [21] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. 2018. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*. PMLR, 4095–4104.
- [22] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems (TOIS)* 37, 1 (2018), 1–35.
- [23] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. 2019. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 33. 4780–4789.
- [24] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. 2020. A comprehensive survey of neural architecture search: Challenges and solutions. *arXiv preprint arXiv:2006.02903* (2020).
- [25] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 1–22.
- [26] Qingquan Song, Dehua Cheng, Hanning Zhou, Jiyan Yang, Yuandong Tian, and Xia Hu. 2020. Towards automated neural interaction discovery for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 945–955.
- [27] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.
- [28] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD’17*. 1–7.
- [29] Ting-Hsiang Wang, Xia Hu, Haifeng Jin, Qingquan Song, Xiaotian Han, and Zirui Liu. 2020. AutoRec: An Automated Recommender System. In *Fourteenth ACM Conference on Recommender Systems*. 582–584.
- [30] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).
- [31] Lingxi Xie, Xin Chen, Kaifeng Bi, Longhui Wei, Yuhui Xu, Zhengsu Chen, Lanfei Wang, An Xiao, Jianlong Chang, Xiaopeng Zhang, et al. 2020. Weight-Sharing Neural Architecture Search: A Battle to Shrink the Optimization Gap. *arXiv preprint arXiv:2008.01475* (2020).
- [32] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. 2019. Pc-darts: Partial channel connections for memory-efficient differentiable architecture search. *arXiv preprint arXiv:1907.05737* (2019).

Path-based Deep Network for Candidate Item Matching in Recommenders

Houyi Li¹, Zhihong Chen¹, Chenliang Li^{3†}, Rong Xiao¹, Hongbo Deng^{1†}, Peng Zhang¹,
Yongchao Liu², Haihong Tang¹

¹Alibaba Group, Hangzhou, China, ²Ant Group, Hangzhou, China

³School of Cyber Science and Engineering, Wuhan University, Wuhan, China

¹{houyi.lhy, jhon.czh, xiaorong.xr, dnb167148, zhangpeng04, piaoxue}@alibaba-inc.com

³cllee@whu.edu.cn, ²yongchao.ly@antgroup.com

ABSTRACT

The large-scale recommender system mainly consists of two stages: matching and ranking. The matching stage (also known as the retrieval step) identifies a small fraction of relevant items from billion-scale item corpus in low latency and computational cost. Item-to-item collaborative filtering (item-based CF) and embedding-based retrieval (EBR) have been long used in the industrial matching stage owing to its efficiency. However, item-based CF is hard to meet personalization, while EBR has difficulty in satisfying diversity. In this paper, we propose a novel matching architecture, Path-based Deep Network (named PDN), through incorporating both personalization and diversity to enhance matching performance. Specifically, PDN is comprised of two modules: *Trigger Net* and *Similarity Net*. PDN utilizes Trigger Net to capture the user's interest in each of his/her interacted item. Similarity Net is devised to evaluate the similarity between each interacted item and the target item based on these items' profile and CF information. The final relevance between the user and the target item is calculated by explicitly considering user's diverse interests, *i.e.*, aggregating the relevance weights of the related two-hop paths (one hop of a path corresponds to user-item interaction and the other to item-item relevance). Furthermore, we describe the architecture design of the proposed PDN in a leading real-world E-Commerce service (Mobile Taobao App). Based on offline evaluations and online A/B test, we show that PDN outperforms the existing solutions for the same task. The online results also demonstrate that PDN can retrieve more personalized and more diverse items to significantly improve user engagement. Currently, PDN system has been successfully deployed at Mobile Taobao App and handling major online traffic.

CCS CONCEPTS

• Information systems → Recommender systems.

† Chenliang Li and Hongbo Deng are the corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8037-9/21/07...\$15.00

<https://doi.org/10.1145/3404835.3462878>

KEYWORDS

Deep Learning, Recommendation Systems

ACM Reference Format:

Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, Haihong Tang. 2021. Path-based Deep Network for Candidate Item Matching in Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3404835.3462878>

1 INTRODUCTION

Recommender systems are important in customer-oriented E-Commerce platforms (e.g., Taobao and Amazon). The purpose is to connect users to their preferred items and produce more profits. Due to the tremendous number of items available in the E-Commerce platforms, many industrial systems [6, 11] are devised with a matching stage and a ranking stage. Specifically, a matching stage is expected to retrieve a small fraction of relevant items in low latency and computational cost, and a ranking stage aims to refine the ranking of these relevant items in terms of the user's interest with more complex models. In this work, we focus on the matching stage since it is the fundamental part and also the bottleneck of the system.

Item-to-item based collaborative filtering, as known as item-based CF, is an information retrieval solution for item matching. It estimates the relevance between two items based on their co-occurrence patterns. Several outstanding merits make item-based CF a natural choice for the matching stage in industry [7, 24, 28]: (1) Since only the items previously interacted by the user are used for retrieval, an item-based CF could well match the efficiency requirement for many online services; (2) Furthermore, a user's interest could be very diverse. By taking each interacted item into consideration, the user's interest can be well covered to its maximum; (3) At last, the relevance between items is mainly derived based on massive user behaviors. These signals are effective to identify the relevant items w.r.t the interacted ones.

However, this kinds of methods also have some limitations. The traditional inverted indexes are hard to meet subtle personalization needs [33]. By considering only item co-occurrence patterns, item-based CF is inferior to accommodate the user's unique characteristics, *e.g.*, gender, consumption capacity and others. Similarly, given the volume of items available in a E-Commerce platform keeps growing, without considering auxiliary information, item-based CF could suffer a lot from the data sparsity problem [22].

To overcome the above limitations, embedding-based retrieval (EBR), especially the deep learning networks with a two-tower architecture, has drawn growing interests recently [17, 31, 32]. Briefly speaking, EBR aims to represent each user and item by embedding their profile respectively. In this sense, the matching process is transferred to perform nearest neighbor (NN) search in the embedding space. Although the representation learning could alleviate the data sparsity problem to some extent, these methods also have some limitations. The two-tower architecture is not easy to explicitly integrate the co-occurrence information between items. Moreover, a user is often represented as a single embedding vector, which is insufficient to encode the diversity of the user's interest [6, 18].

A normal user would interact with hundreds of items that belong to different categories every month, indicating the diversity of user interests. Actually, in real industrial systems, to simultaneously capture the diversity of user interests and ensure personalization, typically there are multiple strategies, e.g., various kinds of inverted indexes based on collaborative filtering models and EBR strategies with different network structures. These models are deployed in parallel for the matching task. Note that the candidates are usually generated with the relevance scores in different scales. It is not straightforward to fuse these incomparable values for promising performance. We argue that this multi-strategy solution may be suboptimal due to the high maintenance cost of these strategies and lack of tailored joint optimization.

Deep Interest Network (DIN) [35] introduces the similarity between interacted and target items through the target attention for a better recommendation. However, this attention mechanism is simply used to fuse user interaction sequences, which ignores the user's interest in each interacted item. Also, DIN is difficult to be applied for the matching stage since it requires recalculating the user representation for each target item. Inspired by DIN, in this work, we propose a novel matching architecture called Path-based Deep Network (PDN), which decouples the target item based attention from user representation learning by building a marriage between item-based CF and EBR. In PDN, we use the thought of representation learning of EBR (*i.e.*, user profile, interacted item sequence and item profile) and item-based CF (*i.e.*, item co-occurrence) to accommodate both user personalization and diverse interest modeling for better performance. Specifically, PDN consists of two main subnetworks: Trigger Net and Similarity Net. Trigger Net (*TrigNet*) is introduced to encode the user's interest by considering each interacted item as a trigger¹. That is, the generated user representation has a variable dimension such that each dimension describes the user's interest on the interacted item. Analogous to item-based CF, Similarity Net (*SimNet*) generates an item representation and each dimension describes the similarity between an interacted item and the target item. Note that, the dimensions of user representation and item representation extracted by EBR are constant, while the dimensions of the user and item representations extracted by PDN are variable which are equal to the number of the user's triggers. As shown in Figure 1, by connecting the user with the target item through her interacted items, we can form a series of 2-hop paths. Based on these 2-hop paths, PDN aggregate the relevance between

the user and the target item by explicitly considering a user's diverse interests for better performance. Another merit is that the whole model is trained with an end-to-end fashion. Therefore, the relevance scores can be compared with each other in a uniform way.

It is worthwhile to highlight that the proposed PDN bears the advantages of both item-based CF and EBR for efficient online processing. On one hand, empowered by the feature embedding, we can utilize *TrigNet* to extract top- m most important triggers w.r.t. the user interests to satisfy the real-time requirement. On the other hand, *SimNet* works independently from *TrigNet*. We can apply parallel computing to support offline index construction with item-to-item relevance calculated by *SimNet* efficiently. In summary, the main contributions of this paper are as follows:

- **A novel matching model.** We propose Path-based deep network by incorporating advantages of item-based CF and EBR. PDN integrates all of profile information for user attention and co-occurrence patterns between items for target attention in the form of 2-hop path aggregation. Both the user personalization and the interest diversity are accommodated for better item matching.
- **Efficient online retrieval.** We construct an industrial-scale online matching system based on PDN. In particular, we describe how to leverage PDN for item retrieval in low latency and computational cost.
- **Offline and online experiments.** Extensive offline experiments on several real-world datasets demonstrates that the proposed PDN achieves much better performance than the existing alternatives. Besides, we evaluate PDN on the recommender system of Taobao with A/B test over a two-week period. The results suggest that a large performance gain is obtained on almost all metrics.

2 RELATED WORK

CF-based methods are successful in building recommender systems at the matching stage [29]. Among them, item-based CF [23, 26], which calculates the similarity matrix of items in advance, and recommend items similar to the user's clicked ones, has been widely employed in industrial settings due to its interpretability and efficiency. Early works utilize statistical measures such as cosine similarity and Pearson coefficient to estimate item similarities. In recent years, several approaches attempt to learn item similarities by optimizing a recommendation-aware objective function. Ning *et al.* [25] propose SLIM for item relevance learning by minimizing the loss between the original user-item interaction matrix and the reconstructed one. He *et al.* [14] propose NAIS with an attention mechanism to distinguish the different importance of historical items in a user profile, which shares a similar idea with DIN [35]. However, attention mechanism based methods are only applicable to the ranking stage due to the complexity of computation.

With the success of EBR [3], two-tower architectures based on deep neural networks have been widely adopted in industrial recommender systems to capture personalized information of a user by leveraging rich content features [17, 31, 32]. Note that, in the matching stage, to process billions or trillions of items in low latency and computational cost, the two towers can not interact with

¹The terms *interacted item*, *trigger*, *trigger item* are exchangeable since they refer to the same meaning in this paper.

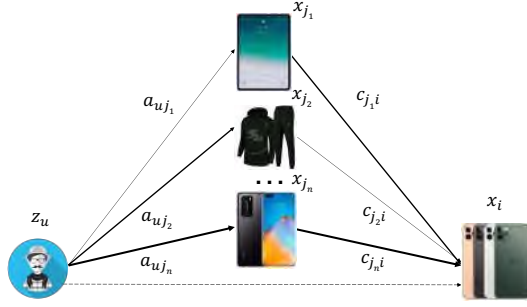


Figure 1: Modeling the preference of user u to target item i with a 2-hop graph where j_1 to j_n represent interacted items. The thickness of edge indicates the relevance.

each other to ensure the parallel feature extraction. Particularly, the DSSM-based model [18] learns the relevance based on the inner product between user features and item features. To extract more discriminative user features, Youtube DNN [6] extends user features by average pooling user behavior, while BST [4] utilizes the powerful Transformer model to capture the sequential signals underlying users' behavior sequences.

Different from the above methods, the PDN we propose combines the advantages of item-based CF and EBF based on deep neural networks, which constructs user-item subnetwork to ensure personalization similar to EMB, and constructs item-item subnetwork to capture multiple user interests similar to item-based CF.

3 PRELIMINARIES

Figure 1 summarizes the recommendation problem in the form of 2-hop paths between the user and the target item. Here, z_u represents the user information, e.g., user id, age, gender, click times, purchase times of each category, for user u ; x_i represents the information of target item, e.g., item id, brand id, category id, monthly sales of target item i ; $\{x_{j_k}\}_{k=1}^n$ represent the auxiliary information of items $\{j_k\}_{k=1}^n$ interacted by u ; $\{a_{uj_k}\}_{k=1}^n$ represent the behavior information from user u on these interacted items, e.g., stay time, purchase times; and $\{c_{j_k i}\}_{k=1}^n$ represent the relevance information between interacted items and target item, which is obtained from an item-based CF algorithm or statistical correlation measure based on item co-occurrence patterns.

As shown in Figure 1, there is a direct link between the user and the target item (shown in dashed line), which indicates the user's intuitive interest on the target item. Also, we can further form n 2-hop paths by bridging through the interacted items. The first hop represents the user's interest in the interacted items and the second hop represents the relevance between the interacted items and the target item. Hence, the item matching with the above available information for recommendation can be formulated as:

$$\hat{y}_{ui} = f(z_u, x_i, \{x_j\}, \{a_{uj}\}, \{c_{ji}\}) \text{ with } j \in N(u) \quad (1)$$

where f is defined as the recommendation algorithm, $\hat{y}_{ui}$ is defined as the relevance score between user u and target item i , and $N(u)$ is defined as the items interacted by u .

Most of the existing work for recommender systems, including item-based CF and EBR, can be regarded as a special case of Eq. 1. For example, the regression form of item-based CF [27] can be

formulated as:

$$\hat{y}_{ui} = f(\{a_{uj}\}, \{c_{ji}\}) = \sum_{j \in N(u)} f_r(a_{uj})c_{ji} \quad (2)$$

where $f_r : \mathcal{R}^m \rightarrow \mathcal{R}^1$ is a weighting function to capture user's interest for each trigger, $c_{ji} \in \mathcal{R}^1$ represents the relevance between interacted item j and target item i based on item co-occurrence information. Hence, the method can be seen as the sum of weights of all 2-hop paths based on $\{a_{uj}\}$ and $\{c_{ji}\}$, and each path weight can be calculated as $f_r(a_{uj})c_{ji}$.

Besides, the methods based on EBR are also a special case of Eq. 1. For example, the matrix factorization (MF) [20] can be formulated as:

$$\hat{y}_{ui} = f(z_u, x_i, \{x_j\}) = q_i(p_u + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} q_j)^T \quad (3)$$

where the q_i , p_u , q_j represent the embedding vector for target item information x_i , user information z_u and interacted item $\{x_j\}$, respectively. MF can be regarded as the sum of the weights of $n+1$ paths. Specifically, the weight of direct path is $q_i p_u$, and the weight of each 2-hop path is $1/\sqrt{|N(u)|} \cdot q_i q_j$. While YoutubeDNN [6] utilizes deep neural networks as a generalization of matrix factorization, which can be formulated as:

$$\hat{y}_{ui} = f(z_u, x_i, \{x_j\}) = q_i \left(MLP(p_u, \frac{1}{|N(u)|} \sum_{j \in N(u)} q_j) \right)^T \quad (4)$$

Here MLP refers to the multilayer perception. DIN can also be formulated as [35]:

$$\begin{aligned} \hat{y}_{ui} &= f(z_u, x_i, \{x_j\}, \{a_{uj}\}) \\ &= MLP(p_u, q_i, \sum_{j \in N(u)} (MLP(q_j, a_{uj}, q_i) \odot q_j)) \end{aligned} \quad (5)$$

where $\odot$ represents the element-wise product. Note that DIN [35] can be considered as a representation of a 2-hop path ($u \rightarrow j \rightarrow i$) in terms of relevance between the target item and each trigger. However, it requires re-calculation of path representation for each target item, making it only applicable for the ranking stage.

To ensure the efficiency of retrieval, item-based CF builds inverted index while EBR applies k-nearest neighbors (KNN) search for online serving. However, because both model architectures are constrained by efficiency, they cannot make use of all available information in Figure 1, resulting in suboptimal performance. For example, item-based CF lacks user and item profiles, while EBR lacks the explicit co-occurrence information between items. Hence, in this paper, we propose a novel architecture, named PDN, to support both personalized and diversity retrieval with low latency.

4 METHOD

In this section, we present the design of the Path-based Deep Network (PDN) for the matching stage of recommender systems. We first introduce the overall architecture of PDN, and then we elaborate on each module of PDN including Embedding Layer, Trigger Net (*TrigNet*), Similarity Net (*SimNet*), Direct Net, and Bias Net.

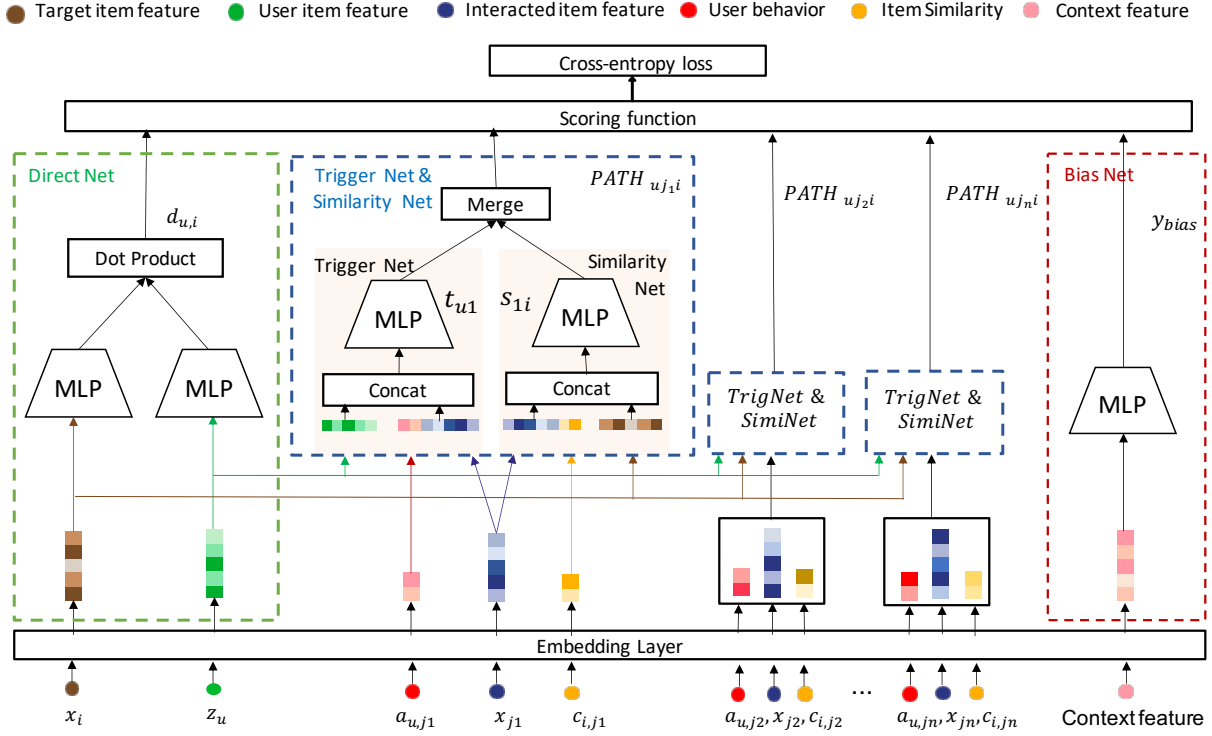


Figure 2: The network structure of PDN. Direct net is used to obtain the weight of the direct path to capture the user’s intuitive interest in the target item, and *TrigNet* and *SimNet* obtain the first-hop weight and the second-hop weight of each 2-hop path respectively to capture the fine-grained user’s personalized diversity interests, bias net is used to capture various types of selection bias for further unbiased serving. (Best viewed in color)

4.1 Overview of PDN

According to Eq 1, the basic workflow of PDN can be formulated as follows:

$$\hat{y}_{ui} = AGG\left(f_d(z_u, x_i), \{PATH_{uji}\}\right) \text{ with } j \in N(u) \quad (6a)$$

$$PATH_{uji} = MEG\left(TrigNet(z_u, a_{uj}, x_j), SimNet(x_j, c_{ji}, x_i)\right) \quad (6b)$$

where f_d is a function to get the relevance weight of the direct path, $PATH_{uji}$ represents the relevance weight of the 2-hop path via trigger item j , AGG is a scoring function to obtain the final relevance score between the user and the target item by summing relevance weights of $n + 1$ paths, MEG is a function to merge the relevance weights in each 2-hop path, and *TrigNet*, *SimNet* are two independent subnetworks as mentioned previously.

To enable PDN to perform fast online item retrieval, we first define MEG as the inner product or summation of the weights of the first hop and the second hop. Then f_d is defined also as the inner product between the user and the target item representations. Therefore, the final form of PDN is written as follows:

$$\hat{y}_{ui} = p_u q_i^T + \sum_{j \in N(u)} MEG\left(TrigNet(z_u, a_{uj}, x_j), SimNet(x_j, c_{ji}, x_i)\right) \quad (7)$$

The overall architecture of PDN is shown in Figure 2. Note that, in each 2-hop path, when the outputs of *TrigNet* and *SimNet* are vectors, they can be considered as representations of corresponding edges, which is called as vector-based PDN. This setting may have

higher model capacity, and make online retrieval more complicated. On the other hand, when the outputs of *TrigNet* and *SimNet* are scalars, they can be regarded as the weights of corresponding edge, which is called as scalar-based PDN. Comparing with vector-based PDN, scalar-based PDN has lower degree of freedom, which can alleviate the complexity of online retrieval using greedy strategy based on path retrieval. Hence, we introduce each component of PDN with the setting of scalar-based PDN in the following.

4.2 Feature Composition & Embedding Layer

As shown in Figure 1, there are four feature fields in our recommender system: user field z_u , user’s behavior field $\{a_{uj}\}$, item co-occurrence field $\{c_{ji}\}$, item field x for both of interacted items $\{x_j\}$ and target item x_i . These fields include one-hot features, e.g., user id, item id, age id, brand id, and continuous features, e.g., monthly sales, stay time, the statistical correlation between items. At first, we transfer dense features into one-hot scheme through the discretization. Then, each one-hot feature is projected into a fixed-size dense representation. After embedding, we concatenate embedding vectors belonging to the same field as the representation of this field. Formally, field representations of user field, user’s behavior field, item co-occurrence field and item field can be written as $E(z_u) \in \mathcal{R}^{1 \times d_u}$, $E(a_{uj}) \in \mathcal{R}^{1 \times d_a}$, $E(c_{ji}) \in \mathcal{R}^{1 \times d_c}$, $E(x) \in \mathcal{R}^{1 \times d_i}$, respectively, where d_u , d_a , d_c and d_i are the dimension size of corresponding field respectively.

4.3 Trigger Net & Similarity Net

After going through the embedding layer, we calculate the relevance weights for each 2-hop path between the user and the target item. For the first hop, we utilize *TrigNet* to capture the user's multi interests by calculating the preference for each of her triggers. Specifically, given user u and her trigger item j , the preference score is calculated as follows:

$$t_{uj} = \text{TrigNet}(z_u, a_{uj}, x_j) = \text{MLP}\left(\text{CAT}(E(z_u), E(a_{uj}), E(x_j))\right) \quad (8)$$

where $\text{CAT}(E(z_u), E(a_{uj}), E(x_j)) \in \mathcal{R}^{1 \times (d_u + d_a + d_i)}$ is the concatenation of the user embedding, user behavior embedding and interacted item embedding, and t_{uj} represents the user's preference for an interacted item j . When there are n distinct interacted items, $T_u = [t_{u1}, t_{u2}, \dots, t_{un}]$ can be considered as a variable dimension representation for user u . EBR-based methods represent user interests by one fixed dimension representation vector, which can be a bottleneck for capturing diverse interests of users [22], because all information about diverse interests of one user is mixed together, causing inaccurate item retrieval for the matching stage. Compared with EBR-based solutions that encodes the user representation with a fixed dimension vector, T_u explicitly describes the user's preference for each interacted item, which can better represent the user's diverse interests and is more interpretable.

It is worth mentioning that the *TrigNet* can employ other more powerful neural networks, such as Recurrent Neural Network (RNN) and transformer-based models for user behavior [4, 30]. However, we would like to emphasize that a simple MLP is more cost-effective to our industrial system.

As to the second hop, we utilize *SimNet* to calculate the relevance between each interacted item and target item based on the item profile and co-occurrence information:

$$s_{ji} = \text{SimNet}(x_j, c_{ji}, x_i) = \text{MLP}\left(\text{CAT}(E(x_j), E(c_{ji}), E(x_i))\right) \quad (9)$$

where $\text{CAT}(E(x_j), E(c_{ji}), E(x_i)) \in \mathcal{R}^{1 \times (2 \times d_i + d_c)}$ is the concatenation of interacted item embedding, co-occurrence embedding and target item embedding, and s_{ji} represents the relevance between item j and item i . $S_i = [s_{i1}, s_{i2}, \dots, s_{in}]$ can be considered as a variable dimension representation of target item i . We emphasize that *SimNet* explicitly learns the relevance based on co-occurrence information and side information of items. In this sense, it can be deployed independently for item-to-item retrieval.

After obtaining $\{t_{uj}\}$ and $\{s_{ji}\}$, PDN merges them to get the relevance weight PATH_{uji} of each two-hop path as follows:

$$\text{PATH}_{uji} = \text{MEG}(t_{uj}, s_{ji}) = \ln(1 + e^{t_{uj}} e^{s_{ji}}) \quad (10)$$

4.4 Direct Net

We further model the user's general interests in a broader range with another set of user and item embeddings. For example, women are more interested in dresses, while men are more interested in belts. This can be consider as a 1-hop path directly connecting the user to the target item. Hence, we utilize a direct network composed of a user tower and an item tower. Specifically, these two towers go through MLP with Leaky Rectified Linear Units (LeakyReLU) based on user field (z_u) and target item field (x_i) to output a user representation $p_u \in \mathcal{R}^{1 \times k}$ and an item representation $q_i \in \mathcal{R}^{1 \times k}$,

respectively. And then, the relevance weight of the direct path can be formulated as follows:

$$d_{u,i} = p_u q_i^T = \text{MLP}(E(z_u)) \text{MLP}(E(x_i))^T \quad (11)$$

where $d_{u,i}$ is the direct relevance between user u and target item i .

4.5 Bias Net

Position bias, and many other types of selection biases, are studied and verified to be an important factor in recommender systems [1, 5, 34]. For example, it is common that users are inclined to click items displayed closer to the top of the list, even though it was not the most useful one of the entire corpus. To remove selection biases during model training, we train a shallow tower with features contributing to selection bias, such as position feature for position bias, and hour feature for temporal bias. The resultant bias logit y_{bias} is added to the final logit of the main model, as shown in Figure 2. Note that, at serving time, the bias net is removed to get an unbiased relevance score.

4.6 Loss Function

Whether user u would click target item i can be seen as a binary classification task. Therefore, we merge the relevance weights of $n + 1$ paths and bias logit to get the final relevance score between u and i , and convert it into user click probability $p_{u,i}$:

$$\hat{y}_{u,i} = \text{softplus}(d_{u,i}) + \sum_{j=1}^n \text{PATH}_{uji} + \text{softplus}(y_{bias}) \quad (12)$$

$$p_{u,i} = 1 - \exp(-\hat{y}_{u,i}) \quad (13)$$

Note that *softplus* function produces the relevance score $\hat{y}_{u,i}$ in the range of $(0, +\infty)$. Hence, we utilize Eq. 13 to convert it to a probability value between 0 and 1. To train the model, we apply the cross-entropy objective function as: $l_{u,i} = -(y_{u,i} \log(p_{u,i}) + (1 - y_{u,i}) \log(1 - p_{u,i}))$ where $y_{u,i}$ is ground-truth label indicating whether the user clicked on the item.

4.7 Discussion

To ensure that the training of PDN can converge to a better optimum, we carefully design the relevance weight for each path. As described above, we utilize $\exp(\cdot)$ instead of other activation functions to constrain the output to be positive, i.e., $e^{s_{ji}}$ and $e^{t_{uj}}$, and further, through Eq. 10 to achieve the merge of the weights of each 2-hop path. This treatment of constraining the output to be positive is intuitive and fits real-world rationality. Note that the relevance weight would be negative in nature. However, this setting could allow PDN to search the local optimum in a much broader parameter space, which easily leads to overfitting. In Figure 3, We illustrate two bad examples by allowing the relevance weight to be negative. As shown in Figure 3a, when a negative target is connected by two totally irrelevant triggers, *SimNet* may generate a positive relevance weight for one path, and a negative one for the other path. After aggregation, the click probability is still quite low, i.e., a perfect match with the ground truth. But it is clear that milk can not have a positive relevance with a mobile phone. Similarly, as shown in Figure 3b, *TrigNet* can also learn a negative preference towards a trigger, mainly to overfit the data.

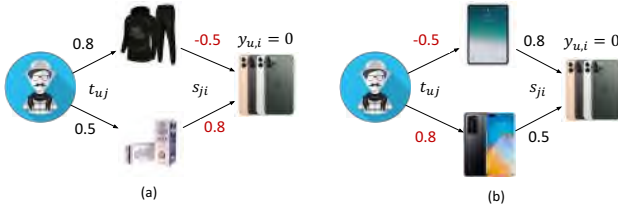


Figure 3: The bad cases in model training where (a) and (b) are negative sample. To optimize the loss of model, the similarity in (a) and the trigger weight in (b) have to be negative.

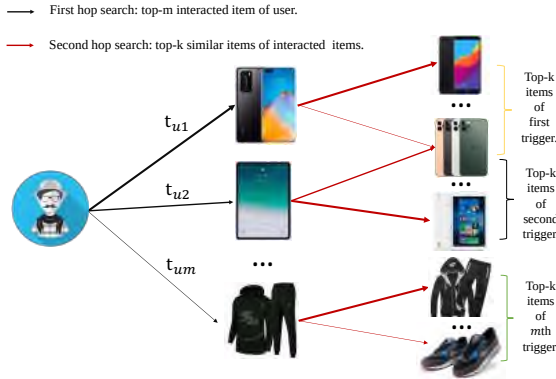


Figure 4: Online top-k path retrieval with greedy strategy.

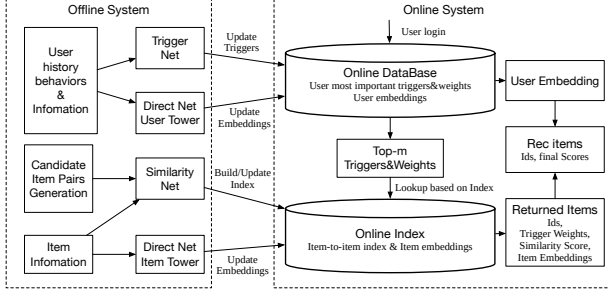


Figure 5: The framwork of online retrieval with PDN.

5 SYSTEM

In this section, we describe the implementation and deployment of PDN at Taobao for item recommendation in detail. As a user opens Mobile Taobao App, the recommender system firstly retrieves thousands of relevant items for this user from the corpus containing up to billions of items. Subsequently, every retrieved items are scored by a ranking model and the order list is displayed to the user as the recommendation.

5.1 Onling Retrieval

As aforementioned, billions or even trillions of items need to be processed in the matching stage. To satisfy the online real-time services, it is not possible to utilize PDN to score all available items. Based on architecture of PDN, it is intuitive that the larger path relevance weight, the more likely the user would be interested

in the item. Therefore, the matching problem can be regarded as retrieving the item node with the larger path weights in the user's 2-hop neighborhood. As shown in Figure 4, we implement online real-time top-K nearest neighbor retrieval in the form of path retrieval via a greedy strategy. Specifically, we decompose the path retrieval into two parts: (a) top-m important trigger search (the first hop) using the *TrigNet*, (b) top-k item search for each top-m trigger (the second hop) based on the index generated by *SimNet*. For *TrigNet*, we deploy the model in a real-time online service for user trigger scoring. For *SimNet*, we use the model to compute and index item relevance in offline fashion. In detail, the online retrieval of PDN can be summarized as:

- **Index generation (Step 1):** Based on *SimNet*, the k most similar items of each item in the corpus are generated offline and stored with the relevance score s_{ji} in the database. The details can be seen in Section 5.2
- **Trigger extraction (Step 2):** When the user opens Mobile Taobao App, we take out all the items that the user has interacted with and utilize *TrigNet* to score all her triggers t_{uj} and return top-m triggers.
- **Top-K retrieval (Step 3):** We start with these top-m triggers to query the database and obtain $m \times k$ candidate items in total. Without considering the bias feature, the top-K items are returned for recommendation as follows.

$$\hat{s}_{u,i} = \text{softplus}(d_{u,i}) + \sum_{j=1}^m \text{softplus}(t_{uj} + s_{ji}) \quad (14)$$

Note that p_u and q_i are static representations. Hence, these two representations can also be inferred offline and stored in the database. When performing online service, they can be queried directly based on user or item id.

5.2 Index Generation

For industrial systems with large item corpus, we need to compress the dense matrix of item relevance, i.e., $\mathcal{R}^{N \times N} \rightarrow \mathcal{R}^{N \times k}$, to reduce index construction time and storage cost. Even this process can be finished offline, it is too cost to calculate the relevance of all $N \times N$ item pairs, so we first reduce $N \times N$ to $N \times \hat{k}$ based on candidate generation, where $\hat{k}$ is an order of magnitude greater than k . The compression operation mainly consists of three steps:

- **Candidate item pair generation.** We mainly generate item pairs from two strategies, one is based on co-occurrence information, such as items that are clicked in the same session, the other is based on the profile information of items, such as items of the same brand. In this sense, the item pairs which have been not co-occurred before but have some similar property can also be considered as candidates.
- **Candidate item pair ranking.** We extract the relevance score s_{ji} of each item pair by using *SimNet*.
- **Index building.** For each item, we take the top-k similar items to form $N \times k$ item pairs, and store them with s_{ji} in the database.

It is worth mentioning that *SimNet* can be used independently for indexing, so it can be used also for item-to-item retrieval.

6 EXPERIEMENTS

In this section, we evaluate PDN against the existing state-of-the-art solutions with both offline and online settings, including ablation study, model analysis and case study.

Table 1: Statistics of experimental datasets.

Dataset	User#	Item#	Interaction#
MovieLens	6,040	3,706	1,000,209
Pinterest	55,187	9,916	1,500,809
Amazon Books	351,356	393,801	6,271,511

6.1 Offline Evaluation

6.1.1 Datasets and Evaluation Protocol. We experiment with three publicly real-world datasets: MovieLens², Pinterest [10], and Amazon books [12] for performance evaluation. The statistics of the three datasets are summarized in Table 1. Following the settings of [15], we filter these datasets in the same way that retained only users with at least 20 interactions. More details on this preprocessing for the three datasets have been elaborated in [15], so we do not restate here.

To evaluate the performance of PDN, we adopt the leave-one-out evaluation protocol [2, 16]. For each user, the last interaction is used as the target item, while the previous interaction items are collected as the user behaviors. Specifically, we followed the common strategy in [9, 19]. That is, all negative items are utilized in each test case, and the target items are ranked among these items. Hit Ratio (HR) [8] and Normalized Discounted Cumulative Gain (NDCG) [13] are adopted as the performance metrics. Here HR can be interpreted as a recall-based measure and NDCG is a ranking-based measure.

6.1.2 Comparing Methods. We compare PDN with the following two-tower methods and conventional item-based CF methods. The two-tower methods are introduced as follows:

- **DSSM [18].** DSSM employs embeddings to represent users and items. The relevance score is calculated based on the inner product between the user representation and the item representation.
- **Youtube DNN [6].** Youtube DNN utilizes the user’s interaction sequence to derive user representations for item recommendation. It treats each item in the user’s historical behaviors equally and adopts average pooling to extract a user’s interest. To ensure fair comparison, hyperparameter tuning is conducted by a grid search, and each method is tested with the best hyperparameters.
- **BST [4].** BST extends the Youtube DNN, by utilizing the transformer layers to capture the user’s short-term interest over the behavior sequence. We use the inner product of user and item representations instead of MLP.

The item-based CF methods are introduced as follows:

- **Pearson-based CF (PCF) [27].** This is the standard item-based CF, which estimates item relevance based on Pearson coefficient.

- **SLIM [25].** SLIM learns item relevance by minimizing the loss between the original user-item interaction matrix and the reconstructed one from the item-based CF model.

The ranking method is introduced as follows:

- **DIN [35].** Deep interest network takes target attention to extract the relationship between user’s interacted sequence and target item.

6.1.3 Results and Discussion. Table 2 shows the experimental results on three public datasets in terms of HR@10 and NDCG@10. All experiments are repeated 5 times and the average results are reported. Clearly, PDN outperforms all comparison methods under most datasets. We can make the following observations. (1) For personalized retrieval, DSSM performs worst among two-tower methods, which indicates that capturing user interest based on the user behavior is critical for recommender systems. The performance of BST is better than that of YouTube DNN, due to the fact that the transformation layer extracts users’ interests by considering the sequential information in their behaviors. PDN achieves the best performance, mainly because these two-tower methods represent each user by one fixed-dimension vector, *i.e.*, a bottleneck for modeling diverse interests. In contrast, PDN utilizes *TrigNet* to extract multi-interest user representation and each dimension describes the user’s interest in an interacted item in a fine-grained way. *SimNet* is also effective in deriving the similarity between an interacted item and the target item (ref. Figure 1). Hence, a more accurate relevance can be estimated by considering the potential interest of the user towards the target item.

(2) When deploying online, we adopt *SimNet* instead of item-based CF to estimate item similarity for item-to-item retrieval. Therefore, we conducted a set of comparison with an item-to-item strategy (*i.e.*, the columns with “Item to Item” in Table 2). *SimNet* performs best among all methods. The reason is that *SimNet* explicitly optimizes the similarity between items by integrating the item profile used by the two-tower methods and co-occurrence information used by item-based CF based on deep neural networks, which utilizes more information to solve sparsity problem encountered by CF. (3) PDN performs better than DIN. We believe that DIN ignores users’ attention to interacted items, while PDN considers both user attention and item attention for better personalized recommendation. (4) Based on ablation study, PDN yields the best performance, confirming that each component contributes to the final results.

6.2 Online Experiments

6.2.1 A/B Tests. Beyond offline studies, we conduct online A/B experiments by deploying our method in the recommender system of Taobao for two weeks. In the control setup (*i.e.*, Baseline), it includes all matching strategies in our current production system. In the variation experiment setup, we applies *SimNet* instead of the item-based method for inverted-index building. For fair comparison, the same ranking component and business logic are applied on the top of both matching stages. Table 3 and Figure 6 summarize the experimental results. We can find that the proposed PDN improves the e-commerce recommender system for all core business metrics, including page click-through rate (PCTR), user click-through rate (UCTR), clicks per user (ClkPU), average session duration (avgSD),

²<http://grouplens.org/datasets/movielens/1m/>

Table 2: HR and NDCG of different methods on the three datasets. The best results are highlighted in boldface. The improvements over the comparing methods are statistically significance at 0.05 level. 'Personalise' means that the output of the score considers the user information, e.g., the output of PDN and EBR-based method. It is worth noting that CF can't get personalized scores, which is represented by '-'. 'Item to Item' refers to retrieval based on the relevance between interacted and target items. Specifically, the acquisition strategy of item similarity is the output of item-based CF, the output of *SimNet* or the inner product between item features extracted from EBR methods.

Group	Method	MovieLens				Pinterest				Amazon Books			
		Personalise		Item to Item		Personalise		Item to Item		Personalise		Item to Item	
		HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Two-tower	DSSM [18]	0.4699	0.2603	0.2243	0.1034	0.2730	0.1394	0.1647	0.0759	0.6433	0.4163	0.1981	0.0894
	Youtube DNN [6]	0.6187	0.3579	0.4362	0.2324	0.7585	0.4351	0.6409	0.3028	0.6539	0.4370	0.2398	0.0955
	BST[4]	0.6316	0.3603	0.0561	0.0277	0.8176	0.4955	0.5371	0.2428	0.6923	0.4403	0.1019	0.0574
Item-based CF	PCF [27]	-	-	0.4033	0.2132	-	-	0.2800	0.1470	-	-	0.0547	0.0203
	SLIM [25]	-	-	0.4400	0.2337	-	-	0.6067	0.3323	-	-	0.1621	0.0679
ranking model	DIN [35]	0.6454	0.3767	0.1662	0.0918	0.8185	0.5051	0.4370	0.2059	0.6973	0.4698	0.1325	0.0612
Incorporated	PDN w/o bias net	0.6323	0.3401	0.5080	0.2795	0.8015	0.4814	0.7654	0.4345	0.6869	0.4520	0.2866	0.1546
	PDN w/o direct net	0.6642	0.3930	0.5124	0.2813	0.8123	0.5286	0.7703	0.4397	0.7012	0.4729	0.2962	0.1613
	PDN	0.6770	0.4071	0.5152	0.2859	0.8283	0.5358	0.7911	0.4613	0.7019	0.4735	0.3505	0.2049

as well as diversity³, which are considered to be good indicators of recommendation satisfaction. Especially, personalized diversity is normally hard for existing production systems, while PDN increases the diversity of recommended items by a large margin, which indicates PDN can capture the diverse interests of users with *TrigNet* and *SimNet* to improve the overall user experience. Besides, we deploy *SimNet* to verify it can be used independently to build index instead of item-based CF for item-to-item retrieval, which also gets a perform gains. Note that, the metrics online are reported with relative improvement, i.e., $(metric_{PDN} - metric_{base})/metric_{base}$.

Table 3: Online A/B test improvements in Taobao (Observed over two week).

Method	PCTR	ClkPU	AvgSD	Diversity
Baseline	+0.0%	+0.0%	+0.0%	+0.0%
<i>SimNet</i>	+9.25%	+5.43%	+4.08%	+14.68%
PDN	+18.04%	+15.38%	+7.87%	+19.60%

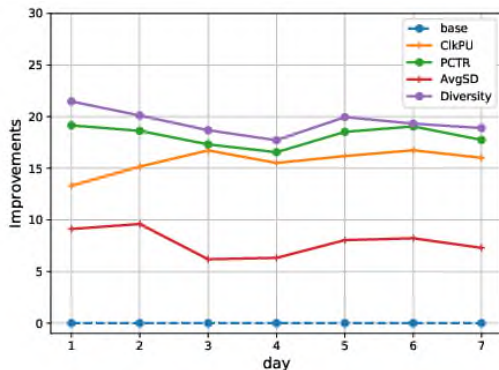


Figure 6: PDN online improvements in one week.

³Diversity represents the proportion of category coverage.

6.2.2 Online Efficiency. Here, we report the efficiency of online serving. Specifically, the overall latency from requests to candidate generation can be done within 6.75 milliseconds, i.e., the queries per second (QPS) is in the thousand level. This is comparable to the retrieval with the standard inverted index. Based on the huge performance gain and low latency, we have deployed PDN online to serve the matching stage of recommendation in Taobao.

6.3 Case Study

Table 4: Impact of user behavior sequence length (n) on HR@300 at Taobao offline logs. Percentages in the brackets indicate the relative improvements over BST.

Method	$n \leq 15$	$15 < n \leq 30$	$30 < n \leq 45$	$n > 45$	all
PCF [27]	0.066 (-46.3%)	0.110 (-59.1%)	0.131 (-31.4%)	0.141 (-41.8%)	0.120
BST [4]	0.123	0.175	0.191	0.200	0.180
PDN	0.263 (+113.8%)	0.326 (+86.3%)	0.329 (+72.3%)	0.295 (+47.5%)	0.297

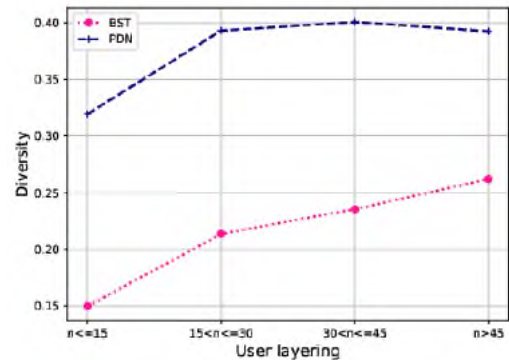


Figure 7: Diversity w.r.t. user behavior sequence length.

6.3.1 Analysis of User Behavior Sequence Length. Based on the model trained with Taobao offline log, we investigate the impact of user behavior sequence length n on performance. As shown in Table 4, we group users based on n , and found that the smaller the n is, the larger the gain is. This result shows that PDN has stronger

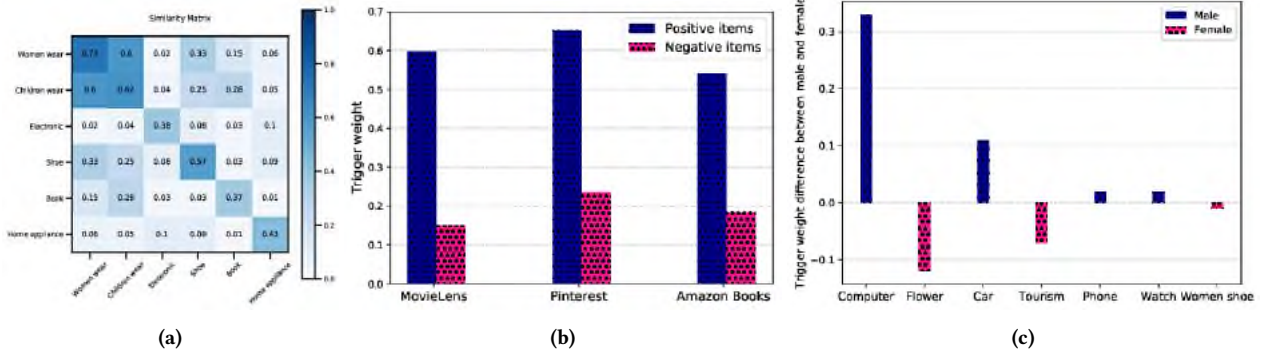


Figure 8: (a) Item-to-item similarity matrix based on *SimNet* at online serving; (b) User-to-item interests based on *TrigNet* on public datasets; (c) User group-to-category interests based on *TrigNet* at online serving.



Figure 9: Comparison of item-to-item retrieval results between *SimNet* and item-based CF based on trigger item. The similarity decreases from left to right.

robustness to n , and it can obtain superior performance even in the case of sparse user behavior sequence. As shown in Figure 7, with the increase of n , the diversity of candidates generated by PDN gradually increases, and is better than that of BST by a large margin. These results show that PDN can capture the diverse interests of users in a fine-grained way to improve the user experience.

6.3.2 Diversity and Accuracy of *SimNet*. To verify the effectiveness of *SimNet*, we utilize the same trigger item to perform item-to-item retrieval using the similarity provided by *SimNet* and item-based CF, respectively. As shown in Figure 9, retrieval through cat-shaped dolls, *SimNet* returns similar cat-shaped dolls and cups printed with cat paw patterns, while item-based CF returns dolls with high monthly sales but not related to cats, e.g., the dog-shaped dolls. This result indicates that *SimNet* can retrieve more relevant and diverse results without being disturbed by the number of interactions, while item-based CF would suffer from “Matthew Effect” [21], which introduces preference bias towards popular items. In other words, item-based CF only calculates similarity based on item co-occurrence information, while our method additionally introduces more information, e.g., item profile, user profile, user behavior, for more accurate item similarity estimation. To further verify the reliability of item similarity based on *SimNet*, we randomly select 1,000 item pairs based on each category pair and calculate their average

similarity from the inverted index built by *SimNet*. As shown in Figure 8a, we find that items belonging to the same category have a higher similarity. Besides, the similarity of related categories is higher than that of unrelated categories. For example, both the pair of women’s wear and children’s wear, and the pair of electronics and home appliances, have higher similarity than other combinations.

6.3.3 Personalization and Effectiveness of *TrigNet*. On the public datasets, we exploit the user’s interest in items based on *TrigNet*. As shown in Figure 8b, the user-to-item trigger weights are averaged on positive and negative items respectively. It is obvious that the weights on positive items are higher than that on negative items. This result indicates the reasonable of *TrigNet*. To verify the personalization ability of *TrigNet*, we randomly selected 1,000 males and 1,000 females, and randomly selected 500 items from several categories, and then apply *TrigNet* to score all user-item pairs and get the average trigger weight difference between male and female. As shown in Figure 8c, we can find that *TrigNet* treats gender differently to reflect the user characteristics precisely. For example, male prefer computer, car, and watch, while female prefer flower, tourism and women shoes.

7 CONCLUSION

In this paper, a novel model named Path-based Deep Network is proposed for candidate item generation. The proposed PDN establishes 2-hop paths from the user to the target item, leading to better personalized yet diverse item retrieval. To the best of our knowledge, this is the first work to build a retrieval architecture based on a 2-hop graph. Besides, we present how to apply this architecture to achieve online retrieval in low latency based on the proposed path retrieval and a greedy strategy. The offline experiments over three real-world datasets are conducted to demonstrate that PDN outperforms existing SOTA alternatives. Moreover, our online experiments further indicates that PDN can improve retrieval quality in terms of five industry metrics. Lastly, PDN has been successfully deployed in our online recommender system in Taobao.

ACKNOWLEDGMENTS

Chenliang Li’s work was supported by National Natural Science Foundation of China (No. 61872278).

REFERENCES

- [1] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *WSDM*.
- [2] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A generic coordinate descent framework for learning from implicit feedback. In *WWW*.
- [3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
- [4] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *DLP-KDD*.
- [5] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. Esam: Discriminative domain adaptation with non-displayed items to improve long-tail performance. In *SIGIR*.
- [6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *RecSys*. Association for Computing Machinery.
- [7] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *RecSys*.
- [8] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.
- [9] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*.
- [10] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning image and user features for recommendation in social networks. In *ICCV*.
- [11] Carlos A Gomez-Urbe and Neil Hunt. 2015. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2015), 1–19.
- [12] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*.
- [13] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *CIKM*.
- [14] X. He, Z. He, J. Song, Z. Liu, Y. Jiang, and T. Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (Dec 2018), 2354–2366.
- [15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
- [16] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*.
- [17] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *SIGKDD*.
- [18] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. *CIKM*.
- [19] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*.
- [20] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. <https://doi.org/10.1109/MC.2009.263>
- [21] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An adversarial approach to improve long-tail performance in neural collaborative filtering. In *CIKM*.
- [22] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*.
- [23] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.
- [24] David C Liu, Stephanie Rogers, Raymond Shiau, Dmitry Kislyuk, Kevin C Ma, Zhigang Zhong, Jenny Liu, and Yushi Jing. 2017. Related pins at pinterest: The evolution of a real-world recommender system. In *WWW*.
- [25] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *ICDM*.
- [26] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*.
- [27] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *WWW*. Association for Computing Machinery.
- [28] Brent Smith and Greg Linden. 2017. Two decades of recommender systems at Amazon. com. *Ieee internet computing* 21, 3 (2017), 12–18.
- [29] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).
- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [31] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations. In *WWW*.
- [32] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *RecSys*.
- [33] Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wen-Yun Yang. 2020. Towards Personalized and Semantic Retrieval: An End-to-End Solution for E-commerce Search via Embedding Learning. In *SIGIR*.
- [34] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *RecSys*.
- [35] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *SIGKDD*. Association for Computing Machinery, New York, NY, USA.

第六部分

年度必看的技术&行业报告

- ◎ 2021热门技术新动向报告
- ◎ 2021热门行业新趋势报告
- ◎ 2021热门主题特别篇：你值得关注的【元宇宙】

• 2021热门技术新动向报告

1. 《2021年达摩院十大科技趋势》

内容简介：

从2020年疫情爆发开始，数字科技就迅速成为人类与疫情抗争的重要力量，比如AI医疗影像辅助诊疗与提升药物研发的效率和精度，再到健康码与智慧城市都有效地帮助我们实现对疫情的精准防控。科技服务社会需求，科技服务产业发展，一起来看看2021达摩院的十大科技趋势。

下载地址：

<https://files.alicdn.com/tpsservice/c4b82c52c17bf37a8d0a4b7a755ac047.pdf>

2. 《阿里巴巴新基建洞察之5G智能经济应用场景》

内容简介：

在5G商用的半年时间里，大家似乎身边并没有感受到 5G 带来的变化，甚至连 5G 手机都看不到几款。那么 5G 真的很快吗？5G 真的会引爆行业应用和机遇吗？哪些行业是 5G 的典型应用场景？它们都有哪些价值？这正是本研究报告要探索的内容。

下载地址：

<https://files.alicdn.com/tpsservice/6ed5ad8cc5ab7a6e609c513f107ef072.pdf>

3. 《未来组织数字化发展评估报告》

内容简介：

数智社会，数据成为最重要的生产要素，经济社会全面数字化、网络化和智能化，不断催生新业态新模式新产业的出现。本篇报告你可以看到未来组织数字化的评价指标构建和定义。

下载地址：

<https://files.alicdn.com/tpsservice/355eeba3de1a66545c211e3363574140.pdf>

4.《云计算的社会经济价值和区域发展评估》

内容简介：

目前，云计算作为一种基于大数据的共享服务模式，通过搭建数据平台、优化计算方式、丰富应用场景，不断提升社会治理的精细化水平和公共服务的便捷化水平。企业上云对社会发展有什么作用？中国目前的云计算发展有哪些特征？云计算到底如何运行？云平台如何助力社会治理？云服务怎样影响你我生活？本研究尝试提供这些问题的参考建议。

下载地址：

<https://files.alicdn.com/tpsservice/2f642e557c8f80255967123986570fae.pdf>

5.《2021年低代码行业研究报告》

内容简介：

本报告研究范围主要围绕独立的低代码开发平台，向APaaS、零代码、BPM等与低代码相关的其他概念延伸进行分析，低代码相关的组件如流程引擎、数据分析引擎等也在本报告研究范围内。

下载地址：

<https://files.alicdn.com/tpsservice/876137c84dff5d8055f593b9d2ea2ffd.pdf>

6.《2021年中国人工智能基础层行业发展研究报告》

内容简介：

2020年中国人工智能核心产业规模超过1500 亿元；其中人工智能基础层市场规模为497亿元，为AI产业总规模的33%。在人工智能由技术落地应用阶段向效率化生产阶段转变的背景下，人工智能基础层的各模块工具有望走向集约型的生产模式。

下载地址：

<https://files.alicdn.com/tpsservice/3db5d00dcbc10a391fb9887d2071325d.pdf>

7.《2021年中国数据库行业研究报告》

内容简介：

约2015年起，中国数据库市场进入了百花齐放、活跃创新的阶段，但仍面临多方挑战：1)分布式数据库在事务、性能等环节仍待优化；2)信创为国产厂商提供成长沃土，未来发展仍待市场磨炼；3)数据频繁迁移、多库长期并存

为厂商提出新的诉求;4)CPU、内存等硬件变化为数据库设计提供更多的想象空间。综合供需两端视角，中国数据库未来发展将呈现多种趋势。

下载地址：

<https://files.alicdn.com/tpsservice/990838ade882c9f05949413685659a9c.pdf>

• 2021热门行业新趋势报告

8.《2021直播电商区域发展指数报告》

内容简介：

2020年直播电商整体规模将突破万亿，达到10500亿元，渗透率将达到8.6%。作为国民经济的基本单元，县域经济的活跃度被视为国民经济的重要风向标。基于指数分析，发掘直播产业发展特色区县，总结、梳理县域直播电商特征及典型模式，供其他区域发展参考借鉴。

下载地址：

<https://files.alicdn.com/tpsservice/37bdb466882ed019d275c6dc6cb9afc.pdf>

9.《2021线上新品消费趋势报告》

内容简介：

2021年，中国经济迈向新阶段，商业数字化加速，线上实物零售占比已达25%，且仍在不断攀升。在线上化的进程中，新品表现抢眼，近年来数量年年翻番，成为驱动线上大盘增长的重要动因。作为线上新品发布平台代表，天猫已进入新品电商时代，联合品牌和消费者一同致力于推动新品研发，使C2B成为可能。

下载地址：

<https://files.alicdn.com/tpsservice/efd83df61ee0c8c85ecfe81a877478b3.pdf>

10.《2021中国消费品牌发展报告》

内容简介：

疫情期间，不同时代的中国消费品牌的表现各有特色。供给侧改革以来，新品牌快速崛起。可比代表性品牌2020年销售额达到2016年的17.9倍。

下载地址：

<https://files.alicdn.com/tpsservice/f3e96e6b528511158015fa68d98cc294.pdf>

11.《2021年中国人工智能+医疗与生命科学行业研究报告》

内容简介：

AI医疗影像的部分诊断类产品已深入红海阶段，产品在三甲医院市场的渗透与覆盖在未来可能即将封顶，对此，AI医疗影像玩家将继开拓影像诊断的其他疾病市场，如冠脉、乳腺、肝脏等，同时开发手术规划与导航这类影像治疗市场，转战新兴的蓝海区。

下载地址：

<https://files.alicdn.com/tpsservice/f3e96e6b528511158015fa68d98cc294.pdf>

12. 《2021年种草内容平台营销价值白皮书》

内容简介：

本次研究将梳理种草内容平台的发展演进及典型平台差异特征，从品牌主和行业专家角度探寻种草在营销活动中的价值和机会，帮助品牌发掘种草内容平台在营销链路中的独特价值。

下载地址：

<https://files.alicdn.com/tpsservice/fde025af1f076249071108194754523d.pdf>

13. 《2021年花草茶市场洞察报告》

内容简介：

2020到2021年，花草茶市场的成交额增速接近茶这一大品类市场的两倍。中高端市场均增速迅猛，未来花草茶品类进一步提升需要找到可持续性消费的品类使用利基。本报告尝试从天猫平台官方数据中找到一些潜在的市场思路。由黑马工厂出品。

下载地址：

<https://files.alicdn.com/tpsservice/0cbc57312a5f1a835739aaf1e10a9d53.pptx>

• 2021热门主题特别篇： 你值得关注的【元宇宙】

14.《2020-2021年 元宇宙发展研究报告》

内容简介：

元宇宙是整合多种新技术而产生的新型虚实相融的互联网应用和社会形态，它基于扩展现实技术提供沉浸式体验，基于数字孪生技术生成现实世界的镜像，基于区块链技术搭建经济体系，将虚拟世界与现实世界在经济系统、社交系统、身份系统上密切融合，并且允许每个用户进行内容生产和世界编辑。元宇宙仍是一个不断发展、演变的概念，不同参与者以自己的方式不断丰富着它的含义。

下载地址：

<https://files.alicdn.com/tpsservice/0c9686decd4879207785702631952cfc.pdf>

15.《通信行业元宇宙深度报告： 十问元宇宙，如何将抽象的概念具象化》

内容简介：

元宇宙发展的最关键的部分在于元宇宙的载体与内容，即“我们如何构建元宇宙？”与“元宇宙中有什么”。尽管如此，元宇宙的载体与内容这两个概念仍十分宽泛。人们对于元宇宙的构想十分多元且抽象，我们希望通过十个具体的问题及研究，将抽象的元宇宙具象化。

下载地址：

<https://files.alicdn.com/tpsservice/25742ca91975e46137e004cb36287f0e.pdf>

16.特别关注：了解一下20年前的“元宇宙”观

内容简介：

1990年11月27日，钱学森在写给汪成为院士的信中提到，“Virtual Reality”中文译名可以是“人为景境”或“灵境”，并表示他“特别喜欢‘灵境’，中国味特浓”。他认为，灵境技术的产生和发展将扩展人脑的感知和人机结合的体验，使人与计算机的结合进入到深度结合的时代。推荐几位老先生1995年的文章，《灵境(虚拟现实)是建立人机和谐仿真系统的关键技术》、《灵境技术与人机和谐仿真环境》、《虚拟现实技术及系统》。

下载地址：

<https://files.alicdn.com/tpsservice/08f6085ea75cb3c7fb6e1b171ab5ab76.zip>

注：以上内容均基于公开资源内容整理

● 附录：

【阿里云研究院】

阿里云研究院是阿里云智能事业群数智化转型智库机构，汇集多名来自数字科技头部企业、国际知名咨询机构和国家高端研究平台的资深专家。以“定义行业、洞察态势、拓展赛道、引领心智”为愿景，以“引领数智化转型新思想”为目标，致力于“用科技探索新商业边界”。研究领域涵盖云计算、人工智能、大数据与产业互联网在新零售、新制造、新金融、新能源等行业数智化转型路径及商业实践，关注前沿科技趋势、数字创新、数字治理、新基建等方向。依托指数分析、战略顶设、策略参考、产业研判、案例透视等多类型研究产品，阿里云研究院联合国家科研机构、顶尖智库、头部高校、行业协会、咨询机构、投资机构、合作伙伴和客户，拉通阿里巴巴集团各事业群，共拓研究新生态。联系人邮箱：peijie.wpj@alibaba-inc.com

【阿里研究院】

阿里研究院成立于2007年，是国内互联网企业中第一家内设研究智库。过去12年，阿里研究院见证、参与和推动了电子商务、数字经济的发展，已成为在国内外数字经济和数字治理研究领域具有广泛影响力的企业智库。官网：<http://www.aliresearch.com/>

【艾瑞咨询】

艾瑞咨询深耕商业决策服务领域，以技术和数据为驱动，为客户提供专业的行业分析、数据产品、市场研究、战略咨询、数字化解决方案等专业服务，助力客户提升认知水平、盈利能力和综合竞争力，为机构的商业决策赋能。专注互联网、新经济、数据化领域 20年，拥有超过 200位 专业研究咨询分析师，累计发布超过3000份行业报告，80% 以上的互联网企业IPO报告引用艾瑞研究成果。跳转官网链接：产业研究报告-艾瑞咨询(iresearch.com.cn)

【TMIC新品创新中心】

TMIC新品创新中心是全球最大的数字化新品研发平台。TMIC以新品数字系统为品牌提供了一套标准的数字化新品研发链路，基于大数据+小样本的洞察挖掘极大加速品牌新品创新的周期和提升成功率。全年实现数字化新品研发超5000款，上市成功率高达70%。



出品单位：
阿里巴巴大淘宝技术部
产品技术战略发展
2021.12.12