2023美团技术年货

CODE A BETTER LIFE

年度合集



美团

序

新春将至，一年一度的美团技术年货也如期到来！

星海横流，岁月成碑。2023 年，美团技术博客走过了整整十个春秋，累计发布了 580 多篇技术文章，感谢大家的一路相伴。

在龙年春节到来之际，我们精选过去一年公众号 30 多篇技术文章和科研论文，整理制作成一本 600 多页的电子书，作为新年礼物赠送给大家。

这本电子书内容覆盖算法、前端、后端、系统、运维、测试等多个技术领域，希望能对同学们的工作和学习有所帮助。

也欢迎大家转给更多有相同兴趣、积极上进的同事和朋友们，一起切磋，共同成长。

面对未来，希望大家有「无惧前路雨潇潇」的勇气，也兼具「乘风破浪会有时」的魄力。

知不足而奋进，望远山而力行。祝愿大家在甲辰龙年，幸福平安，行稳致远。

目录

算法 1

SOTA！目标检测开源框架 YOLOv6 3.0 版本来啦 1

交互式推荐在外卖场景的探索与应用 10

如何利用「深度上下文兴趣网络」提升点击率? 28

美团多场景建模的探索与实践 39

美团外卖智能陪伴型导购的探索与实践 54

大规模食品图像识别：T-PAMI 2023 论文解读 78

斩获 CVPR 2023 竞赛 2 项冠军 | 美团街景理解中视觉分割技术的探索与应用 98

美团外卖推荐关于用户新颖体验优化的技术探索 114

低延迟流式语音识别技术在人机语音交互场景中的实践 130

ICDE 2023 | 多场景多任务学习在美团到店餐饮推荐的实践 141

KDD 2023 | 美团技术团队精选论文解读 153

CVPR 2023 | 美团技术团队精选论文解读 162

后台 / 系统 172

Code：美团代码托管平台的演进与实践 172

一次「找回」TraceId 的问题分析与过程思考 191

美团开放平台 SDK 自动生成技术与实践 216

MJDK 如何实现压缩速率的 5 倍提升? 232

基于 AI+ 数据驱动的慢查询索引推荐 243

美团视觉 GPU 推理服务部署架构优化实践 251

MRCP 在美团语音交互中的实践和应用 263

前端 280

美团前端研发框架 Rome 实践和演进趋势 280

Robust 2.0：支持 Android R8 的升级版热修复框架 306

美团到店终端从标准化到数字化的演进之路 318

测试 339

如何提供一个可信的 AB 测试解决方案 339

基于 UI 交互意图理解的异常检测方法 363

代码变更风险可视化系统建设与实践 388

基于模式挖掘的可靠性治理探索 406

运维 423

超大规模数据库集群保稳系列之一：高可用系统 423

超大规模数据库集群保稳系列之二：数据库攻防演练建设实践 440

超大规模数据库集群保稳系列之三：美团数据库容灾体系建设实践 458

MySQL 自治平台建设的内核原理及实践（上） 473

MySQL 自治平台建设的内核原理及实践（下） 492

AIOps 在美团的探索与实践——事件管理篇 517

2023 精选论文 535

PIER- Permutation-Level Interest-Based End-to-End Re-ranking Framework in E-commerce 535

A Collaborative Transfer Learning Framework for Cross-domain Recommendation 544

Modeling Dual Period-Varying Preferences for Takeaway Recommendation 554

A Multi-stage Framework for Online Bonus Allocation Based on Constrained User Intent Detection 565

C-AOI- Contour-based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform 576

NEON- Living Needs Prediction System in Meituan 586

A Hybrid Approach of Statistics and Embeddings for Multilingual and Multi-Locale Recommendation 597

Divide and Adapt- Active Domain Adaptation via Customized Learning 601

Efficient Second-Order Plane Adjustment 611

算法

SOTA！目标检测开源框架 YOLOv6 3.0 版本来啦

作者：楚怡 奕非 露露

1. 概述

1 月 6 日，美团视觉智能部发布了 YOLOv6 3.0 版本，再一次将目标检测的综合性能推向新高。本次更新除了对 YOLOv6-N/S/M/L 模型进行全系列升级之外，还推出了大分辨率 P6 模型。其中，YOLOv6-L6 检测精度达到 57.2% AP，在 T4 卡上推理速度可达 29 FPS，超越 YOLOv7-E6E，取得当前实时目标检测榜单 SOTA。

技术报告：YOLOv6 v3.0: A Full-Scale Reloading

YOLOv6 Github 传送门：https://github.com/meituan/YOLOv6，欢迎 Star 收藏，随时取用。

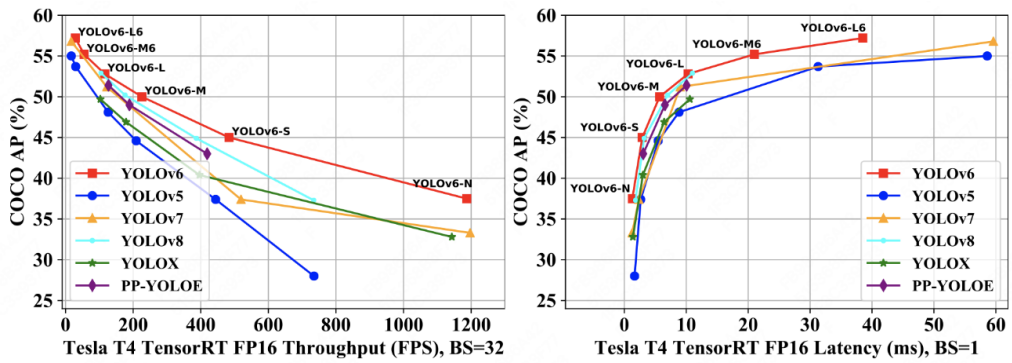


图 1　YOLOv6 各尺寸模型与其他 YOLO 系列框架的性能对比图

Method	Input Size	AP^{val}	AP^{val}_{50}	FPS (bs=1)	FPS (bs=32)	Latency (bs=1)	Params	FLOPs
YOLOv5-N [5]	640	28.0%	45.7%	602	735	1.7 ms	1.9 M	4.5 G
YOLOv5-S [5]	640	37.4%	56.8%	376	444	2.7 ms	7.2 M	16.5 G
YOLOv5-M [5]	640	45.4%	64.1%	182	209	5.5 ms	21.2 M	49.0 G
YOLOv5-L [5]	640	49.0%	67.3%	113	126	8.8 ms	46.5 M	109.1 G
YOLOv5-N6 [5]	1280	36.0%	54.4%	172	175	5.8 ms	3.2 M	18.4 G
YOLOv5-S6 [5]	1280	44.8%	63.7%	103	103	9.7 ms	12.6 M	67.2 G
YOLOv5-M6 [5]	1280	51.3%	69.3%	49	48	20.1 ms	35.7 M	200.0 G
YOLOv5-L6 [5]	1280	53.7%	71.3%	32	30	31.3 ms	76.8 M	445.6 G
YOLOv5-X6 [5]	1280	55.0%	72.7%	17	17	58.6 ms	140.7 M	839.2 G
YOLOX-Tiny [3]	416	32.8%	50.3%*	717	1143	1.4 ms	5.1 M	6.5 G
YOLOX-S [3]	640	40.5%	59.3%*	333	396	3.0 ms	9.0 M	26.8 G
YOLOX-M [3]	640	46.9%	65.6%*	155	179	6.4 ms	25.3 M	73.8 G
YOLOX-L [3]	640	49.7%	68.0%*	94	103	10.6 ms	54.2 M	155.6 G
PPYOLOE-S [17]	640	43.1%	59.6%	327	419	3.1 ms	7.9 M	17.4 G
PPYOLOE-M [17]	640	49.0%	65.9%	152	189	6.6 ms	23.4 M	49.9 G
PPYOLOE-L [17]	640	51.4%	68.6%	101	127	10.1 ms	52.2 M	110.1 G
YOLOv7-Tiny [16]	416	33.3%*	49.9%*	787	1196	1.3 ms	6.2 M	5.8 G
YOLOv7-Tiny [16]	640	37.4%*	55.2%*	424	519	2.4 ms	6.2 M	13.7 G*
YOLOv7 [16]	640	51.2%	69.7%*	110	122	9.0 ms	36.9 M	104.7 G
YOLOv7-E6E [16]	1280	56.8%	74.4%*	16	17	59.6 ms	151.7 M	843.2 G
YOLOv8-N [6]	640	37.3%	52.6%*	561	734	1.8 ms	3.2 M	8.7 G
YOLOv8-S [6]	640	44.9%	61.8%*	311	387	3.2 ms	11.2 M	28.6 G
YOLOv8-M [6]	640	50.2%	67.2%*	143	176	7.0 ms	25.9 M	78.9 G
YOLOv8-L [6]	640	52.9%	69.8%*	91	105	11.0 ms	43.7 M	165.2 G
YOLOv6-N	640	37.0% / 37.5%‡	52.7% / 53.1%‡	779	1187	1.3 ms	4.7 M	11.4 G
YOLOv6-S	640	44.3% / 45.0%‡	61.2% / 61.8%‡	339	484	2.9 ms	18.5 M	45.3 G
YOLOv6-M	640	49.1% / 50.0%‡	66.1% / 66.9%‡	175	226	5.7 ms	34.9 M	85.8 G
YOLOv6-L	640	51.8% / 52.8%‡	69.2% / 70.3%‡	98	116	10.3 ms	59.6 M	150.7 G
YOLOv6-N6	1280	44.9%	61.5%	228	281	4.4 ms	10.4 M	49.8 G
YOLOv6-S6	1280	50.3%	67.7%	98	108	10.2 ms	41.4 M	198.0 G
YOLOv6-M6	1280	55.2%‡	72.4%‡	47	55	21.0 ms	79.6 M	379.5 G
YOLOv6-L6	1280	57.2%‡	74.5%‡	26	29	38.5 ms	140.4 M	673.4 G

表 1　YOLOv6 各尺寸模型与其他 YOLO 系列框架的性能对比结果

注：YOLOv6 系列模型均在训练 300epoch 且不使用预训练模型或额外检测数据集下获得，“‡”表示采用了自蒸馏算法，“*”表示从官方代码库对发布模型进行重新测评的指标。以上速度指标均在 T4 TRT7.2 环境下测试。

2. 关键技术介绍

本次更新主要在 Neck 网络设计、训练和蒸馏策略等方面进行了创新和优化：

- 设计了表征能力更强的可重参化双向融合 PAN（RepBi-PAN）Neck 网络；
- 提出了全新的锚点辅助训练（Anchor-Aided Training）策略；
- 提出了解耦定位蒸馏（Decoupled Location Distillation）策略以提升小模型的性能。

2.1 表征能力更强的 RepBi-PAN Neck 网络

有效的多尺度特征融合网络对目标检测的效果尤为关键。特征金字塔网络 (FPN) 通过自上而下的路径来融合来自骨干网络不同 Stage 的输出特征以弥补网络学习过程中目标位置信息的损失。鉴于单向信息流传输的局限性，PANet 在 FPN 之上添加了一个额外的自底向上路径。 BiFPN 为不同的输入特征引入了可学习的权重，并简化了 PAN 以实现更好的性能和更高的效率。PRB-FPN 通过具有双向融合的并行残差 FPN 结构来保留高质量的特征，以进行准确定位。

受到上述工作的启发，我们提出了一个表征能力更强的可重参化双向融合 PAN（RepBi-PAN）Neck 网络。一般而言，骨干网络浅层特征分辨率高，具有丰富的空间信息，有利于目标检测中的定位任务。为了聚合浅层特征，常见的做法是在 FPN 中增加 P2 融合层以及一个额外的检测头，但这往往会带来较大的计算成本。

为了实现更好的精度和时延权衡，我们设计了一个双向联结（Birectional Concatenate, BiC）模块，在自上而下的传输路径中引入自底向上的信息流，使得浅层特征能以更高效的方式参与多尺度特征融合，进一步增强融合特征的表达能力。此模块能够帮助保留更准确的定位信号，这对于小物体的定位具有重要意义。

此外，我们对上一版本的 SimSPPF 模块进行了特征增强优化，以丰富特征图的表示能力。我们发现 YOLOv7 使用的 SPPCSPC 模块能够提升检测精度，但对网络推理速度的影响较大。于是我们对其进行了简化设计，在检测精度影响不大的情况下，大大提升了推理效率。同时，我们引入了可重参数化思想并对 Neck 网络的通道宽度和深度进行了相应的调整。最终 RepBi-PAN 网络结构如下图 2 所示：

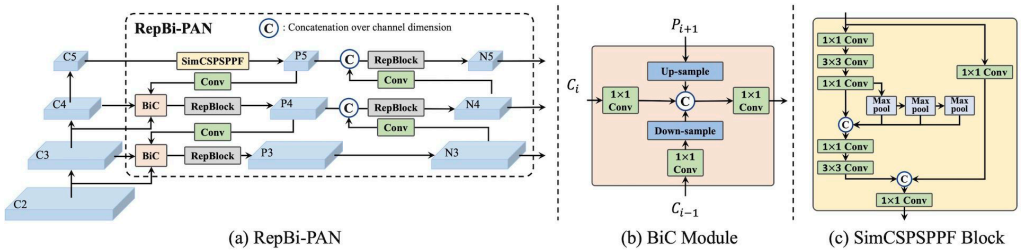


图 2　RepBi-PAN 网络结构图

Model	BiC		AP^{val}	AP^{s}	AP^{m}	AP^{l}	FPS (bs=32)
	Bottom-up	Top-down					
YOLOv6-S	✗	✗	43.1%	23.4%	48.0%	59.9%	**513**
	✗	✓	**43.7%**	**25.2%**	**48.7%**	**60.4%**	492
	✓	✓	**43.7%**	25.0%	**48.7%**	59.7%	485
YOLOv6-L	✗	✗	50.9%	32.4%	56.0%	**68.0%**	**125**
	✗	✓	**51.3%**	**34.2%**	56.5%	67.6%	120
	✓	✓	51.1%	33.6%	**56.7%**	67.9%	119

表 2　BiC 模块消融实验结果

从表 2 可以看到，在 YOLOv6-S/L 模型上，仅在 PAN 网络自上而下的传输路径引入 BiC 模块后，对推理速度影响保持在 4% 的情况下，检测精度分别提升 0.6% 和 0.4% AP。当我们尝试额外地在自底向上的信息流中将常规联结替换成 BiC 模块时，反而没有获得进一步正向的增益，因此我们仅在自上而下的路径中应用 BiC 模块。与此同时，我们还注意到，BiC 模块能够为小目标的检测精度带来 1.8% AP 的提升。

Model	SPP Blocks	AP^{val}	FPS (bs=32)
YOLOv6-N	SimSPPF [5]	35.8%	**1190**
	SimSPPF [5]*3	35.9%	1072
	SimSPPCSPC [16]	**37.4%**	1078
	SimCSPSPPF	36.9%	1176
YOLOv6-S	SimSPPF [5]	43.7%	**492**
	SimSPPF [5]*3	43.6%	447
	SimSPPCSPC [16]	44.0%	432
	SimCSPSPPF	**44.1%**	477
YOLOv6-M	SimSPPF [5]	48.6%	**227**
	SimCSPSPPF	**48.7%**	218
YOLOv6-L	SimSPPF [5]	**51.3%**	**120**
	SimCSPSPPF	51.1%	117

表 3　不同的 SPP 模块对模型精度和速度的对比结果

在表 3 中，我们对不同的 SPP 模块对模型精度和速度影响做了实验对比，其中包括经过我们简化设计的 SPPF、SPPCSPC 和 CSPSPPF 模块。除此之外，我们还尝试了在骨干网络 C3、C4 和 C5 的输出特征后分别采用了 SimSPPF 模块以加强特征的

聚合表达，在表中用 SimSPPF * 3 表示。从实验结果来看，重复使用 SimSPPF 模块虽然增加了计算量，但并没有带来检测精度的进一步提升。

经简化设计的 SPPCSPC 模块对比 SimSPPF 模块 在 YOLOv6-N/S 模型上分别提升了 1.6% 和 0.3% AP，但对推理速度 FPS 降低约 10%。而当我们将 SimSPPF 模块替换为优化后的 SimCSPSPPF 模块后，在 YOLOv6-N/S/M 模型上分别取得了 1.1%/0.4%/0.1% 的精度增益，同时推理速度对比 SimSPPCSPC 模块有较大的提升。因此，为了更好的精度 - 效率权衡，在 YOLOv6-N/S 上采用 SimCSPSPPF 模块，而在 YOLOv6-M/L 上采用 SimSPPF 模块。

2.2 全新的锚点辅助训练（Anchor-Aided Training）策略

基于深度学习的目标检测技术从学习范式上主要可分为 Anchor-based 和 Anchor-free 两大类，这两类方法针对不同尺度的目标检测上分别存在不同的优势。我们使用 YOLOv6-N 作为基线，对 Anchor-based 和 Anchor-free 范式的异同点进行了相关的实验和分析。

Paradigm	AP^{val}	AP^{s}	AP^{m}	AP^{l}
Anchor-free	35.5%	16.0%	39.5%	51.0%
Anchor-based	**35.6%**	**17.2%**	**39.8%**	**52.1%**

表 4　YOLOv6-N 上 Anchor-free 和 Anchor-based 比较

从表 4 中可以看出，当 YOLOv6-N 分别采用 Anchor-based 和 Anchor-free 训练范式时，模型的整体 mAP 几乎接近，但采用 Anchor-based 的模型在小、中、大目标上的 AP 指标会更高。从以上的实验可以得出结论：相比于 Anchor-free 范式，基于 Anchor-based 的模型存在额外的性能增益。

同时我们发现，YOLOv6 使用 TAL 进行标签分配时，其模型精度的稳定性与是否采用 ATSS 预热有较大关系。当不使用 ATSS 预热时，对同样参数配置的 YOLOv6-N 进行多次训练，模型精度最高可达 35.9% mAP，最低至 35.3% mAP，相同模型会有 0.6% mAP 的差异。但当使用 ATSS 预热时，模型精度最高却只能到达 35.7% mAP。从实验结果可以分析得出，ATSS 的预热过程利用了 Anchor-based 的预设信息，进而达到稳定模型训练的目的，但也会在一定程度上限制网络的峰值能力，因此

并不是一种最优的选择。

受到上述工作的启发，我们提出了基于锚点辅助训练（Anchor-Aided Training，AAT）策略。在网络训练过程中，同时融合 Anchor-based 和 Anchor-free 的两种训练范式，并对全阶段网络进行映射及优化，最终实现了 Anchor 的统一，充分发挥了结合不同 Anchor 网络的各自优势，从而进一步提升了模型检测精度。具体来说：

- 一方面，我们会在网络的分类头和回归头上分别添加 Anchor-based 辅助分支，在训练阶段，该分支与 Anchor-free 分支分别进行独立的 Loss 计算，之后会对 Loss 进行相加，各自反向传播进行网络的优化。通过 Anchor-based 辅助分支，为网络训练引入额外的内嵌指导信息，并与 Anchor-free 分支的信息进行整合，从而达到对结合不同 Anchor 网络的全方位融合的目的，进一步挖掘网络自身的潜力，充分发挥其效能。
- 另一方面，在网络标签匹配的过程中引入了同特征点密集采样的机制。通过扩大每次样本匹配过程中所选取候选框的范围，增加候选框中正样本的数量，并且对同一特征点重复投放采样点，进一步提升在训练过程中候选框的质量。与此同时，在网络的每一层中还会搭配原始的 Anchor-free 分支，进一步提升候选框的多样性。

除此之外，我们还提出灵活配置的训练策略，仅在训练过程中引入额外的辅助分支，在测试过程中不予使用。最终在不增加推理时间的情况下，提升网络精度，无痛涨点。最终 AAT 策略的示意图如下图 3 所示：

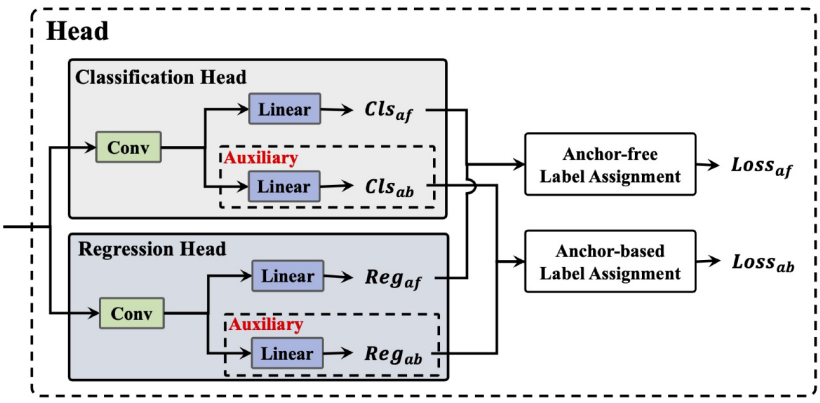


图 3　AAT 训练策略示意图

采用 AAT 训练策略的消融实验结果如下表 5 所示。我们在 YOLOv6 的各尺寸模型上进行了实验，其中 YOLOv6-S 模型采用 AAT 策略后有 0.3% 的精度增益，而在 YOLOv6-M/L 模型上分别带来了 0.5% 的精度增益。值得注意的是，YOLOv6-N/S/M 在小目标检测的精度指标得到了显着增强。

Method	AAT	AP^{val}	AP^{s}	AP^{m}	AP^{l}
YOLOv6-N	✗	**36.9%**	17.2%	41.1%	52.9%
	✓	**36.9%**	**18.7%**	**41.2%**	**53.0%**
YOLOv6-S	✗	44.1%	24.7%	48.7%	**61.1%**
	✓	**44.4%**	**25.4%**	**49.6%**	60.2%
YOLOv6-M	✗	48.6%	29.7%	53.7%	**65.5%**
	✓	**49.1%**	**31.1%**	**54.0%**	65.4%
YOLOv6-L	✗	51.3%	**34.2%**	56.5%	67.6%
	✓	**51.8%**	33.4%	**56.8%**	**68.8%**

表 5　AAT 消融实验结果

2.3 无痛涨点的 DLD 解耦定位蒸馏策略

在目标检测的蒸馏任务中，LD 通过引入 DFL 分支，从而达到了在网络中对定位信息蒸馏的目的，使分类和定位信息得以同步回传，弥补了 Logit Mimicking 方法无法使用定位蒸馏信息的不足。但是，DFL 分支的添加，对于小模型速度的影响是很明显的。添加了 DFL 分支后，YOLOv6-N 的速度下降了 16.7%，YOLOv6-S 的速度下降了 5.2%。而在实际的工业应用当中，对于小模型速度的要求往往很高。因此，目前的蒸馏策略并不适合于工业落地。

针对这个问题，我们提出了基于解耦检测任务和蒸馏任务的 DLD（Decoupled Location Distillation）算法。DLD 算法会在网络每一层的回归头上分别添加了额外的强化回归分支，在训练阶段，该分支同样会参与 IoU 损失的计算，并将其累加到最终的 Loss 中。

通过增加的额外的强化回归分支，可以对网络添加更多的额外约束，从而对网络进行更全面细致的优化。并且，DLD 算法在对强化回归分支进行训练时，引入了分支蒸馏学习策略。分支蒸馏学习策略会仅使用 DFL 分支参与网络标签分配的过程，并将标签分配的结果投入到强化回归分支进行引导学习，从而参与强化回归分支的损失函数计算和

反向传播优化。

- 一方面，DFL 分支的精度更高，在整个训练周期可以起到对强化分支蒸馏的作用，进一步提升强化分支的精度。
- 另一方面，通过分支蒸馏进行的引导学习，可以进一步将 DFL 分支的效果传递给强化回归分支，为之后的灵活配置起到铺垫作用。

除此之外，DLD 算法同样搭配了灵活配置的训练策略，在训练过程中采用双回归分支结构，对网络进行更全面细致的优化，进一步对齐双分支的回归能力。在测试过程中，移除掉冗余的 DFL 分支，仅保留强化回归分支，在简化网络的同时保持网络精度，最终实现了对目标检测算法可无痛涨点的 DLD 蒸馏算法。DLD 的消融实验结果如下表 6 所示：

DLD	Double epochs	AP^{val}
✗	✗	44.4%
✗	✓	44.6%
✓	✗	**45.1%**

表 6　DLD 在 YOLOv6-S 上的消融实验结果

在表 6 中，我们在 YOLOv6-S 模型上分别对比了训练双倍轮数和采用 DLD 策略对模型性能的影响，从实验数据可以看出，当训练 600epoch 时，YOLOv6-S 仅能达到 44.6% mAP。而采用 DLD 蒸馏策略后，YOLOv6-S 检测精度比使用双倍轮数训练的高 0.5%，最终达到 45.1%。由此可得，DLD 蒸馏策略可在不影响推理效率的前提下，提升小模型的检测精度，实现无痛涨点。

3. 总结

本文对 YOLOv6 3.0 版本的技术创新和优化进行了详细解析，希望能帮助用户理解相关算法设计的思路以及具体实现。

未来，我们还会持续完善 YOLOv6 社区生态，同时也欢迎社区同学加入我们，共同建设一个适合工业界应用的更快更准的目标检测框架。

再次附上 YOLOv6 Github 的传送门：https://github.com/meituan/YOLOv6 ，感谢您的 Star 收藏。

4. 作者简介

楚怡、奕非、露露等，均来自美团视觉智能部。

交互式推荐在外卖场景的探索与应用

作者：姬晨 亚成 姜飞等

1. 背景

1.1 什么是交互式推荐?

交互式推荐是一种互动式实时推荐产品模块，主要通过理解用户需求、以互动的方式进行推荐。交互式推荐由 Youtube 在 2018 年提出 [1]，主要用于解决推荐系统的延迟 [2] 和与用户互动偏弱的问题。

从 2021 年下半年开始，美团外卖推荐技术团队在外卖首页 Feed 上持续进行探索，2022 上半年完成全量。具体流程如视频 1 所示：用户从首页 Feed 进入商家详情页并退出之后，动态地插入新的推荐内容到用户推荐列表中。其主要优势是根据用户的实时需求动态插入卡片进行反馈，进而增强用户的使用体验。



视频 1　外卖首页 Feed 中的交互式推荐形态

1.2 为什么需要交互式推荐?

我们发现，外卖首页 Feed 在**用户即时兴趣的捕捉和反馈**上存在痛点，“对比型”用户的选购效率和体验不佳。外卖首页 Feed 作为泛意图用户主要选购场景之一，用户在浏览到成单过程中通常需要进行一番对比、才能逐步收敛意图，然后做出最终决策。

但受限于长列表的翻页模式，首页 Feed 根据用户需求实时调整推荐结果的能力不足。具体表现在，一部分用户的浏览深度不足一页，推荐系统没有额外的机会根据用户兴趣调整推荐结果。另一部分用户虽然有较深的浏览深度，但需要等到翻页时推荐系统才能重新理解用户意图，实时性不足。

业界优化这类问题的主要产品形态有交互式推荐、动态翻页、端上重排这三种。交互式推荐由于是在用户可视范围内插入，用户感知较强；后两种的主流形态是在用户不可见区域更新推荐，用户感知相对较弱。其实，这三种形态在美团外卖均有尝试，本文重点聚焦于交互式推荐的介绍。

2. 问题与挑战

我们在外卖场景搭建交互式推荐时，主要面临以下难点和挑战：

- 不同于传统的推荐系统，交互式推荐是由用户触发的推荐，外卖场景下，如何更好的匹配用户实时需求，搭建出一套适用于外卖的、基于端智能框架的推荐系统是我们首要解决的问题。
- 作为首页 Feed 内部的个性化模块，交互式推荐只做单一模块的优化是不够的，还要考虑首页 Feed 整体的访购效率。那么，如何选择优化目标，以及如何衡量效果和收益，是摆在我们面前的第二个问题。
- 主流的 Feed 形态是双列商品瀑布流，但外卖首页 Feed 是以商家为主的单列列表，如何避免交互在用户的选择路径上带来的“干扰感”，在合适的时机触发交互式推荐，是我们面临的第三个问题。
- 交互式推荐具有动态插入效果，用户对于推荐结果好与坏的感受会更加明显。如何更好理解用户即时意图，如何利用首页 Feed 列表推荐结果优化交互式推荐的单商家卡片，是我们面临的第四个问题。

本文将从以上四个方面，详细介绍外卖首页 Feed 交互式推荐从 0 到 1 搭建的全过程，

以及针对以上问题的解决思路。

3. 主要工作

3.1 交互式推荐框架

3.1.1 整体链路

上文提到，要实现交互式推荐，搭建出一套适用于外卖的、基于端智能框架的推荐系统非常重要。搭建思路可以用“4W1H”来总结：

- **Where/How**：交互式推荐卡片展示在哪？交互式推荐卡片的展现形式是什么？涉及产品形态。
- **Who/When**：交互式推荐需要对什么样的用户触发？在什么时机下触发？涉及用户意图理解。
- **What**：交互式推荐卡片需要展示什么？涉及推荐策略。

基于对上述问题的思考和多方探讨，我们最终和产品、端智能、客户端、应用服务和推荐工程等多个相关团队一起，搭建了这套适用于外卖首页 Feed 的交互式推荐链路。

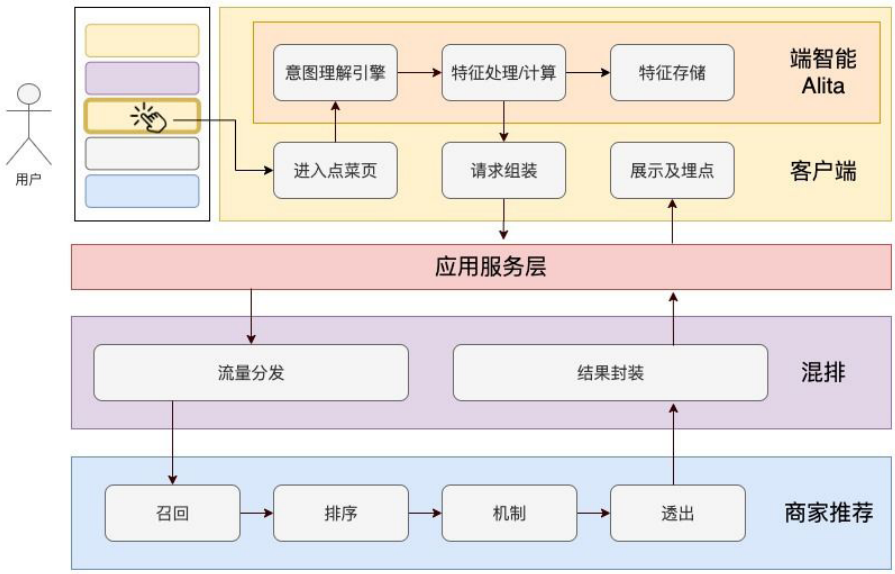


图 1　交互式推荐整体链路

上图 1 展示了从“用户点击首页 Feed 商家卡片”开始，到交互式推荐卡片展现”的全流程。用户进入点菜页后，由客户端调用端智能的意图理解引擎；满足交互式推荐的触发条件后，进行特征处理、计算和存储，并将计算好的将特征传递给客户端组装推荐请求；推荐请求由应用服务层透传给混排服务，再由混排调用商家推荐模块，经过召回、排序、机制、透出阶段，最终返回结果到客户端进行展示。

3.1.2 产品形态

文章开头部分的视频 1 是我们线上的最终形态（在用户点击商家下方插入单个商家卡片），但在此之前，我们对交互式推荐的卡片形态和交互逻辑进行了多轮尝试。

- 在卡片形态上，我们先后探索、上线了搜索词卡片、多商家聚合卡片（如视频 2 所示）、单商家卡片（如视频所示）等多种形态，测试不同卡片类型对用户选购的影响。
- 在交互逻辑上，为了避免插入动画对用户选购的“干扰感”，也对比了“在点击卡片上覆盖”和“在点击卡片下方插入”两种交互，测试对于用户选购的影响。



视频 2　交互式推荐双商家卡片展示样式

在观测不同产品形态的效果差异时，我们重点关注插入的交互式卡片对于首页 Feed 的千人成交额的影响，实验数据见下表：

卡片形态	首页Feed千人成交额	首页Feed UV_CXR
多商家聚合 vs 无交互式推荐	-0.45%	-0.28%
单商家 vs 无交互式推荐	+0.10%	-0.01%

交互逻辑	首页Feed千人成交额	首页Feed UV_CXR
在点击卡片上覆盖 vs 无交互式推荐	-1.35%	-1.09%
在点击卡片下方插入 vs 无交互式推荐	+0.08%	-0.05%

其中，UV_CXR = 交易用户数 / 曝光人数。

在探索过程中，我们也迭代了以下 3 点认知：

- 首先，单列列表中，越原生的卡片（和主流卡片相似度高），对于用户的干扰越小，也更容易被用户接受，从而产生点击、成单等行为。
- 其次，不论是搜索词推荐，还是多商家聚合推荐，尽管看上去外露供给更多，但在转化链路中增加了落地页环节，实际带来的折损会更高（如下图 2 所示）；同时，由于交互式推荐要保证一定的相关性，落地页形态对于供给丰富度的要求更高，但是 LBS（基于位置）推荐中供给相对更少，因而难度也更大。
- 另外，在用户选购过程中，“多店对比”是非常常见的场景，因此，将点击商家覆盖住尽管节约了一个坑位，但带来的负向影响要大于其正向收益。

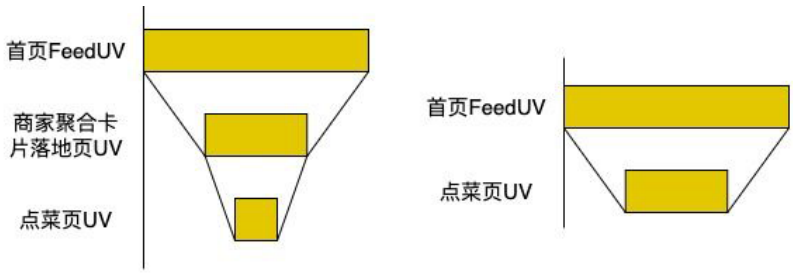


图 2　转化漏斗：多商家聚合卡片转化漏斗（左），单商家卡片转化漏斗（右）

3.2 评估方式和评估指标

交互式推荐的目标是提升首页 Feed 整体访购效率，进而提升用户体验。评估其收益的核心指标，落脚到首页 Feed 整体转化效率。然而，交互式推荐有触发策略约束条件（详见 3.3 节），流量占比也较低，同时和首页 Feed 在用户成单上存在“挤占”效应。当交互式卡片的访购效率有较大幅度提升时，才能撬动首页 Feed 的整体效率。因此，只观测首页 Feed 整体效率，无法指导交互式推荐的日常策略的迭代和效果的分析，需要更直接、置信的指标衡量“什么是好的交互式推荐算法”。

我们评估交互式推荐算法，主要考虑两个维度：

- 插入卡片的覆盖面
- 插入卡片的匹配度

为了应对以上两个维度的评估，我们分别引入“曝光页面 Page 占比”、“同位置订单增量”来衡量交互式推荐对首页 Feed 曝光的影响。

评估覆盖面，常见指标是曝光量、曝光占比。但交互式推荐卡片插入后会改变首页 Feed 曝光量，直接计算其在首页 Feed 的曝光占比不合理。因此，我们将统计曝光的维度从“量”改为“页面”，通过计算交互式卡片曝光 Page 数在首页曝光 Page 数的占比来评估卡片覆盖面。

观测 Page 占比（下文简称“曝光页面 Page 占比”）便于评估交互式推荐的覆盖面和其理论上限的差异，如果用户在首页 Feed 的每次点击商家都触发并展示交互式商家卡片，曝光 Page 占比和首页 Feed 的 Page CTR（曝光点击率）等同。因此，结合首页 Feed 的 Page CTR，我们就可以观察到交互式推荐覆盖面和其理论上限的差距，便于继续优化逼近上限。曝光页面 page 占比对同一页面多次触发不敏感，因此，我们引入了交互式卡片曝光 PV 占比（交互式卡片曝光量占首页 Feed 曝光量的比值）、曝光 UV（曝光交互式卡片的用户占首页 Feed 曝光用户的比值）占比辅助观测对首页 Feed 的影响。

评估匹配度，常见指标是曝光到成单的转化率。我们认为，交互式推荐插入的商家应该比上下文中的其他商家更符合用户当前的兴趣。因此，最直观的指标就是对比交互式卡片和首页 Feed 其他卡片的转化率，但这两类卡片对比，会存在三种偏差：

- **人群偏差：**交互式推荐曝光人群是产生过点击的用户，这部分用户群体的转化率天然高于“大盘”。
- **位置偏差：**交互式推荐卡片由用户点击触发，由于排在前面的商家往往更容易被点击，因此交互式卡片的曝光位置相对更加靠前。
- **资源类型偏差：**首页 Feed 列表涉及较多主题、广告等异构流量，大部分异构流量往往并不是从转化率出发进行排序。

上述 3 个偏差，使得交互式卡片转化率天然比首页 Feed 其他卡片转化率高，因此简单比较交互式卡片和首页 Feed 正常卡片之间的差异并不能正确评估交互式推荐本身的价值。而交互式推荐的产品特性是将原来曝光在首页 Feed 的卡片依次往后挤压，因此只有当交互式推荐卡片的转化率高于原本该位置的卡片（也就是后来被挤压到交互式卡片的下一位的卡片）的时候，交互式推荐才能产生正面效果。

基于此，我们使用“对比同一次请求内，与下一位自然商家预估转化率的相对差异”（下文简称相对下一位差异）来衡量推荐卡片的匹配度，“同请求”解决了人群偏差问题，“下一位”缓解位置偏差问题，“自然商家”解决了资源类型偏差问题。

此外，扩大覆盖面通常会引起匹配度的下降，为了平衡这两个指标，我们引入“相对下一位差异乘以交互式卡片曝光量”作为策略迭代的辅助观测指标，其物理意义是插入交互式卡片并将原来的卡片往下挤压之后，在该位置产生的预期订单数的增量（下文简称“同位置订单增量”）。

3.3 用户意图理解

交互式推荐，由推荐系统感知到用户的“交互”触发。其理解用户意图的流程主要包含两个阶段：1）用户对推荐系统的哪些行为可以触发交互式推荐；2）触发交互式推荐时用户的即时意图是什么样的。下文将会围绕这两部分展开说明。

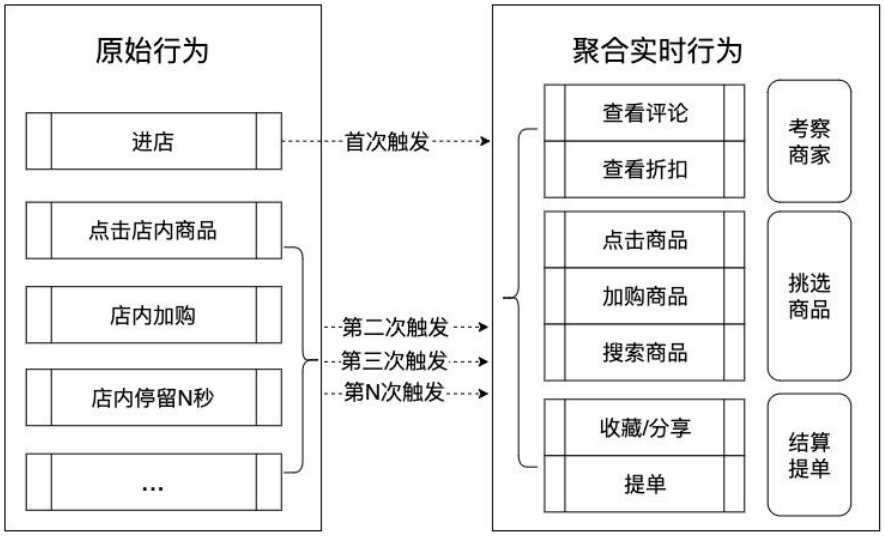


图 3　用户意图理解引擎

3.3.1 首次触发策略

为了探索不同触发时机对交互式卡片“相对下一位差异”、“曝光页面 Page 占比”等指标的影响，我们根据用户的不同需求，尝试了加购物车、点击菜品、停留时长（10s/5s/2s/ 进店立即）触发等时机。实验证明，放宽触发条件会增大交互式推荐不置信触发的风险，导致交互式卡片效率下降，但会带来交互式卡片曝光量的提升，覆盖更多的用户，有利于后续策略迭代。最终，我们采用“用户从首页 Feed 进店后立即触发”的首次触发策略。

3.3.2 连续触发策略

由于用户在商家详情页的停留时长无法确定，且工程上无法实现在用户返回列表的瞬间请求推荐服务并展示结果，因此，在用户浏览商家详情页期间，系统需要连续多次向服务端请求推荐结果，用户在商家详情页的停留时长越长、需求越明确，服务端的推荐结果也就越精准。所以，我们采用“连续触发策略”，即随着用户在店内停留时间的增加，或者用户产生新的菜品点击、加购特征后，客户端会连续多次向后端服务发起请求，更新推荐结果。

3.3.3 用户实时需求理解

如何通过端智能 [3–4] 更好理解用户意图是我们关注的重点，相比服务端，用户在端上的特征主要有以下 2 个特点：

- **实时性更好：**从“准实时”到“超实时”的交互。
- **维度更细：**从“交互 Item”进化到“Item 交互的 Micro– 粒度”。

因此，借助端智能的能力，我们不再受限于首页 Feed 的分页请求更新机制，能够根据用户行为，更好理解用户需求，实时智能决策更新推荐结果，缓解反馈信号感知延迟问题。

用户点击商家卡片后在店内的主要行为可以帮助我们更好理解用户实时需求。图 4(a) 展示了部分店内行为，图 4(b) 分析了部分不同的行为对比查看商家介绍行为，在用户当日完单率（当日完单率定义：当日用户在商家内发生某一行为，并在该自然日内外卖有成单）上的差异，说明不同行为下用户的需求有明显差异。

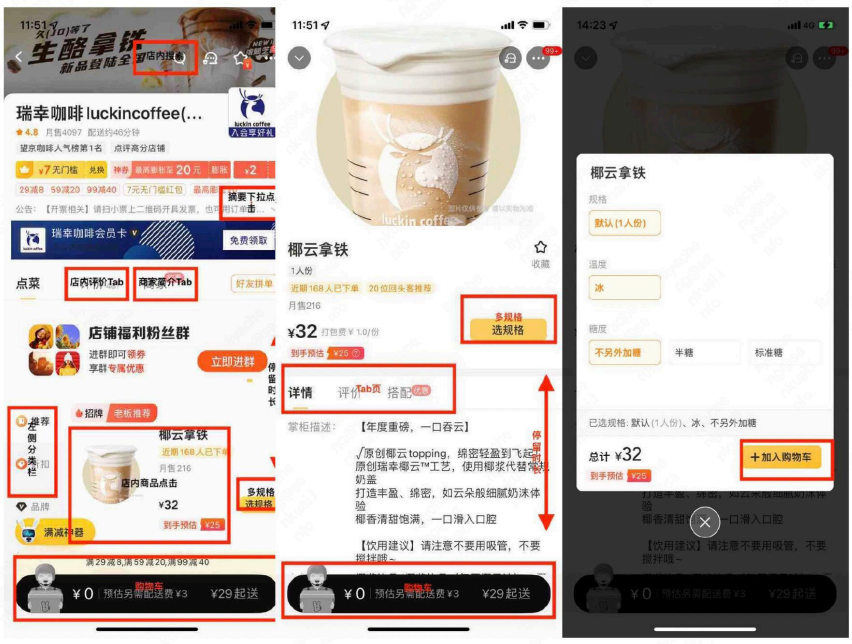


图 4(a)　用户店内主要行为

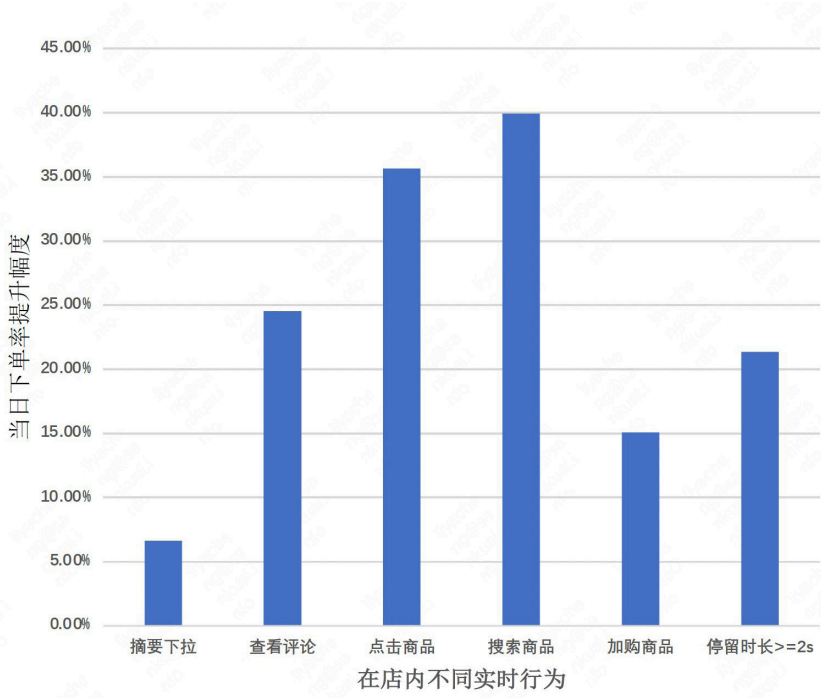


图 4(b)　用户店内主要行为对比“查看评论”当日下单率差异

3.4 推荐排序策略

首页 Feed 一次展示整个列表，而交互式推荐一次仅展示 1 个商家卡片。为了推荐更精准的结果，就需要交互式推荐更准确的理解用户实时的外卖需求。因此，我们在首页 Feed 的推荐链路的基础上，通过优化“召回 -> 排序 -> 机制 -> 透出”链路，覆盖更多用户的同时，不断提升交互式推荐和用户兴趣的匹配度。

3.4.1 召回

召回阶段分为两步（如下图 5 所示）：

- 利用多个召回算法策略从用户附近召回数百个候选 POI 商家。
- 利用相似品类过滤方案来过滤和当前用户意图明显不同的商家，并将生成的候选结果传给排序阶段。为更好理解用户即时意图，我们提出了 Item2Item Multi-Trigger 召回和相似品类过滤方案。

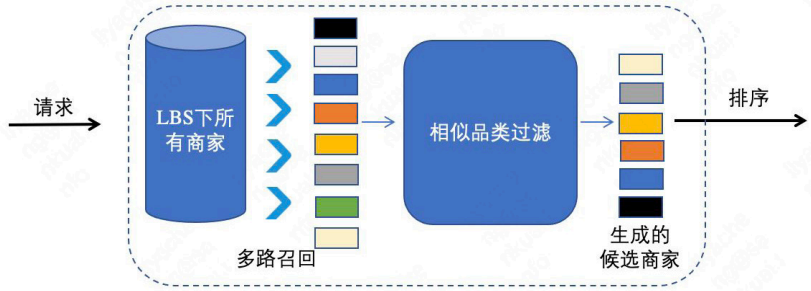


图 5　召回阶段流程图

首先，一方面，我们直接复用了首页 Feed 推荐的召回链路，融合了双塔召回 [5]、User2Item 召回 [6]、热销召回等多个召回算法。另一方面，为了加强对用户即时意图的理解和关注，我们新增了一路 Item2Item Multi-Trigger 旁路召回。

具体做法是：我们将用户在首页 Feed 点击、加购的 POI，作为 Item2Item 召回的 Trigger，召回更多符合用户即时意图的商家。其中每个用户的 Trigger 数是不同的，每个 Trigger 召回的商家数量也是不同的，其数量满足 N/M（N 是 I2I Multi-Trigger 召回的 POI 总数，M 是 Trigger 数量）。

其次，用户最近点击的商家能够帮我们更好理解用户即时意图。我们考虑到外卖场景“对比型”用户居多的特点，为了给用户带来更好的体验，提出了“同叶子品类”策略：约束了曝光的交互式卡片商家，必须和触发商家的叶子品类（体现的是商家口味，和主营菜品有关，比如串串香、鸡肉卷）相同。但是该方案会带来 2 个问题：

- LBS 约束下同叶子品类商家较少，导致交互式卡片曝光量少。
- 外卖场景下商家品类定义的粗细（每个商家包含由粗到细的多个级别的品类）粒度不同，交互式推荐需要一套统一的品类定义方式，既要保证推荐商家品类与用户兴趣的高相关性，又需要推荐结果有一定的多样性。

因此，我们参考了现有商家品类定义，考虑了商家口味、消费者相似性以及商家包含的商品品类分布等维度，通过聚类的方法重新定义了交互式推荐的相似品类。具体来说，我们将 200 个左右的细粒度的品类，定义为 70 个左右的粗粒度品类，在满足“对比型”用户诉求的同时，也为更多用户带来了新颖性、多样性的体验。

我们提出的策略，在明显提升交互式卡片曝光页面 Page 占比的同时，对比下一位差异

明显提升，具体效果可以见下表：

方法	交互式推荐同位置订单增量	对比下一位差异	交互式卡片曝光页面Page占比	交互式卡片曝光PV占比	交互式卡片曝光UV占比
Item2Item Multi-Trigger召回	+6.10%	+17.50%	+5.70%	-1.60%	-0.24%
相似品类过滤	+10.25%	+4.72%	+12.09%	+15.98%	+8.16%

3.4.2 排序

排序阶段，模型的主要任务是预测 CTR、CXR（曝光转化率），并将预估结果传给机制阶段。

优化交互式推荐的排序模型，我们主要面临样本分布差异、训练样本少的问题。交互式推荐的单商家卡片形态和首页 Feed 的列表形态天然有差异，导致其样本分布上（如点击率、转化率、人群分布上）明显不一致，直接使用首页 Feed 的推荐模型缺少对交互式推荐场景个性化的关注，效果会有明显衰减。简单的做法是直接使用交互式推荐的样本训练模型，但是交互式推荐单场景样本较少，会导致模型鲁棒性不足。

因此，我们选择了业界通用的 Fine-tune 的方法，在首页 Feed 排序模型的基础上，利用交互式推荐的样本 Fine-tune 排序模型。同时，我们充分利用了 3.3.3 节构建的用户实时需求理解模块，优化模型效果。当然，我们也探索了不同网络结构对于模型效果的提升，但是限于算力资源等约束，我们并没有上线更为复杂的交互式推荐排序模型，具体的模型结构如下图 6 所示。

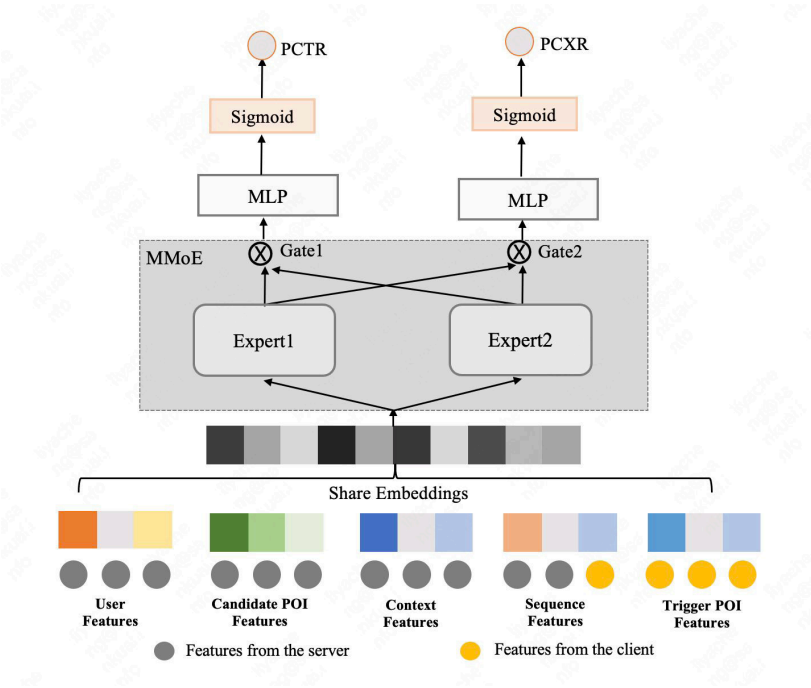


图 6　模型结构图

输入模型的 Embdding 数据，经过 MMoE[7] 层和 3 层的 MLP 网络，得到预测的 pCTR、pCXR 结果。其中，模型的输入特征分为 5 种类型：1）用户特征；2）商家特征；3）上下文特征；4）序列特征；5）触发交互式推荐的商家特征。序列特征，包含了实时曝光、点击等序列，并且使用了 Micro-Behavior[8] 详细信息。

触发交互式推荐的商家特征，包含商家的 embdding 表示、配送信息、优惠信息等，排序模型离 / 在线效果如下表所示。可以看出，对比首页 Feed 排序模型，优化后的交互式推荐排序在卡片效率上显著优于直接使用首页 Feed 排序模型的效果。

模型	CXR AUC	CXR GAUC[9]	交互式推荐同位置订单增量	对比下一位差异	交互式卡片曝光页面Page占比	交互式卡片曝光PV占比	交互式卡片曝光UV占比
+Fine-tune vs 首页Feed排序模型	+1.66pp	+1.91pp	+22.79%	+44.68%	-1.95%	+2.20%	-2.52%
+Fine-tune&& 实时需求理解 vs 首页Feed排序模型	+1.79pp	+2.02pp	+24.09%	+51.92%	-3.10%	+2.92%	-2.41%

3.4.3 机制

为了更为灵活的承载业务目标，我们引入了机制模块，目标是对排序阶段传过来的候选商家，根据不同业务目标进行顺序调整（如 CTR、CXR、新颖性等）。最终，在兼顾新颖性的目标下，交互式推荐按照预估 CXR 排序，以最大化卡片和列表转化率。与此同时，我们还从解决负面反馈、优化体验两方面进行了体验优化：

- 通过业务规则约束来解决负面反馈，主要包括：重复曝光商家过滤、预下单商家过滤、同品牌商家过滤、用户不喜欢 & 黑名单商家过滤，以及避免插入配送费过高、配送距离过远的商家。
- 在交互体验方面：1) 通过在新颖性等目标上的探索，给用户更丰富的推荐体验；2）通过推荐理由的优化，为用户解释推荐原因。

3.4.4 透出

透出阶段，主要判断机制阶段传过来的 Top 1 的商家是否展示给用户。理论上，用户每次“触发”交互式推荐，系统都有可能推荐一个新的商家进行展示。但是，不考虑商家质量的推荐策略对用户体验和首页 Feed 效果都有很大损伤。因此，我们探索了卡片的透出策略，即机制阶段透传的 Top 1 卡片是否展示。

如下图 7 所示，商家展示区域划分为 ABCD 四个区域：交互式推荐插卡位置（A）、触发商家（B）、触发商家上文（C）、触发商家下文（D）。交互式卡片插入后，D 区域第一位商家下滑，动画效果会将用户的注意力吸引到交互式卡片 A 上。但是，用户是否在交互式推荐的商家 A 中成单，不仅和它是否满足用户偏好相关，还离不开和上下文商家 C、B、D 的对比效果——至少，A 应该比 C、B、D 区域的商家更符合用户当前意图。

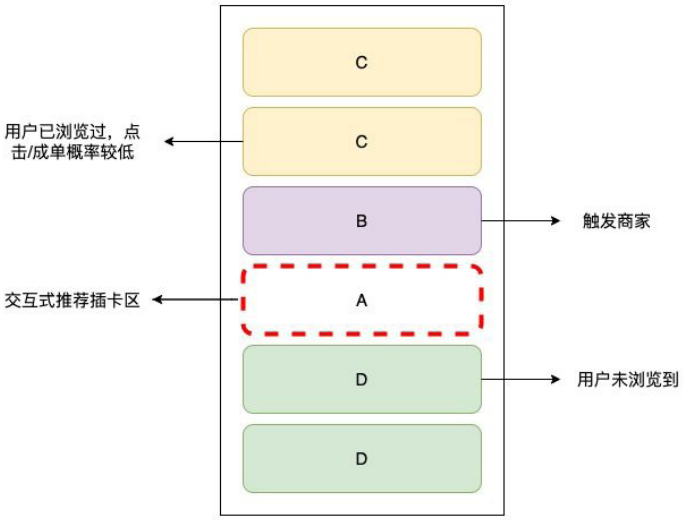


图 7　商家展示区域划分

我们更关注首页 Feed 的转化率，因此，当交互式推荐商家比同上下文的其他商家的 pCXR 更高时，这个商家才应该被曝光，其形式是：

$$\mathbf{pCXR}_{\text{交互式推荐商家}} > \alpha * \mathbf{pCXR}_{\text{下一位商家}}$$

随后，有几个问题：对比 B、C、D 区域哪些商家？交互式推荐商家 A 的 pCXR 要比对比商家高多少（通过公式 1 比例系数 α 控制）才应该被展示？对于后者，我们通过实验得出；对于前者，我们分析如下：

- C 区域：用户已经浏览过，点击 / 成单的可能性更低。显然，对比该区域商家意义不大。
- B 区域：用户“最近一次点击”&&“触发交互式推荐”的商家，用户对该商家很感兴趣。与其比较看似有明确意义，但该商家能曝光，说明是同类 / 相似商家中的佼佼者，在排序模型 / 特征不做改动的前提下，难以找到 pCXR 更好的商家。
- D 区域：用户未浏览到，交互式商家卡片 A 一旦展示，其插卡的动态效果同时会使得用户更注意该区域。因此，与这部分商家做对比更符合直觉。

由于限制了卡片的透出条件，交互式卡片曝光量明显下降。实验证明，当交互式推荐

卡片对比下 1 位商家卡片的 pCXR 更高时，其同位置订单增量最高，曝光页面 Page 占比损失最少，策略最优。我们采用该方案。从实验数据看出，对比下 N 位 pCXR 均值时，随着 N 取值的变化，会影响交互式卡片的曝光量和效率，其效果与直接调整 pCXR 的过滤阈值 α 等同。在实际生产环境中，选取“交互式推荐同位置订单增量”较高时的参数 α 即可，这里我们取 1。

方案	交互式推荐同位置订单增量	对比下一位差异	交互式卡片曝光页面Page占比	交互式卡片曝光PV占比	交互式卡片曝光UV占比
比**下1位**poi预估pCXR高时透出	+573%	+69.06%	-43.89%	-51.59%	-36.82%
比**下3位**poi预估pCXR高时透出	+497%	+75.40%	-52.70%	-60.19%	-45.43%
比**下20位**poi预估pCXR高时透出	+406%	+62.82%	-52.49%	-59.76%	-44.90%

4. 总结与展望

本文介绍了我们在首页 Feed 交互式推荐探索尝试，其中主要包括：

- 依托端智能的能力，结合外卖场景下的用户“多店对比”的选购特点，搭建了“动态插入单卡商家”的交互式推荐系统。
- 充分考虑插入卡片对首页 Feed 上下文的影响，提出了“同位置订单增量”等指标，从匹配度、覆盖面两方面构建了“什么是好的交互式推荐系统”的评估方式。
- 从业务理解、用户需求建模等方面出发，通过优化“触发 -> 召回 -> 排序 -> 机制 -> 透出”链路，提升系统对用户意图理解的准度，优化用户体验。

目前，交互式推荐已在首页 Feed 全量，我们也收获了如下的业务收益：

- 首页 Feed 千人成交额 +0.43%，首页 Feed 曝光新颖性 +1.16%。
- 交互式卡片相对下一位自然商家的转化率 +132%。

未来，我们将从以下方向进行探索优化：

- **优化产品形态：**从定制推荐理由、触发时机等多个角度继续优化交互式推荐的产品功能，并将动态化推荐能力扩展到外卖其他场景。
- **承载更多业务目标：**在满足用户精准推荐的前提下，综合新颖性、多样性等多种

差异化的业务目标，进行建模优化。

- **扩大端智能优势：**现有的交互式推荐系统将特征处理、召回、排序、机制等流程都放在服务端完成，但是服务端 <-> 云端的性能限制了更多信息的加工利用，未来可以放到端上完成训练、预估，在实现“千人千模”的极致个性化体验的同时，有效保护用户的隐私。与此同时，我们可以借助端智能的优势，探索端上重排序的方案。

5. 本文作者

| 姬晨、亚成、王炜、成龙、姜飞、王聪、北海等，来自到家事业群 / 到家研发平台 / 搜索推荐技术部。
| 姝阳、张婧等，来自到家事业群 / 外卖事业部 / 产品部。

6. 参考文献

[1] Christakopoulou K, Beutel A, Li R, et al. Q&R: A two-stage approach toward interactive recommendation[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 139-148.

[2] Xinran He, Junfeng Pan, et al. Practical lessons from predicting clicks on ads at facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising. ACM, 1 - 9, 2014.

[3] Gong Y, Jiang Z, Feng Y, et al. EdgeRec: recommender system on edge in Mobile Taobao[C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020: 2477-2484.

[4] 端智能在大众点评搜索重排序的应用实践.

[5] 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 ’13). Association for Computing Machinery, New York, NY, USA, 2333 - 2338.

[6] 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 Comput. Surv. 47, 1, Article 3 (may 2014), 45 pages.

[7] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 1930 - 1939.

[8] 美团外卖推荐情境化智能流量分发的实践与探索.

[9] Zhou G, Zhu X, Song C, et al. Deep interest network for click-through rate prediction[C]// Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 1059-1068.

招聘信息

搜索推荐算法工程师

1. 发展大规模深度学习、图学习等技术，利用注意力机制、记忆网络、关系网络等模块，从跨多个时空场景的海量数据中理解用户需求、发掘用户兴趣，优化点击率、转化率模型，为用户展示更合适、更感兴趣的美食和商品。
2. 发展强化学习、可解释深度学习、多模态学习、多目标优化等技术，优化重排、混排模型，智能调控流量分发，优化平台生态，实现消费者和商家共赢。
3. 使用知识图谱、计算视觉、自然语言生成等技术，面向用户兴趣，帮助商家自动化智能生成展示内容和文案，提升推广效率。
4. 追踪并研究人工智能前沿技术，探索技术在零售、医疗电商场景的应用。

感兴趣的同学可以将简历发送至：libeihai02@meituan.com。期待与大家一起共事，共创未来。

如何利用「深度上下文兴趣网络」提升点击率?

作者：旭阳 浩然 刘奇 景文 王哲

点击率（CTR）预测是在线广告、推荐系统等领域的一个基础任务，主要目标是预测用户点击某个 Item 的概率。用户行为序列中蕴含着丰富的用户兴趣信息，这对于 CTR 预测至关重要。然而，大部分用户行为序列模型只从用户的点击行为中建模用户的正向兴趣，而忽视了上下文信息，即点击 Item 周围的其他 Item，导致效果受限。

我们认为上下文信息在用户行为序列建模中非常重要，因此提出了一种名为深度上下文兴趣网络（DCIN）的新模型，可以对点击及点击周围的展示上下文进行统一建模，以学习用户的上下文感知兴趣。DCIN 由三个关键模块组成：1）位次感知的上下文聚合模块（PCAM），通过注意力机制对展示 Item 进行聚合；2）反馈 – 上下文融合模块（FCFM），通过非线性特征交互，融合点击和展示上下文的表征；3）兴趣匹配模块（IMM），激活与目标 Item 相关的兴趣。

此外，我们还对线上性能进行了针对性优化，使得 DCIN 模型可以在大规模的工业广告系统上部署。我们提出的 DCIN 方法在离线和线上实验中都取得了显著的效果提升。目前已经在我们美团线上广告系统落地并全量，带来了 1.5% 的 CTR 提升和 1.5% 的 RPM 提升。

1. 背景

点击率（CTR）预测任务的目标是预测用户对某个 Item 的点击概率，这对于在线广告和推荐系统等领域至关重要 [5, 16]。这里我们主要关注 CTR 预测中的用户行为序列建模问题。用户行为序列建模的目标是挖掘用户行为序列中蕴含的丰富的用户兴趣，更准确地理解用户的兴趣和偏好，从而提供更个性化的推荐服务。许多工作 [7, 11, 13, 18, 21, 22] 从用户的点击行为中捕捉用户的兴趣，大大提高了 CTR 模型的效果。

然而，一些研究 [13, 15] 发现用户的点击行为并不能完整地刻画用户兴趣，导致仅从用户点击行为中得到的用户表征有偏。为了解决这个问题，DFN[17]、DUMN[1] 等一些开创性的工作引入了用户的未点击行为来捕捉用户的负面偏好（如不喜欢），取得了不错的

效果。不过这些方法虽然引入了未点击行为，但它们将用户的点击和未点击行为分开建模，忽视了点击和未点击 Item 之间的交互。通常，智能手机的屏幕可以同时展示许多 Item。用户的点击行为不仅由目标 Item 与用户兴趣之间的匹配程度决定，而且还受到展示上下文的影响 [2, 12, 20]。

如下图 1 所示，当用户想买一件 T 恤时，如果周围的 Item 是智能手机、鞋子或吹风机，他可能会点击绿色 T 恤，因为这些 Item 对他来说很无聊。但是，当周围的展示 Item 都是 T 恤时，他可能会点击蓝色的那一个，因为蓝色 T 恤比绿色 T 恤更符合他的兴趣。基于上述分析，我们认为展示上下文在用户兴趣建模中也是至关重要的。



图 1　展示上下文示例，用户的点击行为可能会受到其他展示 Item 的影响

为了从用户的点击行为和相应的展示上下文中提取出具有上下文感知的用户兴趣表征，我们提出了深度上下文兴趣网络（DCIN）。具体来说，我们首先设计了一个位次感知的上下文聚合模块（PCAM），该模块通过注意力机制对展示 Item 进行聚合。同时，我们考虑到位次偏差 [9, 10]，并将位次信息集成到 PCAM 中。

其次，我们通过反馈 – 上下文融合模块（FCFM），利用非线性特征交互将点击和相应的展示上下文的表征融合在一起，以捕捉用户的上下文相关兴趣。最后，我们使用兴趣匹配模块（IMM）来匹配与目标 Item 相关的行为。此外，我们还设计了相应的线上落地方案，使得 DCIN 模型可以在大规模的工业系统上部署。具体来说，我们的贡献如下：

- 我们强调了引入上下文信息进行用户兴趣建模的重要性。上下文信息可以帮助我

们更准确、全面地捕捉用户的兴趣；

- 我们创新性地设计了序列模型 DCIN，它有效地集成了展示上下文和位次信息，以获取用户的上下文感知兴趣；
- 我们进行了广泛的线下和线上实验，结果都证明了 DCIN 的优越性。目前 DCIN 已经在我们的在线广告系统中全量，并带来了 1.5% 的点击率（CTR）提升和 1.5% 的每千次展示收入（RPM）的提升。

2. 深度上下文兴趣网络

2.1 CTR 模型介绍

点击率预估是在给定用户行为C（点击行为）和D（浏览行为），以及上下文c的情况下，使用模型F预测用户u点击目标Item t的概率p，公式为$p = F(C, D, u, t, c)$。用户的历史点击行为是一个包含对应特征（如Item id、类别等）的点击项目的顺序列表，即$C = \{c_1, c_2, \ldots, c_N\}$，其中$c_i$是第$i$个点击Item，$N$表示点击序列的长度。

为了建模展示上下文，我们为每个点击选择M个对应的展示Item，因此用户的浏览行为D可以表示为：$D = [d_{11}, d_{12}, \ldots, d_{1M}], \ldots, [d_{N1}, d_{N2}, \ldots, d_{NM}]$，其中$d_{ij}$是点击$c_i$周围的第$j$个展示Item。在为每次点击选择周围的展示Item时，我们试图将点击保持在中间。我们用p_i和q_{ij}分别表示c_i和d_{ij}的绝对位次。我们还计算c_i和d_{ij}之间的相对位次$r_{ij} = p_i - q_{ij}$。我们利用广泛使用的Embedding技术将稀疏特征转化为低维度的稠密向量。例如，e_{c_i}代表点击Item c_i的Embedding。

2.2 深度上下文兴趣网络

DCIN 的架构如下图 2 所示，主要包含三个模块。具体来说，我们首先设计了位次感知的上下文聚合模块（PCAM），通过注意力机制来聚合展示上下文。然后，我们通过反馈 – 上下文融合模块（FCFM），利用非线性特征交互融合点击和相应的展示上下文，以获取上下文感知的兴趣表征。最后，我们使用兴趣匹配模块（IMM），自适应地学习与目标 Item 相关的用户兴趣。

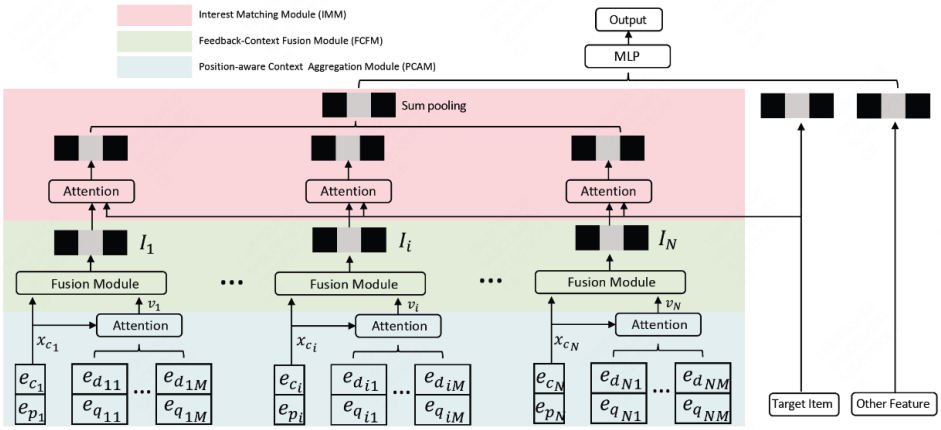


图 2　DCIN 整体框架

2.2.1 深度上下文兴趣网络

如图 1 所示，智能手机一屏通常会展示很多 Item。用户的点击行为不仅由 Item 与他们的兴趣之间的匹配程度决定，也受到展示上下文的影响 [2, 20]。然而，仅从他们的点击行为中建模用户的兴趣会忽略上下文信息，导致提取的兴趣表征有偏。为了捕获用户的上下文感知兴趣，我们设计了 PCAM，它通过注意力机制为每次点击聚合展示上下文。

此外，用户通常倾向于点击屏幕上的位次靠前的 Item（称为位次偏差 [9, 10]），而不考虑相关性。因此，Item 的位次同样影响用户的点击行为。我们还将位次信息集成到 PCAM 中，以更好地理解用户的实际兴趣。因此，PCAM 可以被公式化如下：

$$v_i = \sum_{j=1}^{M} \mu_{ij} W_V e_{d_{ij}},$$

其中，v_i代表点击c_i对应的展示上下文，表示为展示Item $e_{d_{ij}}$的加权总和，W_V是学习参数。μ_{ij}是注意力权重，可以被表示为：

$$\mu_{ij} = \frac{exp(\alpha_{ij})}{\sum_{j=1}^{M} exp(\alpha_{ij})},$$
$$\alpha_{ij} = W_Q x_{c_i} \cdot W_K x_{d_{ij}} + e_{r_{ij}},$$

其中，α_{ij}是第i次点击与其第j个展示Item之间的相关性，$x_{c_i} = concat(e_{c_i}, e_{p_i})$代表项目嵌入$e_{c_i}$和其绝对位次$e_{p_i}$的拼接，同理$x_{d_{ij}}$。$e_{r_{ij}}$是相对位次$p_{ij}$的一维嵌入向量，$W_Q$和$W_K$是可学习参数。

2.2.2 反馈 – 上下文融合模块

以前的用户序列模型 [7, 11, 13, 18, 21, 22] 直接将用户的点击 Item 视为用户的兴趣，忽略了上下文信息。为了捕捉用户的上下文感知兴趣，我们提出了 FCFM 来通过非线性特征交互融合点击和相应展示上下文的表征。具体来说，FCFM 是通过一个 2 层的 MLP（多层感知器）来实现的：

$$f_i^{(k+1)} = ReLU(W^{(k)} f^{(k)} + b^{(k)}),$$

其中，$f_i^{(k)}$是第k层的输出，$W^{(k)}$和$b^{(k)}$是学习参数。其中输入的形式如下：

$$f_i^{(0)} = concat(x_{c_i}, v_i, x_{c_i} - v_i, x_{c_i} * v_i)$$

其中，$*$表示元素间的乘积。然后，我们从点击c_i及其相应的展示上下文v_i中获得了上下文感知兴趣$I_i = f_i^2$。为了实现泛化和可训练性，我们在不同的点击中共享可学习的参数。

2.2.3 兴趣匹配模块

通过前两个模块，我们整体地建模了点击Item和其展示上下文，以捕获用户的上下文感知兴趣，表示为$I = I_1, I_2, \ldots, I_N$，这比仅从点击行为中提取的用户兴趣更精确和全面。为了捕获与目标Item相关的用户兴趣，我们可以轻松集成最新的序列模型，如DIN[22]，DIEN[21]，BST[3]等。在这里，我们选择了DIN方法来聚合用户兴趣，可以表示为：

$$U = \sum_{i=1}^{N} \frac{exp(e_t \cdot I_i)}{\sum_{j=1}^{N} exp(e_t \cdot I_j)} \cdot I_i,$$

其中，U是最终的用户表示，它将与其他特征拼接后输入到MLP中进行CTR预估。我们使用广泛使用的二分类交叉熵损失函数来训练我们的DCIN模型。

2.3 线上服务优化

对于工业广告系统来说，服务延迟的限制非常严格。由于我们引入了展示 Item 来捕获用户的上下文感知兴趣，序列长度被扩展了数十倍，而在线模型的服务延迟通常与序列长度成正比，因此直接在工业在线系统中使用 DCIN 会带来服务延迟的重大负担。为了解决这个问题，我们针对性地进行了一些在线优化。

具体来说，由于PCAM和FCFM与目标Item无关，我们可以提前计算用户的上下文感知兴趣表示$I = I_1, I_2, \ldots, I_N$，在线服务时直接将其作为模型的输入。这节省了大量的在线服务时间（约10毫秒），使DCIN可以处理的用户行为序列最大长度比RACP [6]（引入了页面反馈建模的SOTA模型）长28倍，并且仅增加约1毫秒的在线延迟。

3. 实验

3.1 实验设置

3.1.1 数据集

将展示上下文和位次信息集成到序列模型中是一种全新的探索，目前缺乏相关的公开数据集。因此，我们从在线广告系统中构建了一个 31 天的工业数据集，其中包含数十亿的样本。按照惯例，我们将前 30 天的数据划分为训练集，最后一天的数据划分为测试集，并保留用户最近的 50 次点击行为，每次点击的周围 20 个展示 Item 被用来建模展示上下文。

3.1.2 基线模型

我们选取六种序列模型与 DCIN 进行比较。所有模型都使用相同的特征进行公平比较：

- Wide&Deep[4] 由一个 Wide 部分和一个 Deep 部分组成，用于学习原始和高阶特征交互；
- DeepFM[8] 用 FM 层替换了 Wide&Deep 中的 Wide 部分；
- DIN[22] 引入了一种注意力机制，用于捕获用户对不同目标 Item 的多样化兴趣；
- DIEN[21] 提出了一个基于门控循环单元（GRU）的兴趣演化层，用于模拟用户的动态兴趣；
- DFN[17] 试图从点击行为中学习正向偏好，从未点击行为中学习负向偏好；
- RACP[6] 引入了页面反馈建模，是已发布的用户行为建模的 SOTA 方案。

3.1.3 评估指标

我们在 CTR 预测任务中使用了广泛采用的评估指标 AUC，我们进一步引入 RelaImpr[19] 来衡量相对于基础模型的改进：

$$RelaImpr = \frac{AUC(measured\ model) - 0.5}{AUC(base\ model) - 0.5} - 1$$

3.2 实验结果

我们重复所有实验 5 次，并报告他们的平均结果。在实验中，每个特征的嵌入维度为

16。FCFM 中 MLP 的隐藏大小和输出层分别为 64/16，和 1024/512/128。我们使用学习率为 1e−4 的 Adam，Batch Size 大小为 3000。

3.2.1 离线效果

实验结果如下表所示，从中我们可以发现，我们提出的 DCIN 在 AUC 上显著优于所有方法，并在基础模型上实现了 21.24% 的 RelaImpr。如 2.3 节所介绍，DCIN 可以处理的最长用户行为序列比 RACP 长 28 倍。为了公平比较，DCIN(Short) 使用与 RACP 相同的序列长度，而 DCIN 使用我们在线广告系统中实现的更长序列。

具体来说，首先 DIN 和 DIEN 比 Wide&Deep 和 DeepFM 表现得更好，因为它们捕获了用户对不同目标 Item 的多样化和时变性兴趣。其次，DFN 略微优于 DIN 和 DIEN，因为它试图从未点击行为中捕获不喜欢的信号。第三，RACP 比 DFN 表现得更好，因为它模拟了用户的上下文化页面兴趣。第四，我们提出的 DCIN(Short) 优于 RACP，这出于两方面原因。一方面，RACP 只模拟页面兴趣，忽略了同一页面上的不同点击也可能表明不同的兴趣，降低了模型捕获用户细粒度兴趣的能力。另一方面，RACP 忽略了 Item 位次对用户点击行为的影响。最后，DCIN 优于 DCIN(Short)，因为它引入了更长的序列。

模型	AUC	RelaImpr
Wide&Deep	0.63588	0.00%
DeepFM	0.63602	0.10%
DIN	0.65012	10.48%
DIEN	0.65102	11.14%
DFN	0.65220	12.01%
RACP	0.65872	16.81%
DCIN(Short)	0.66186	19.12%
DCIN*	**0.66475**	**21.24%**
DCIN(short)−w/o postition	0.65850	16.64%
DCIN(short)−w/o FCFM	0.65950	17.38%

3.2.2 消融实验

我们还进行了消融测试，以验证 DCIN 中不同组件的有效性和必要性。首先，我们在 PCAM 中去除位次信息，并进行 DCIN(short)−w/o postition 的实验。如表 1 所示，DCIN(short)−w/o postition 的表现比 DCIN(Short) 差，因为 Item 的位次对用户的点击行为也有重要影响。

其次，我们在DCIN中去除了FCFM模块，并进行了DCIN(short)-w/o FCFM的实验，其中我们直接将展示上下文v_i视为用户的兴趣，可以公式化为$I_i = v_i$而不是$I_i = f_i^2$。如表1所示，由于缺乏点击和展示上下文之间的融合，DCIN(short)-w/o FCFM的表现也比DCIN(Short)差。

3.2.3 线上 A/B 测试

我们进行了严格的在线 A/B 测试，在线上对 DCIN 进行了两周的评估。与 DIN（线上基线）相比，DCIN 实现了 CTR+1.5%、RPM+1.5% 的提升。目前，DCIN 已经在线部署并全量，显著提升了广告收入。

3.2.4 Case 研究

为了展示 DCIN 在上下文感知兴趣表示方面的有效性，我们随机选择 10 个点击 Item，并分别为每个点击选择 100 个不同的展示上下文。然后我们提取相应的上下文感知兴趣 $IiIi$，并通过 t-SNE [14] 可视化它们的分布。如图 3 所示，点击 Item 之间的明显分离表明了学习中点击行为的主导地位。同一点击的不同上下文的明显多样性证明了 DCIN 捕获细粒度上下文感知兴趣表示的能力。

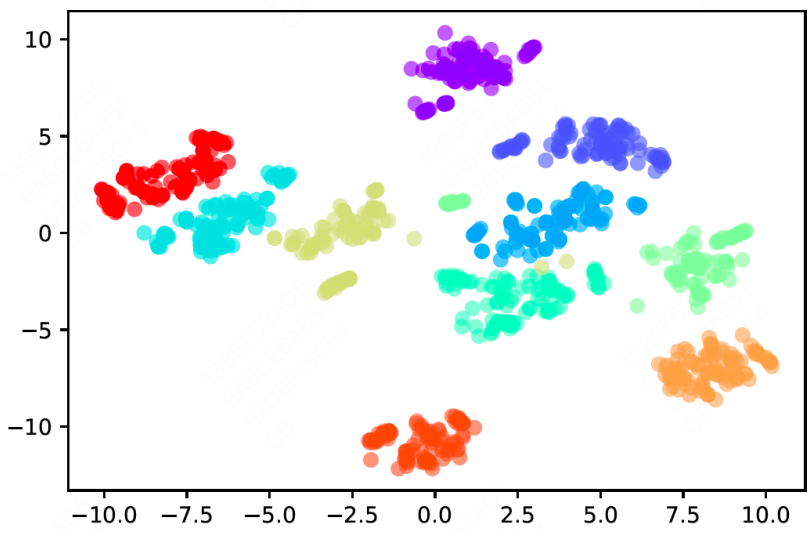


图 3　10 个点击 Item 在 100 个不同展示上下文下提取得到的上下文感知兴趣的可视化。每种颜色代表一个点击 Item

为了验证上下文建模的有效性，我们可视化了目标 Item 和在不同展示上下文下的同一点击 Item 之间的 IMM 的注意力权重。如图 4 所示，DIN（绿色）的权重是相同的，因

为 DIN 只将点击行为作为输入，所以它无法区分不同的显示上下文。而 DCIN 可以根据不同的上下文学习到不同的权重，右图展示的 Item 之间的竞争更激烈，其上下文反映的用户兴趣更精确，因此 DCIN 学习到了更高的权重。

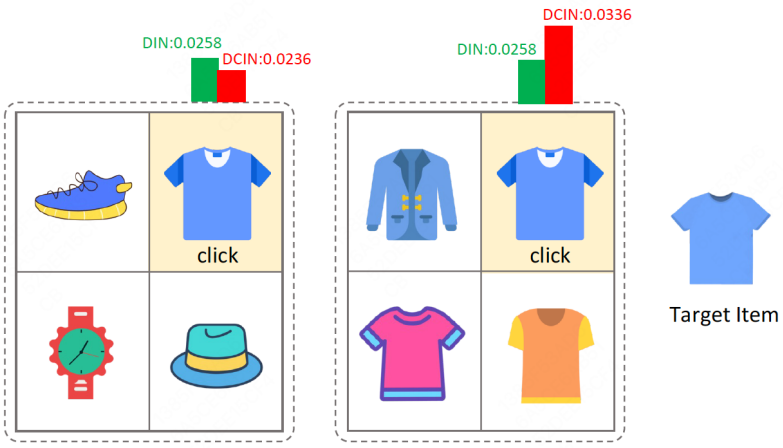


图 4　不同展示上下文下的同一点击 Item 与目标 Item 之间的注意力权重。由于缺少上下文信息，DIN 产生相同的注意力权重，而我们提出的 DCIN 可以差异化点击权重

4. 结论

我们强调了展示上下文和位次信息对兴趣建模的必要性，并创新性地设计了序列模型 DCIN。无论是在离线还是在线评估中，DCIN 都取得了显著的效果提升。目前，DCIN 已经在美团的在线广告系统中部署并全量，取得了很好的业务结果。

5. 参考文献

[1] Zhi Bian, Shaojun Zhou, Hao Fu, Qihong Yang, Zhenqi Sun, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. Denoising User-aware Memory Network for Recommendation. In Fifteenth ACM Conference on Recommender Systems. 400 - 410.

[2] Jes ú s Bobadilla, Ra ú l Lara-Cabrera, ngel Gonz á lez-Prieto, and Fernando Ortega. 2020. Deepfair: deep learning for improving fairness in recommender systems. arXiv preprint arXiv:2006.05255 (2020).

[3] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data. 1 - 4.

[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi

Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems. 7 – 10.

[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] Zhifang Fan, Dan Ou, Yulong Gu, Bairan Fu, Xiang Li, Wentian Bao, Xin-Yu Dai, Xiaoyi Zeng, Tao Zhuang, and Qingwen Liu. 2022. Modeling Users' Contextualized Page-wise Feedback for Click-Through Rate Prediction in E-commerce Search. (2022).

[7] Yufei Feng, Fuyu Lv,Weichen Shen, MenghanWang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. arXiv preprint arXiv:1905.06482 (2019).

[8] 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).

[9] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: a position-bias aware learning framework for CTR prediction in live recommender systems. In Proceedings of the 13th ACM Conference on Recommender Systems. 452 – 456.

[10] Jianqiang Huang, Ke Hu, Qingtao Tang, Mingjian Chen, Yi Qi, Jia Cheng, and Jun Lei. 2021. Deep Position-wise Interaction Network for CTR Prediction. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1885 – 1889.

[11] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 1419 – 1428.

[12] Xiang Li, Shuwei Chen, Jian Dong, Jin Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2023. Decision-Making Context Interaction Network for Click- Through Rate Prediction. Proceedings of the AAAI Conference on Artificial Intelligence 37, 4 (Jun. 2023), 5195 – 5202.

[13] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2671 – 2679.

[14] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, 11 (2008).

[15] MenghanWang, Mingming Gong, Xiaolin Zheng, and Kun Zhang. 2018. Modeling dynamic missingness of implicit feedback for recommendation. Advances in neural information processing systems 31 (2018).

[16] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2265 – 2268.

[17] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Deep feedback network for recommendation. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 2519 – 2525.

[18] Weinan Xu, Hengxu He, Minshi Tan, Yunming Li, Jun Lang, and Dongbai Guo. 2020.

Deep interest with hierarchical attention network for click-through rate prediction. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 1905 – 1908.

[19] Ling Yan, Wu-Jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled group lasso for web-scale ctr prediction in display advertising. In International Conference on Machine Learning. PMLR, 802 – 810.

[20] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In Proceedings of the Web Conference 2021. 2980 – 2991.

[21] 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 Proceedings of the AAAI conference on artificial intelligence, Vol. 33. 5941 – 5948.

[22] 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.

美团多场景建模的探索与实践

作者：王驰 森杰 树立 文帅 尹华 肖雄

1. 引言

美团到家 Demand-Side Platform（下文简称 DSP）平台，主要负责在美团外部媒体上进行商品或者物料的推荐和投放，并不断优化转化效果。随着业务的不断发展与扩大，DSP 对接的外部渠道越来越丰富、展示形式越来越多样，物料展示场景的差异性愈发明显（如开屏、插屏、信息流、弹窗等）。

例如，用户在午餐时间更容易点击【某推荐渠道下】【某 App】【开屏展示位】的快餐类商家的物料而不是【信息流展示位】的啤酒烧烤类商家物料。场景间差异的背后本质上是用户意图和需求的差异，因此模型需要对越来越多的场景进行定制化建设，以适配不同场景下用户的个性化需求。

业界经典的 Mixture-of-Experts 架构（MoE，如 MMoE、PLE、STAR[1] 等）能一定程度上适配不同场景下用户的个性化需求。这种架构将多个 Experts 的输出结果通过一个门控网络进行权重分配和组合，以得到最终的预测结果。早期，我们基于 MoE 架构提出了使用物料推荐渠道进行场景划分的多场景建模方案。然而，随着业务的不断壮大，场景间的差异越来越大、场景数量也越来越丰富，这版模型难以适应业务发展，不能很好地解决 DSP 背景下存在的以下两个问题：

1. **负迁移现象：**以推荐渠道为例，由于不同推荐渠道的流量在用户分布、行为习惯、物料展示形式等方面存在差异，其曝光数、点击率也不在同一个数量级（如下图 1 所示，不同渠道间点击率相差十分显著），数据呈现典型的“长尾”现象。如果使用推荐渠道进行多场景建模的依据，一方面模型会更倾向于学习到头部渠道的信息，对于尾部渠道会存在学习不充分的问题，另一方面尾部渠道的数据也会给头部渠道的学习带来“噪声”，导致出现负迁移。
2. **数据稀疏难以收敛：**DSP 会在外部不同媒体上进行物料展示，而用户在访问外部媒体时，其所处的时空背景、上下文信息、不同 App 以及物料展示位等信息共同构成了当前的场景，这样的场景在十万的量级，每个场景的数据又十分稀

疏，导致模型难以在每个场景上得到充分的训练。

在面对此类建模任务时，业界现有的方法是在不同场景间进行知识迁移。例如，SAML[2] 模型采用辅助网络来学习场景的共享知识并迁移至各场景的独有网络；ADIN[3] 和 SASS[4] 模型使用门控单元以一种细粒度的方式来选择和融合全局信息到单场景信息中。然而，在 DSP 背景中复杂多变的流量背景下，场景差异性导致了场景数量的急剧增长，现有方法无法在巨量稀疏场景下有效。

因此，在本文中我们提出了 DSP 背景下的自适应场景建模方案（AdaScene, Adaptive Scenario Model），同时从知识迁移和场景聚合两个角度进行建模。AdaScene 通过控制知识迁移的程度来最大化不同场景共性信息的利用，并使用稀疏专家聚合的方式利用门控网络自动选择专家组成场景表征，缓解了负迁移现象；同时，我们利用损失函数梯度指导场景聚合，将巨大的推荐场景空间约束到有限范围内，缓解了数据稀疏问题，并实现了自适应场景建模方案。

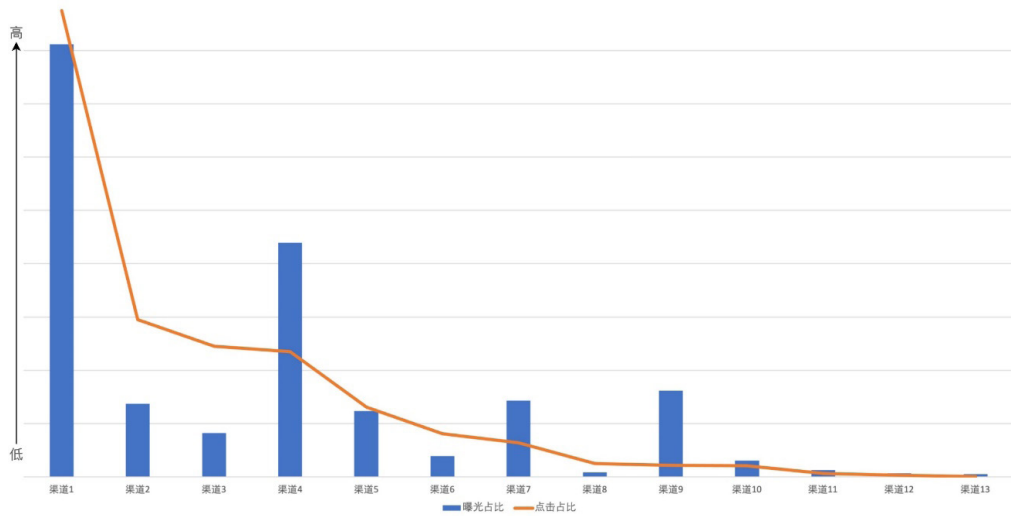


图 1　不同渠道规模差异

2. 自适应场景建模

在本节开始前，我们先介绍多场景模型的建模方式。多场景模型采用输入层 Embedding + 混合专家（Mixture-of-Experts, MoE）的建模范式，其中输入信息包括了用户侧、商家侧以及场景上下文特征。多场景模型的损失由各场景的损失聚合而成，其损失

函数形式如下：

$$Loss = \sum_{i=1}^{K} \sum_{j \in T_i} \alpha_j * L_j(W_i; D)$$

其中，K 为场景数量，α_i 为各场景的损失权重值。

我们提出的 AdaScene 自适应场景模型主要包含以下 2 个部分：场景知识迁移（Knowledge Transfer）模块以及场景聚合（Scene Aggregation）模块，其模型结构如下图 2 所示。场景知识迁移模块自适应地控制不同场景间的知识共享程度，并通过稀疏专家网络自动选择 K 个专家构成自适应场景表征。场景聚合模块通过离线预先自动化衡量所有场景间损失函数梯度的相似度，继而通过最大化场景相似度来指导场景的聚合。

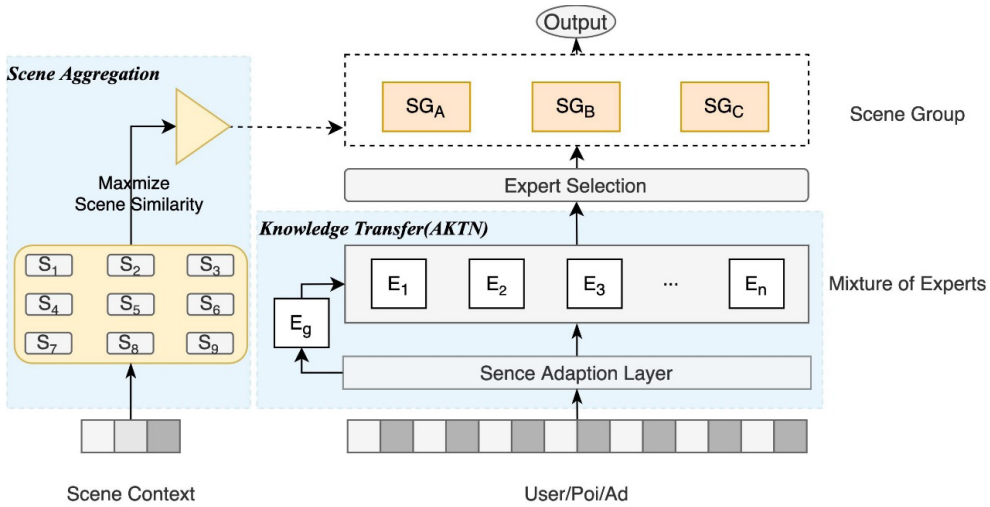


图 2　自适应场景建模 AdaScene 示意图

该模型结构的整体损失函数如以下公式所示：

$$min \quad \sum_{k=1}^{K} \alpha_k * \sum_{i=1}^{G_k} L_{s_i}(W_{s_i}, W_{sh}; G),\ G_k \in G$$

$$s.t. \quad \sum_{k=1}^{K} G_k = N$$

其中，α_k 为每个场景组的损失函数所对应的系数，G_k 为第 k 个场景组下的的场景数量，G 为某种场景组的划分方式。

下面，我们分别介绍自适应场景知识迁移和场景聚合的建模方案。

2.1 自适应场景知识迁移

在多场景建模中，场景定义方式决定了场景专家的学习样本，很大程度上影响着模型对场景的拟合能力，但无论采用哪种场景定义方式，不同场景间用户分布都存在重叠，用户行为模式也会有相似性。

为提升不同场景间共性的捕捉能力，我们从场景特征和场景专家两个维度探索场景知识迁移的方法，在以物料推荐渠道 ×App× 展示形态作为多场景建模 Base 模型的基础上，构建了如下图 3 所示的自适应场景知识迁移模型（Adaptive Knowledge Transfer Network, AKTN）。该模型建立了场景共享参数与私有参数的知识迁移桥梁，能够自适应地控制知识迁移的程度、缓解负迁移现象。

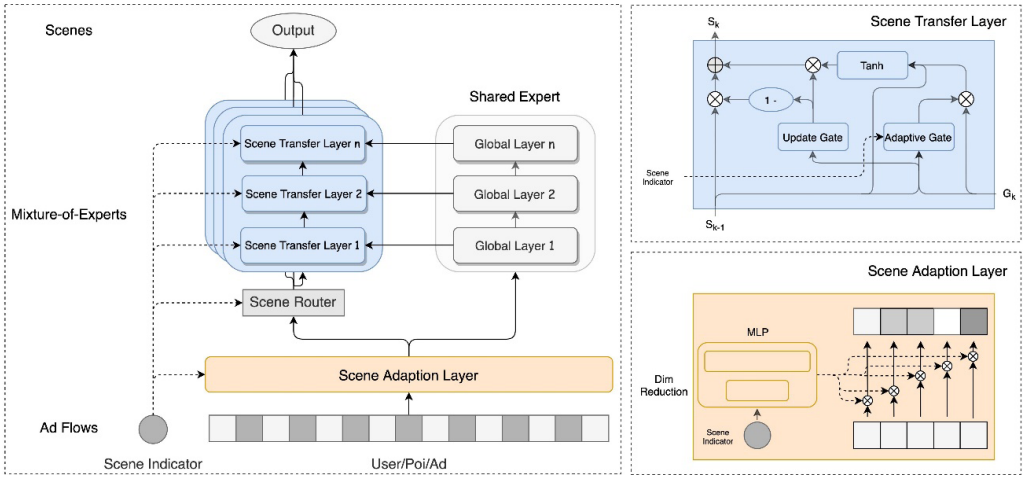


图 3　AKTN（Adaptive Knowledge Transfer Network）

- **场景特征适配：**通过 Squeeze-and-Excitation Network[5] 构建场景适应层（Scene Adaption Layer），其结构可表示为 $F_{SE}=FC(ReLU(FC(x)))$，其中 FC 表示全连接层，$ReLU$ 为激活函数。由于不同场景对原始特征的关注程度存在较大差异，该层能够根据不同场景的信息生成原始特征的权重，并利用这些权重对输入特征进行相应的变换，实现场景特定的个性化输入表征，提高模型的场景信息捕捉能力。
- **场景知识迁移：**使用 GRU 门控单元构建场景知识迁移层（Scene Transfer Layer）。GRU 门控单元通过场景上下文信息对来自全局场景专家和当前场景专家的信息流动进行控制，筛选出符合当前场景的有用信息；并且，该结构能以层级方式进行堆叠，不断对场景输出进行修正。

场景特征适配在输入层根据场景信息对不同特征进行权重适配，筛选出当前场景下模型最关注的特征；场景知识迁移在隐层专家网络中进行知识迁移，控制共享专家中共性信息向场景独有信息的流动，使得场景共性信息得以传递。

这两种知识迁移方式互为补充、相辅相成，共同提升多场景模型的预估能力。我们对比了不同模块的实验效果，具体结果如下表 1 所示。可以看出，引入场景知识迁移和特征权重优化在头部、尾部渠道都能带来一定提升，其中尾部小流量场景上（见下表 1 子场景 2、3）有更为明显的提升，可见场景知识迁移缓解了场景之间的负迁移现象。

GAUC(pp)	整体	大流量场景				小流量场景			
		子场景1	子场景4	子场景5	子场景8	子场景2	子场景3	子场景6	子场景7
+场景特征适配	+0.11	+0.05	+0.05	+0.08	+0.07	+0.67	+0.43	+1.01	+0.70
+场景知识迁移	+0.09	+0.02	+0.07	+0.04	+0.19	+0.59	+0.02	+0.21	+0.50
AKTN(融合)	+0.16	+0.09	+0.14	+0.14	+0.10	+0.72	+0.54	+0.72	+0.56

表 1　AKTN 实验效果

相关研究和实践表明[6][7][8]，稀疏专家网络对于提高计算效率和增强模型效果非常有用。因此，我们在 AKTN 模型的基础上，在专家层进一步优化多场景模型。具体的，我们将场景知识迁移层替换为自动化稀疏专家选择方法，通过门控网络从大规模专家中选取与当前场景最相关的 KK 个构成自适应场景表征，其选择过程如下图 4 所示：

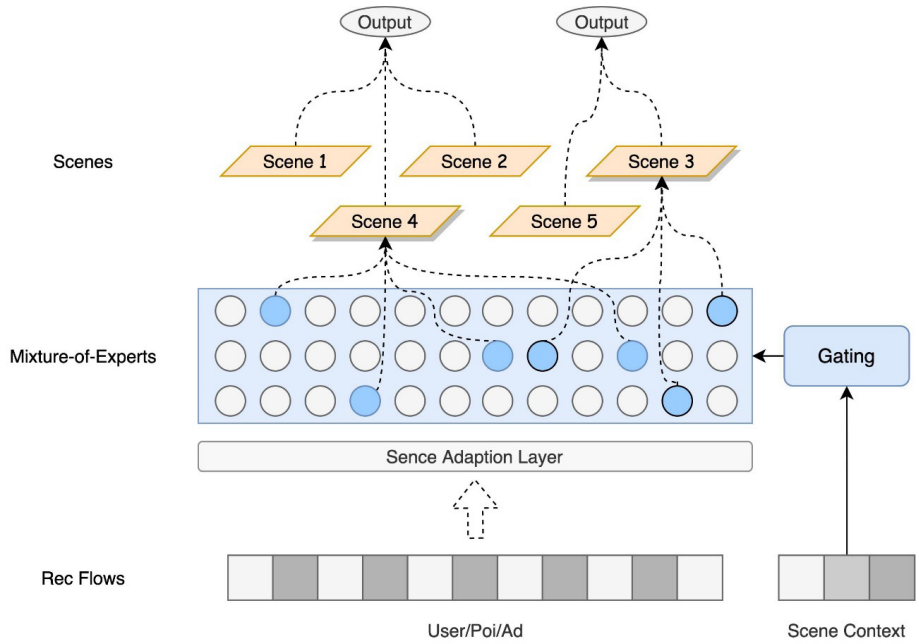


图 4　稀疏专家网络示意图

在实践中，我们通过使用可微门控网络对专家进行有效组合，以避免不相关任务之间的负迁移现象。同时大规模专家网络的引入扩大了多场景模型的选择空间，更好地支持了门控网络的选择。考虑到多场景下的海量流量和复杂场景特征，在业界调研的基础上对稀疏专家门控网络进行了探索。

具体而言，我们对以下稀疏门控方法进行了实践：

- **方法一：**通过 KL 散度衡量子场景与各专家之间的相似度，以此选择与当前场景最匹配的 k 个专家。在实现方式上，使用场景 * 专家的二维矩阵计算相似性，并通过 KL 散度选择出最适合的 k 个专家。
- **方法二：**每个子场景配备一个专家选择门控网络，m 个场景则有 m 个门控网络。对于每个场景的门控网络，配备 k 个单专家选择器 [9]，每个单专家选择器负责从 n 个专家中选择一个作为当前场景的专家（n 为 Experts 个数）。在实践中，为提高训练效率，我们对单专家选择器中权重较小的值进行截断，保证每个单专家选择器仅选择一个专家。

在离线实验中，我们以物料推荐渠道 * 展示形态作为场景定义，对上述稀疏门控方法进行了尝试，离线效果如下表 2 所示：

方法	门控方法	GAUC（pp）
1	Base（PLE）	–
2	方法一	+0.09
3	方法二	+0.28

表 2　稀疏门控方法效果

可以看出，基于软共享机制的专家聚合方法能够更好地通过所激活的相同专家网络对各场景之间的知识进行共享。相较于常见的以截断方式为主的门控网络，使用二进制编码的方式使得其在不损失其他专家网络信息的同时，能够更好地收敛到目标专家数量，同时其可微性使得其在以梯度为基础的优化算法中训练更加稳定。

同时，为了验证稀疏门控网络能否有效区分不同场景并捕捉到场景间差异性，我们使用 n=16 个专家中选择 K=7 个的例子，对验证集中不同场景下各专家的利用率、选择专家的平均权重进行了可视化分析（如图 5– 图 7 所示），实验结果表明该方法能够有效地选择出不同的专家对场景进行表达。

例如，图 6 中 KP_1 更多地选择第 5 个专家，而 KP_2 更倾向于选择第 15 个专家。

并且，不同场景对各专家的使用率以及选择专家的平均权重也有着明显的差异性，表明该方法能够捕捉到细分场景下流量的差异性并进行差异化的表达。

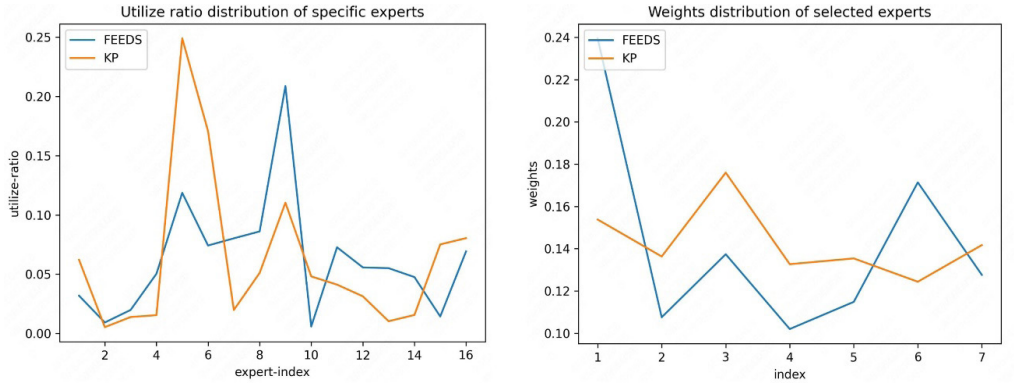


图 5　同渠道下不同展示形式专家分布

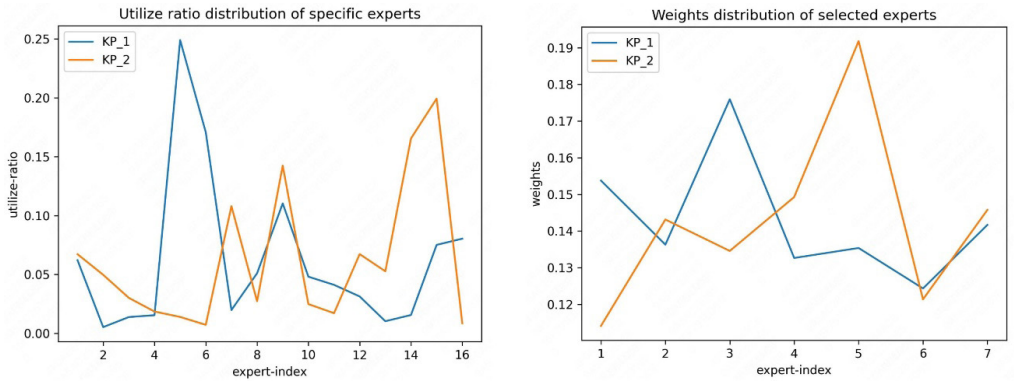


图 6　开屏展示不同渠道的专家分布

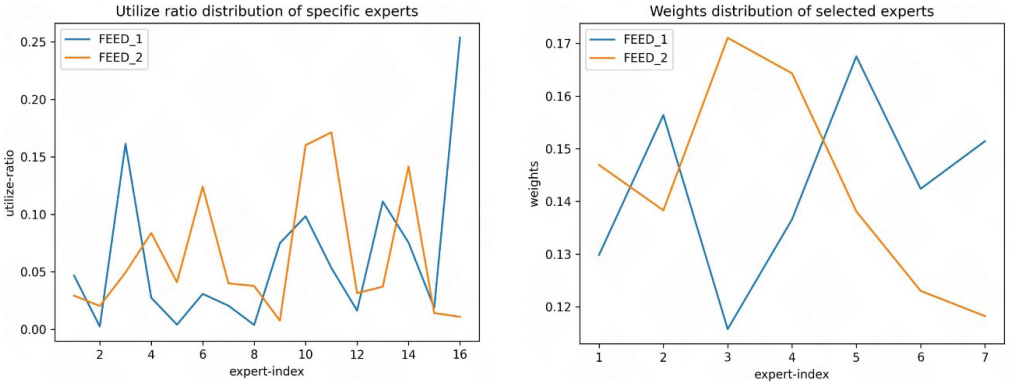


图 7　信息流展示不同渠道的专家分布

实验证明，在通过大规模专家网络对每个场景进行建模的同时，基于软共享机制的专家聚合方法能够更好地通过所激活的相同专家网络对各场景之间的知识进行共享。 同时，为了进一步探索 Experts 个数对模型性能的影响，我们在方法二的基础上通过调整专家个数和 topK 比例设计了多组对比实验，实验结果如下表 3 所示：

GAUC（pp）	10%	20%	40%	60%	80%
4 Experts	–	–	+0.12	+0.15	+0.14
8 Experts	+0.18	+0.20	+0.24	+0.22	+0.20
16 Experts	+0.21	+0.24	+0.27	+0.23	+0.24
32 Experts	+0.19	+0.26	+0.28	+0.27	+0.26

表 3　方法二调参实验

从实验数据可以看出，大规模的 Experts 结构会带来正向的离线收益；并且随着选取专家个数比例的增加（表 3 横轴），模型整体的表现效果也有上升的趋势。

2.2 自适应场景聚合

理想情况下，一条请求（流量）可以看作一个独立的场景。但如引言所述，随着 DSP 业务持续发展，不同的物料展示渠道、形式、位置等持续增加，每个场景的数据十分稀疏，我们无法对每个细分场景进行有效训练。因此，我们需要对各个推荐场景进行聚类、合并。我们使用场景聚合的方法对此问题进行求解，通过衡量所有场景间的相似度，并最大化该相似度来指导场景的聚合，解决了数据稀疏导致难以收敛的问题。具体的，我们将该问题表示为：

$$G^* = \underset{G}{arg\,max} \sum_{k=1}^{K} \sum_{i=1}^{G_k} f_{s_i},\ G_k \in G$$

$$s.t. \quad \sum_{k=1}^{K} G_k = N$$

其中 G 表示某种分组方式，f_{si} 为场景 s_i 在分组 G_k 内与其他场景的总体相似度。在将 N 个场景聚合成 K 个场景组的过程中，我们需要找到使得场景间整体相似度最大的分组

方式 G^*。

因此，我们在 2.1 节场景知识迁移模型的基础上，增加了场景聚合部分，提出了基于 Two-Stage 策略进行训练的场景聚合模型：

- **Stage 1：**基于相似度衡量方法对各场景的相似度进行归纳，并以最大化分组场景的相似度为目标找到各场景的最优聚合方式（如 Scene1 与 Scene 4 可聚合为场景组合 Scene Group SGA）；
- **Stage 2：**基于 Stage 1 得到的场景聚合方式，以交叉熵损失为目标函数最小化各场景下的交叉熵损失。

其中，Stage 2 与 2.1 节中所述一致，本节主要针对 Stage 1 进行阐述。我们认为，一个有效的场景聚合方法应该能自适应地应对流量变化的趋势，能够发现场景之间的内在联系并依据当前流量特点自动适配聚合方法。我们首先想到的是从规则出发，将人工先验知识作为场景聚合的依据，按照推荐渠道、展示形式以及两者叉乘的方式进行了相应迭代。然而这类场景聚合方式需要可靠的人工经验来支撑，且在应对海量流量时不能迅速捕捉到其中的变化。

因此，我们对场景之间关系的建模方法进行了相关的探索。首先，我们通过离线训练时场景之间的表征迁移和组合训练来评估场景之间的影响，但这种方式存在组合空间巨大、训练耗时较长的问题，效率较低。

在多任务的相关研究中 [10][11][12][13]，使用梯度信息对任务之间的关系进行建模是一种有效的方法。类似的在多场景模型中，能够根据各场景损失函数的梯度信息对场景间的相似度进行建模，因此我们采用多专家网络并基于梯度信息自动化地对场景之间的相似度进行求解，模型示意如下图 8 所示：

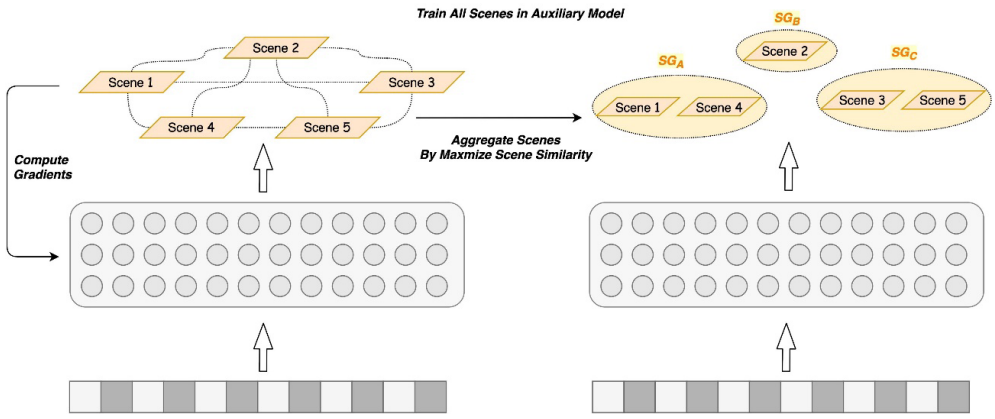


图 8　场景聚合示意图

基于上述思路，我们对场景之间的关系建模方法进行了以下尝试：

1. Gradient Regulation

基于梯度信息能够对场景信息进行潜在表示这一认知，我们在损失函数中加入各场景损失函数关于专家层梯度距离的正则项，整体的损失函数如下所示，该正则项的系数 $\lambda_{si,sj}$ 表示场景之间的相似度，$dist$ 为常见的评估梯度之间距离的方法，比如 l_1，l_2 距离。

$$Loss = \sum_{i=1}^{N} L_{si} + \sum_{0<i,j<=N} \lambda_{si,sj} * dist(\nabla_{Ei} L_{si}, \nabla_{Ej} L_{sj}), \quad i \neq j$$

2. Lookahead Strategy

在Gradient Regulation方法中，场景1对场景2的影响与场景2对场景1的影响是一致的，即$\lambda_{s_i \to s_j} = \lambda_{s_j \to s_i}$，但这种假设是不合理的。在离线训练中，由于不同场景所对应的模型参数存在不同，场景间的相互影响应该有所差异。

进一步的，为了更加显式的建模出场景之间的相互影响，在训练时我们采用提前更新参数的策略，通过单个场景损失函数的梯度信息对模型参数进行更新后，其他场景损失函数的增益表示该场景与其他场景的相似度。具体的求解方法如下所示，W_{sh}为模型的共享参数，W_{s_i}为场景s_i的独有参数，最终对所有训练Batch内产生的训练增益的均值$\lambda_{s_i \to s_j}$作为场景之间的相似度。

$$Loss = \sum_{i=1}^{N} L_{s_i}(D_{s_i}; W_{sh}, W_{s_i})$$

$$W_{sh|s_i}^{t+1} = W_{sh}^{t} - \eta * \nabla_{W_{sh}^t} L_{s_i}(D^t; W_{sh}^t, W_{s_i}^t)$$

$$\lambda_{s_i \to s_j}^{t} = 1 - \frac{L_j(D^t; W_{sh|s_i}^{t+1}, W_{s_j}^t)}{L_j(D^t; W_{sh}^t, W_{s_j}^t)}$$

$$\lambda_{s_i \to s_j} = \frac{1}{T} \sum_{t=1}^{T} \lambda_{s_i \to s_j}^{t}$$

3. Meta Weights

Lookahead Strategy 该方法对场景间的关系进行了显式建模，但是这种根据损失函数的变化计算场景相关系数的策略存在着训练不稳定、波动较大的现象，无法像 Gradient Regulation 这一方法对场景相似度进行求解。

因此，我们引入了场景间的相关性系数矩阵（meta weights），结合前两种方法对该问题进行如下建模，通过场景s_i的数据对其与其他场景的相关性系数$\lambda_{s_i \to s_j}$进行更新，同时基于该参数对全局的参数模型W进行优化。针对这种典型的两层优化问题，我们基于MAML[14]方法进行求解，并将meta weights作为场景间的相似度。

$$min_\lambda \sum_{0<i<=N} L_{si}(D_{si}; W^*_{sh}, W^*_{si})$$

$$s.t.\ W^* = arg\ min_W \sum_{j=1}^{N} \lambda_{si \to sj} * L_{sj}(D_{sj}; W_{sh}, W_{si})$$

我们以推荐渠道和展示形式（是否开屏）的多场景模型作为 Base，对上述 3 种方法做了探索。为了提高训练效率，我们在设计 Stage 1 模型时做了以下优化：

- **优化梯度信息**：在计算梯度信息的时候我们发现存在一定的梯度消失或爆炸的问题，使得模型训练不稳定，因此在训练过程中对采用L2范数对梯度进行裁剪，提高了训练的收敛速度；
- 为了更高效地对场景组的整体相似度进行表示，对于目标场景，我们将该场景与其他场景之间两两相似度的均值作为该场景在这个场景组下与其他场景的总体相似度。具体的计算过程如以下公式所示，其中G_k表示第k个场景组中的场景数量，$\lambda_{s_j \to s_i}$表示场景s_j与s_i之间的相似度：

$$f_{s_i} = \frac{\sum_{j=1}^{G_k} \lambda_{s_j \to s_i}}{G_k - 1}, \quad i \leqslant G_k, i \neq j$$

我们对每个方法的 GAUC 进行了比较，实验效果如下表 4 所示。相较于人工规则，基于梯度的场景聚合方法都能带来效果的明显提升，表明损失函数梯度能在一定程度上表示场景之间的相似性，并指导多场景进行聚合。

方法	Stage 1	Stage2	GAUC（pp）
1	人工规则（Base）	AKTN	–
2	Gradient Regulation	AKTN	+0.18
3	Lookahead Strategy	AKTN	+0.22
4	Meta Weights	AKTN	+0.26

表 4　场景聚合实验数据

为了更全面的展现场景聚合对于模型预估效果的影响，我们选取 Meta Weights 进行分组数量的调优实验，具体的实验结果如下表 5 所示。可以发现：随着分组数的增大，GAUC 提升也越大，此时各场景间的负迁移效应减弱；但分组超过一定数量时，场景间总体的相似度减小，GAUC 呈下降趋势。

分组个数	n = 1	n = 3	n = 5	n = 7	n = 9	n = 11
GAUC（pp）	−0.14	+0.12	+0.23	+0.26	+0.24	+0.21

表 5　不同聚合场景数量实验数据

此外，我们对 Meta Weigts 方法中部分场景间的关系进行了可视化分析，分析结果如下图 9 所示。以场景作为坐标轴，图中的每个方格表示各场景间的相似度，颜色的深浅表示渠道间的相似程度大小。

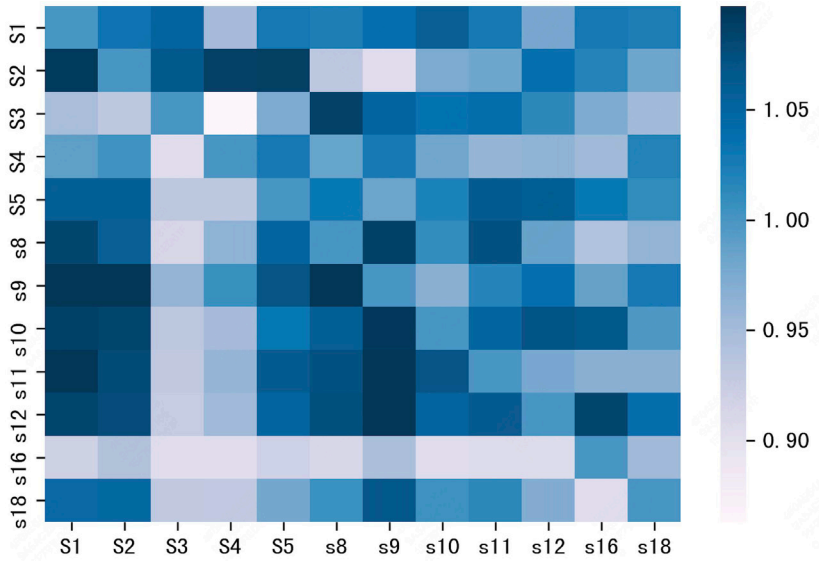


图 9　部分细分场景下的相似度示例

从图中可以发现，以渠道和展示形式为粒度的细分场景下，该方法能够学习到不同场景间的相关性，例如 A 渠道下的信息流 (s16) 与其他场景的相关性较低，会将其作为独立的场景进行预估，而 B 渠道下的开屏展示（s9）与 C 渠道开屏展示（s8）相关性较高，会将其聚合为一个场景进行预估，同时该相似度矩阵不是对称的，这也说明各场景间相互的影响存在着差异。

3. 总结与展望

通过多场景学习的探索和实践，我们深入挖掘了推荐模型在不同场景下的建模能力，并分别从场景知识迁移、场景聚合方向进行了尝试和优化，这些尝试提供了更好的理解和解释推荐模型对不同类型流量和场景的应对能力。然而，这只是多场景学习研究的开始，后续我们会探索并迭代以下方向：

- **更好的场景划分方式：**当前多场景的划分主要还是依据渠道（渠道 * 展示形态）作为流量的划分方式，未来会在媒体、展示位、媒体 * 时间等维度上进行更详细地探索；
- **端到端的流量聚合方式：**在进行流量聚合时，使用了 Two-Stage 的策略进行聚合。然而，这种方式不能充分地利用流量数据中相关的信息。因此，需要探索端到端的流量场景聚合方案将更直接和有效地提高推荐模型的能力。

结合多场景学习，在未来的研究中将不断探索新的方法和技术，以提高推荐模型对不同场景和流量类型的建模能力，创造更好的用户体验以及商业价值。

4. 作者简介

王驰、森杰、树立、文帅、尹华、肖雄等，均来自美团到家事业群 / 到家研发平台。

5. 参考文献

[1] STAR：Sheng, Xiang-Rong, et al. “One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction.” Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021.

[2] SAML：Chen, Yuting, et al. “Scenario-aware and Mutual-based approach for Multi-scenario Recommendation in E-Commerce.” 2020 International Conference on Data Mining Workshops (ICDMW). IEEE, 2020.

[3] ADIN：Jiang, Yuchen, et al. “Adaptive Domain Interest Network for Multi-domain Recommendation.” Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022.

[4] SASS：Zhang, Yuanliang, et al. “Scenario-Adaptive and Self-Supervised Model for Multi-Scenario Personalized Recommendation.” Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022.

[5] Squeeze-and-Excitation：Hu, Jie, Li Shen, and Gang Sun. “Squeeze-and-excitation networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[6] 美团外卖推荐情境化智能流量分发的实践与探索

[7] PaLM：https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html
[8] GLaM：https://proceedings.mlr.press/v162/du22c.html
[9] 单专家选择器：https://arxiv.org/abs/2106.03760
[10] HOA：https://proceedings.mlr.press/v119/standley20a.html
[11] Gradient Affinity：https://proceedings.neurips.cc/paper/2021/hash/e77910ebb93b511588557806310f78f1-Abstract.html
[12] SRDML：https://dl.acm.org/doi/abs/10.1145/3534678.3539442
[13] Auto-Lambda：https://arxiv.org/abs/2202.03091
[14] MAML：https://arxiv.org/abs/1703.03400

美团外卖智能陪伴型导购的探索与实践

作者：翔锟 营飞 关璐等

1. 背景

1.1 什么是外卖搜索导购?

搜索导购是外卖搜索中的个性化搜索词建议模块的统称，在搜索路径中的各处为用户提供引导，提升用户的选购效率。下面按照用户使用外卖搜索的路径来介绍搜索导购的各个模块：① 用户进入外卖首页，美团在顶部搜索框的内部和下方分别提供搜索建议，即**框内词**和**框下词**；② 用户点击搜索框会进入到搜索引导页中，我们在历史搜索下方的**搜索发现**中提供搜索建议；③ 用户在搜索框中主动输入搜索词后，将会唤起搜索词联想功能，即**搜索 SUG**，为用户提供补全搜索词的建议；④ 用户发起搜索后将进入到搜索结果页。

整体流程如下图 1 所示，其中红框部分即为本文讨论的重点：框内词、框下词以及搜索发现模块。



图 1　用户使用外卖搜索的主要路径

1.2 为什么要做智能陪伴型的外卖搜索导购?

在餐饮领域，用户体验永远是线下商家优化的重点。其中，导购服务直接影响着用户的购前决策，对商家的转化率和营业额有着重大的意义。像海底捞等商家甚至主打极致的用户体验，通过无微不至的店员服务让用户感到“时时刻刻被关怀”。外卖服务将餐饮

消费场景从线下搬到了线上，用户通过搜索 / 推荐场景找到自己感兴趣的商家和菜品，但相较于线下环境中有“时刻关怀着”的店员，线上这个过程显得较为“被动和冰冷”。

基于以上灵感，我们决定从 2022 年开始着力打造**智能陪伴型导购**。在用户使用搜索 / 推荐时，我们围绕用户表现出的兴趣主动提供更智能的搜索建议，更好地承接用户实时变化的意图和被激发出的灵感。同时，我们也解决了用户打字输入成本高、不知道附近的供给能否满足他们的需求以及不知道如何清晰表达他们的需求等三个方面的痛点。

2. 问题与挑战

如何将智能陪伴型导购这一充满“赛博朋克”感的构想在美团外卖场景进行落地呢？我们遇到了以下的挑战：

- **挑战 1：**线下的导购追求一对一的极致实时的个性化体验，而外卖搜索的导购如何由被动导购（用户手动刷新才会触发导购系统的刷新）变为主动导购（智能感知和判断用户在什么时间、什么场景需要针对性的刷新导购），强化用户实时的个性化体验，智能感知和判断用户每一个需要服务的场景？解决方案：借助美团自研的 Alita 端智能 [1][2] 的意图感知能力，智能判断何时需要针对性地刷新。同时，对模型引入实时异构行为序列进行建模，打造端云结合排序模型，感知用户实时多变的兴趣。
- **挑战 2：**导购的优化除了满足用户体验之外，还需要为整体的转化目标负责，如何让导购的优化与下游场景 / 整体场景的优化目标保持一致？解决方案：对全场景数据进行统一建模，打造基于自监督学习的统一模型，在对样本进行“无效”过滤以及归因的基础上，同时联动搜索结果页进行全链路的多目标效率优化。
- **挑战 3：**如何像基于大模型对话入口一样将用户在物理世界的行为转化为机器 / 算法可以理解的信息，并像真人导购一样和用户“对话”？解决方案：我们后续计划通过对用户行为的理解，尝试在适当时机主动将用户引导到大模型的对话入口，满足其深度对话的需求。

3. 主要工作

我们将以上提到的问题和挑战作为主线，分为两个章节进行详细介绍。其中在 3.1 章节

介绍框内词智能刷新，3.2 章节介绍导购 Query 推荐全场景统一模型。

3.1 框内词智能刷新

3.1.1 智能刷新推荐框架

整体链路

上文提到，要实现框内词智能刷新，搭建出一套适用于外卖的、基于端智能框架的 Query 推荐系统非常重要。搭建思路可以用 4W1H 来总结：

- **Where/How**：框内词智能刷新的 Query 展示在哪？框内词智能刷新的 Query 的展现形式是什么？（涉及产品形态）
- **Who/When**：框内词智能刷新需要对什么样的用户触发？在什么时机下触发？（涉及用户意图感知）
- **What**：框内词智能刷新需要展示什么？（涉及算法策略）

在对上述问题进行深入思考和多方探讨后，我们最终与产品、端智能、客户端、应用服务和推荐工程等相关方一起，搭建了适用于外卖导购的框内词智能刷新 Query 推荐链路：

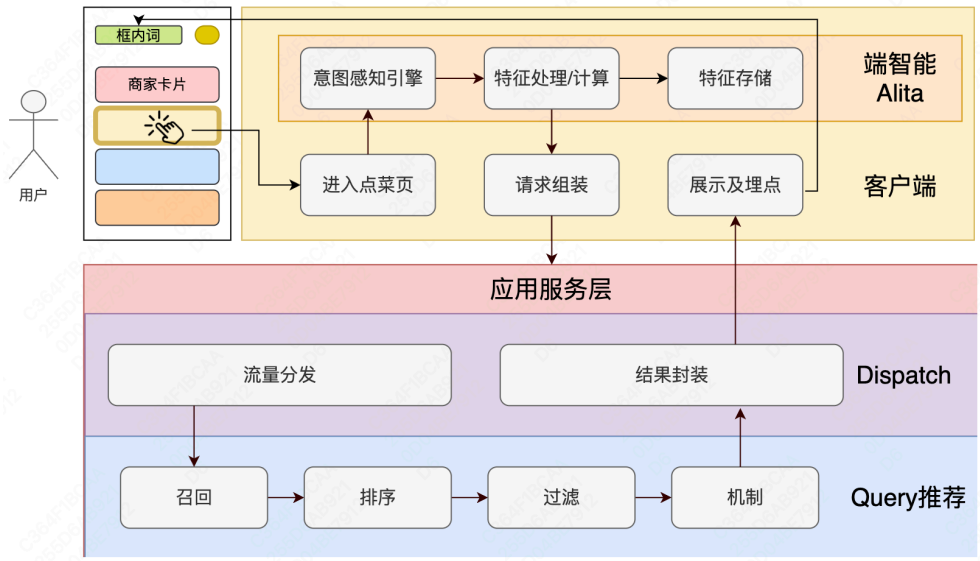


图 2　框内词智能刷新整体链路

上图 2 展示了从“用户点击首页 Feed 商家卡片”开始，到框内词智能刷新的 Query 展现”的全流程。当客户端收集到用户进店行为后，调用 Alita 的意图感知引擎；满足框内词智能刷新的触发条件后，对用户在端上的进行特征处理、计算和存储，并将计算好的特征传递给客户端组装框内词智能刷新的请求；框内词智能刷新的请求经过应用服务层的 Disptch 透传给 Query 推荐的后端服务，经过召回、排序、过滤、机制、封装阶段，最终返回框内词智能刷新的结果到客户端进行展示。

产品形态

早期导购服务在用户进入外卖首页时会主动触发刷新，但除非用户手动触发刷新操作，框内词会一直轮播首次刷新时的结果，造成了大量的流量浪费。

为了解决这个问题，我们尝试利用基于 Alita 的意图感知能力。如视频演示，当用户在首页浏览商家并加购菜品“【 爆款 】麻酱凉皮”后返回首页，吸顶的搜索框中的框内词会自动更新，并轮播到新的搜索词“凉皮”，同时展示擦亮动效。当用户点击搜索按钮后，会直接跳转到搜索结果页，展示更多与凉皮相关的商家及菜品，帮助用户进行选购。

3.1.2 智能刷新推荐策略

端实时信号收集

用户实时的下单兴趣与用户在主页、店内的实时行为有着较强的关联，如何通过端上实时信号的收集能力去获取、加工实时行为以便于推荐系统更好理解用户意图是我们关注的重点。相比服务端，用户在端上的特征主要有以下两个特点：

1. 实时性更好，从“准实时”到“超实时”的交互；
2. 维度更细，从“交互 Item”进化到“Item 交互的 Micro- 粒度”[3]。

因此，借助端智能的能力，我们不再受限于用户主动下拉请求刷新的机制，能够实时感知用户实时行为、偏好，实时智能决策更新推荐结果，缓解反馈信号感知延迟问题。

我们梳理了用户点击商家卡片后在店内以及店内商品详情页的主要行为，利用 Alita 在端上信息收集的能力，获取了当前 Session 内的商家的点击信息及在店内其停留时长等信息，同时我们记录了用户在店内针对菜品的各种行为，具体包括查看、点击或者加

购等，以参数的形式传给后端供推荐排序策略使用，更细粒度地捕捉用户实时变化的述求。

端实时触发策略

为了探索最优的触发条件，我们根据用户在店内的不同行为和用户停留时间，结合如下图 4 所示的用户店内停留时长分布以及用户首页点击数量分布，尝试了菜品点击菜品、加购菜品等行为触发，停留时长（10s/5s/2s/0s）触发等条件。

实验证明，放宽触发条件虽然会通过增加框内词智能刷新请求的曝光量，覆盖更多的用户，有利于后续策略迭代，但也会增大不置信触发的风险，导致框内词的效率下降。最终，我们选定将进店大于 2s 或者进店点击过菜品作为智能刷新的触发条件，两个条件满足其一即可。

此外，考虑到用户在店内浏览商家详情页期间会点击或加购不止一个菜品，而且用户在商家详情页的停留时长越长、用户行为越丰富，服务端的推荐结果也就越精准，所以我们没有选择当用户满足触发条件后（进店大于 2s 或者进店点击过菜品）立即触发作为我们的触发时机，而是选择对于满足触发条件的请求，会在“用户从首页 Feed 的商家详情页离店后立即触发”作为我们的触发时机。

这么做的好处是，离店时会将用户在店内的所有行为通过客户端带到服务端，帮助智能刷新系统可以更精确地捕获用户的喜好，同时还可以减少同一商家内多次触发带来的不必要的性能开销。同时为了缓解性能的压力，我们设置了同一 Session 内最多触发 30 次刷新的策略限制，经过统计该策略覆盖的 UV 超过 90%，已经可以满足我们的需求。

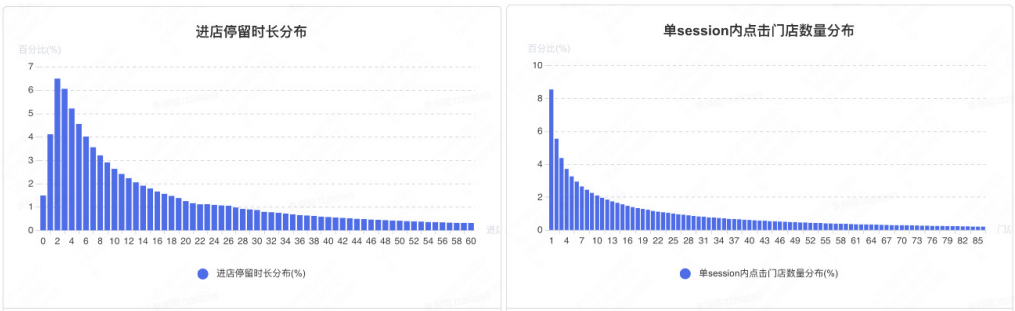


图 4　进店停留时长分布（左）及单 session 内点击门店数量分布（右）

在线效果及线上指标计算口径如下：

- **搜索 UV_RPM** = 搜索实付交易额 / 搜索曝光人数（DAU）* 1000
- **搜索 UV_CXR** = 搜索交易用户数 / 搜索曝光人数（DAU）
- **框内词 UV_CTR** = 框内词点击用户数 / 框内词曝光用户数

智能刷新触发策略优化	导购服务内
	框内词UV_CTR
进店3s触发 vs 不触发	+1.42%
离店触发 vs 进店3s触发	+9.13%

智能刷新召回策略

通过数据分析，我们发现 14.9% 的用户会在进店行为后发起搜索，且此种情况贡献了搜索 20% 的 QV，证明了搜 / 推场景联动可以帮助外卖场景“新动线”（搜索联动推荐联合优化，带动外卖全局的增长）的推进。进一步分析发现，以上情况下，60% 的用户的搜索 Query 与用户 Last-Click 行为存在强相关性，表现为用户通常会在点击商家后继续搜索 Last-Click 店铺相应的品牌或者主营品类。

基于以上分析，本文对召回策略进行优化。与用户手动刷新框内词的请求不同，我们对框内词智能刷新系统的召回进行了以下两点优化，一个是对于 Trigger 实时性的优化，另一个是为了更贴合用户实时的兴趣意图，我们利用更实时的 Trigger 设计了多种 X2Q 的召回 [3]。

- **端云联动的 Trigger 实时性优化：**通过将端上 Alita 的特征透传到用户兴趣中心 Tribe 中，在 Tribe 中实现端特征解析加工，使 Tribe 拥有更加实时的用户特征信息。我们融合了在端侧收集到的实时信号以及云侧用户兴趣中心收集到的实时点击序列，将最近一个商家作为我们的触发 Trigger。
- **新增 POI2Query 召回：**根据商家历史 30 天搜索成单数据，聚合搜索场景下单到当前 POI 的 Top Query 集合，得到搜索成单商家 Query 召回；通过挖掘过去 30 天用户在点击对应 POI 后搜索的 Top Query 集合，得到点击商家后搜索 Query 召回。

- **新增 SPU2Query 召回：**利用端上获取到的触发框内词智能刷新的点击和加购 SPU 信息作为 Trigger，根据 SPU 历史 30 天搜索成单数据，聚合搜索场景下单到当前 SPU 的 Top Query 集合。
- **新增 POI2Query2Query 召回：**早期的优化验证了 POI2Query 的有效性，所以我们通过 Swing CF 的方式理解计算了 Query 之间的相似性，将用户触发端智能的 POI 泛化的 Top1 的 Query 作为 Trigger，泛化出相似的相关 Query。

我们在线上进行了多组召回的消融实验，具体效果如下表：

智能刷新召回策略优化	导购服务内	搜索贡献		
	框内词UV_CTR	搜索DAU	搜索UV_CXR	搜索UV_RPM
+端云融合的Last-Click作为Trigger	+9.56%	+0.02%	+0.03%	+0.06%
+ 搜索成单商家Query召回	+7.47%	+0.04%	+0.04%	+0.08%
+ 点击商家后搜索Query召回	+6.70%	+0.03%	+0.03%	+0.04%
+ 搜索成单菜品Query召回	+5.64%	+0.02%	+0.03%	+0.02%
+ 搜索成单商家Query泛化召回	+3.11%	+0.01%	+0.00%	+0.02%

智能刷新排序策略

基于端 / 云融合排序模型：鉴于场景冷启，我们刚开始直接使用了非智能刷新场景的导购排序模型（用户手动刷新场景调用的框内词模型），该模型引入了用户实时商家点击序列、实时搜索序列以及用户的 Last Behavior 进行联合建模。由于智能刷新场景有实时性要求更高的特点，如图 5 所示，我们在用户行为序列的使用中，融合了端上搜集的更为实时的信号，提升了用户行为序列的实时性。

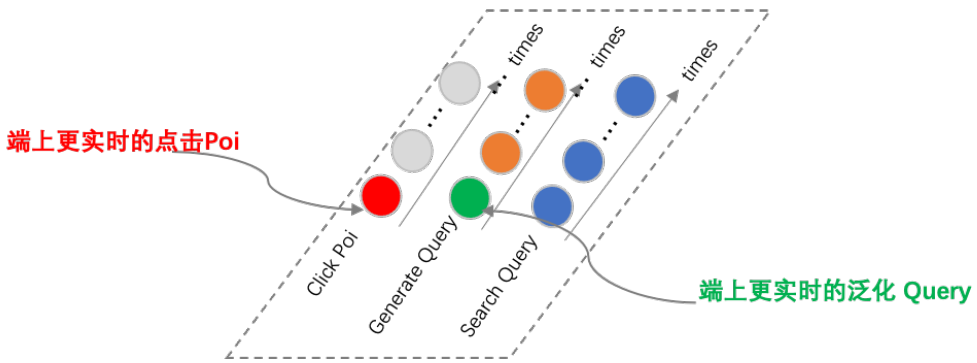


图 5　端 / 云融合的用户兴趣序列

基于预训练的自监督学习模型（利用用户主动输入数据进行预训练，场景数据 Fine-tuning 的方式进行学习）：为了更好地学习到本场景内用户的兴趣偏好，本文在词服务多场景建模的模型中加入了智能刷新场景内的数据进行模型迭代。为了防止因为新场景数据较少导致过拟合，我们利用用户主动输入数据进行预训练，场景数据 Fine-tuning 的方式进行学习，并取得了不错的效果，详细做法见 3.2。

线上消融实验的具体效果见下表：

智能刷新排序策略优化	离线指标		导购服务内	搜索贡献		
	CTR AUC	CXR AUC	框内词 UV_CTR	搜索DAU	搜索 UV_CXR	搜索 UV_RPM
+ 基于端/云融合排序模型 vs 非智能刷新场景的导购排序模型	+8.73pp	+7.92pp	+12.71%	+0.03%	+0.03%	+0.04%
+ 基于端/云融合排序模型 + 融合本场景内数据 vs 基于端/云融合排序模型	+5.17pp	+5.34pp	+5.28%	+0.02%	+0.01%	+0.03%
+ 预训练的自监督学习模型 vs 基于端/云融合排序模型 + 融合本场景内数据	+6.38pp	+5.91pp	+7.92%	+0.04%	+0.03%	+0.03%

3.2 导购 Query 推荐全场景统一模型

随着端智能在导购场景的应用，词服务场景已经扩展到了如图 1 所示的五个场景（框内词、框下词、搜索发现、框内词智能刷新以及猜你想搜）。然而，如果每个场景都单独迭代一个模型，这将耗费大量时间和精力。此外，单场景模型无法利用多个场景的共性进行辅助学习。对于新场景或小场景等训练数据较少的场景，直接使用单个场景的数据进行训练，很容易导致过拟合问题。因此，我们尝试统一导购的五个 Query 推荐场景的模型。

下面，本文将从特征建设、样本建设、用户实时异构行为序列建模、导购联动搜索结果页全链路多目标效率优化以及基于自监督学习的全场景统一模型等五个模块分别介绍我们的工作。

3.2.1 特征建设

特征在排序的效果中，起到非常重要的作用，基于外卖搜索导购的实际场景，结合外卖推搜特征 Matrix 规范及特征重要度分析，如下图 6 所示，我们从 User、POI、Query、Context 以及组合等 5 个维度 [4] 入手来构建外卖导购特征 Matrix，特征数量共 120+。

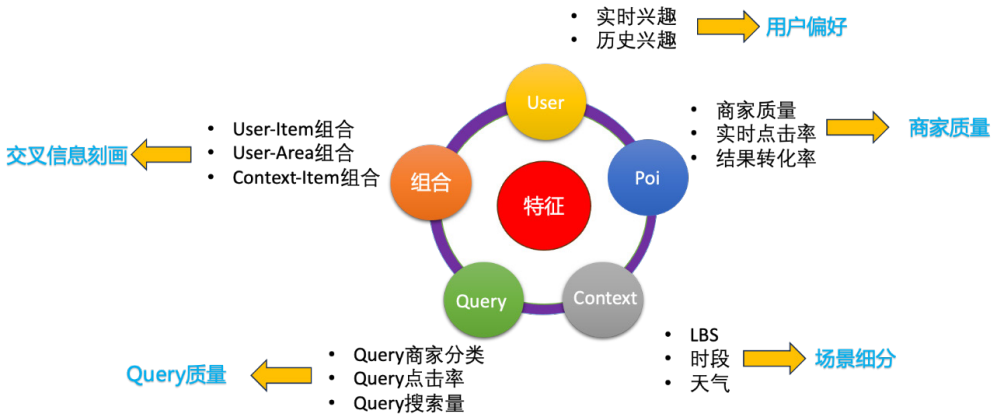


图 6　外卖搜索导购特征 Matrix

- **User 维度：**该维度包括用户在不同时间窗口内的不同行为统计特征，比如时间窗口可以选择 1 天 /7 天 /30 天 /60 天 /180 天等，行为特征主要有搜索行为、点击行为以及购买行为等，以此来表征用户的兴趣。
- **POI 维度：**主要是 POI 在不同时间窗口的行为统计特征来表征 POI 的热度，时间窗口和行为类型参考 User 维度的设定，除此之外，还包括 POI 的商家分类等基础属性信息。
- **Query 维度：**主要包括 Query 的搜索量、点击率以及成单率等统计特征，此外还包括 Query 的商家分类及分词信息。
- **Context 维度：**主要是时间，如时段、星期几、节假日；场景特征，如当前所处场景（框内、框下、搜索发现）等等。
- **组合维度：**Context-Item 组合维度，组合 Areaid_x_item 以及 Period_x_item 的交叉维度，挖掘不同时段下 Item 的曝光点击率等特征；User-Item 组合维度，User 在不同时间窗口内对 Item 搜索、点击等行为次数等。

除了上面提到的 5 种维度的特征外，为了配合异构行为序列建模，我们也设计了用户的异构序列特征，该部分在 3.2.3 中详细介绍。

3.2.2 样本建设

“无效”样本过滤

在用户点外卖时，打开美团外卖 App 会默认访问外卖首页，同时会默认曝光词服务。

因此，无论用户是否想浏览词服务，前端都会对其进行曝光，同时在引导页的搜索发现也是同理。而传统的点击率模型的样本标注会将曝光未点击 Item 标注为负样本，因此对那些并不是想浏览词服务的用户（如进入 App 就直接去了主列表或者金刚页），就会引入有偏负样本，即把用户可能只是因为没注意到词服务而没点击的样本，都算成了用户不想点击的负样本，从而使得训练与评估样本集不能精确反映用户的兴趣偏好。

为了消除上面提到的有偏负样本的影响，我们基于搜索导购词服务的实际场景特点（框内词、框下词在首页默认曝光及搜索发现在引导页默认曝光），我们进行了“无效”样本过滤，对于是否“无效”的定义，我们进行了以下尝试和探索：

- **User 维度的“无效”样本过滤：**通过统计用户在过去的历史 7 天内对导购有无行为，如果有过点击行为，我们认为这部分用户是可以感知到导购功能的，保留其所有的请求数据，而对 7 天在导购无行为的用户，如果本次请求中用户没有任何点击，我们会将本条请求的样本全部过滤掉。具体的，在首页如果用户在当前请求对框内词和框下词都没有点击，我们会将本次请求的样本全部过滤，同理该策略也会使用在引导页的搜索发现样本上。
- **Session 维度的“无效”样本过滤：**如果当前 Session 内用户在导购内有点击行为，我们会将 Session 内所有请求样本都保留，反之我们会认为本次 Session 用户没有感知到导购，会将本次 Session 内所有请求样本视为“无效”样本。
- **QV 维度的“无效”样本过滤：**如果当前请求用户没有任何点击行为，我们会将本次请求的样本视作“无效”样本。具体的，在首页如果当前请求框内词和框下词都没有点击的样本，我们会将本次请求的样本过滤，同理该策略也会使用在引导页的搜索发现样本上。

可以看到，User 维度 ->Session 维度 ->QV 维度是过滤条件依次收紧的过程，通过下表的离线实验我们发现，对于三个场景 GAUC 的表现上，QV 维度的“无效”样本过滤效果最佳。

“无效”样本过滤	框内词 CTR-GAUC	框下词CTR-GAUC	搜索发现 CTR-GAUC
+ User维度“无效”样本过滤后训练的模型 vs 全量样本训练的模型	+0.86pp	+1.28pp	+1.32pp
+ Session维度“无效”样本过滤后训练的模型 vs 全量样本训练的模型	+1.69pp	+1.94pp	+2.75pp
+ QV维度“无效”样本过滤后训练的模型 vs 全量样本训练的模型	+2.04pp	+3.41pp	+3.84pp

我们将离线最优的版本（QV 维度“无效”样本过滤）在线上进行了实验，效果如下表所示：

项目内容	导购服务			流量转化效率		
	框内词 UV_CTR	框下词 UV_CTR	搜索发现 UV_CTR	搜索 DAU	搜索UV-CXR	搜索UV-RPM
QV维度“无效”样本过滤优化	+5.19%	+7.85%	+5.79%	+0.11%	+0.17%	+0.22%

搜索多场景样本归因

在外卖搜索导购中，每天有数百万用户浏览了某个 Query 之后虽然没有在导购对其点击 / 下单，却在搜索其他入口（SUG/ 主动搜索）对其点击 / 下单。例如在图 7 的左图中，框下词模块为用户推荐出“川菜”，但是用户没有使用框下词，而且去主动搜索“川菜”，进而在 SUG/ 主动搜索下单，同理在图 7 的搜索发现模型也存在相似的情况。

然而，在导购精排模型的样本中会将在本场景曝光未点击的样本视为 CTR/CXR 的负样本，因此模型在训练时会收到这些 Query 不符合用户点击 / 成单偏好的监督信号，从而学着去将它们排得靠后，造成模型优化的方向与实际中这些 Query 满足用户点击 / 成单偏好的事实相反的偏差。因此，我们需要解决这一 Query 推荐场景下的样本归因偏差问题，来提升外卖搜索场景下的千人成交率。



图 7　搜索多场景样本归因（左图为框下词归因，右图为搜索发现归因）

为解决这一问题，关键在于要让模型具有预估导购内点击 / 成单能力之外，还要有预估用户搜索全局点击 / 成单的能力。为此，我们将模型中的 CTR/CXR 的预估目标，由曝光后导购入口内点击 / 成单概率升级为曝光后搜索全局点击 / 成单概率这样一种可落地的实现方式。

基于这一方案，我们在构造样本时，抓住关键细节，为严格保证搜索全局点击 / 成单 Label 的物理含义，对于导购入口曝光未点击 / 成单，之后搜索其他入口（如 SUG/ 主动搜索）有点击 / 成单的情况，只在样本中将最接近成交时间戳的导购入口曝光的未点击 / 成单负样本修正为点击 / 成单正样本。经数据分析，我们发现上述搜索全局成交归因建模方式可对每天的训练样本扩充约 30% 的点击 / 成单正样本，有较为充足的影响面。

由于样本归因偏差建模修改了 Label，因此需要一个新的离线评估准则。为了评估模型的有效性，我们回归问题本质，仍抓住模型是否做到了保证对本入口点击 / 成单样本的预估能力的基础上，具有了更强的搜索全局点击 / 成单预估能力的关键要点，来构建评估体系。

具体地，我们交叉评估基线模型与搜索多场景样本归因纠偏模型的 CTR/CXR 预估，分在导购入口内点击 / 成单样本及在搜索全局点击 / 成单样本上的 GAUC。结果显示，搜索多场景样本归因模型，在保证对导购入口点击 / 成单样本的排序能力基本不降的同时，显著提升了对搜索全局点击 / 成单样本的排序能力，进而证明模型的有效性。

全网成交纠偏模型的在离线效果如下表所示：

项目内容	未归因数据：导购曝光+导购点击+导购成单						归因数据：导购曝光+搜索点击+搜索成单					
	框内词		框下词		搜索发现		框内词		框下词		搜索发现	
	CTR-GAUC	CXR-GAUC	CTR-GAUC	CXR-GAUC	CTR-GAUC	CXR-GAUC	CTR-GAUC	CXR-GAUC	CTR-GAUC	CXR-GAUC	CTR-GAUC	CXR-GAUC
全网成交纠偏模型 vs 基线模型	+0.05pp	-0.04pp	-0.02pp	+0.05pp	+0.01pp	-0.03pp	+0.40pp	+0.61pp	+1.68pp	+1.76pp	+2.86pp	+2.34pp

我们将全网成交纠偏模型在线上进行了实验，效果如下表所示：

项目	导购服务-词服务						搜索整体流量转化效率		
	框内词 UV_CTR	框内词 UV_CXR	框下词 UV_CTR	框下词 UV_CXR	搜索发现 UV_CTR	搜索发现 UV_CXR	搜索DAU	搜索UV-CXR	搜索UV-RPM
全网成交纠偏模型	+1.06%	+1.83%	+2.32%	+1.46%	+3.62%	+2.97%	+0.04%	+0.10%	+0.18%

3.2.3 用户实时异构行为序列建模

用户兴趣建模一直被业界认为是实现排序结果“千人千面”的关键，是个性化排序系统中必不可少的一个模块。虽然用户兴趣建模已有诸多研究，但结合外卖搜索业务本身的LBS 特色进行用户兴趣建模，仍是一项不小的挑战。目前，词服务模块在用户行为兴趣向量上的探索还处于空白阶段，于是我们决定对词服务模型进行用户兴趣建模，以此来提升模型捕获用户兴趣的能力。

搭建导购用户兴趣建模五维体系：如图 8 所示，我们搭建了理想版的导购用户兴趣建模五维体系，形成了“行为周期、行为情景、行为模式、行为表示、兴趣抽取”用户兴趣五维建模体系迭代认知 [5]。

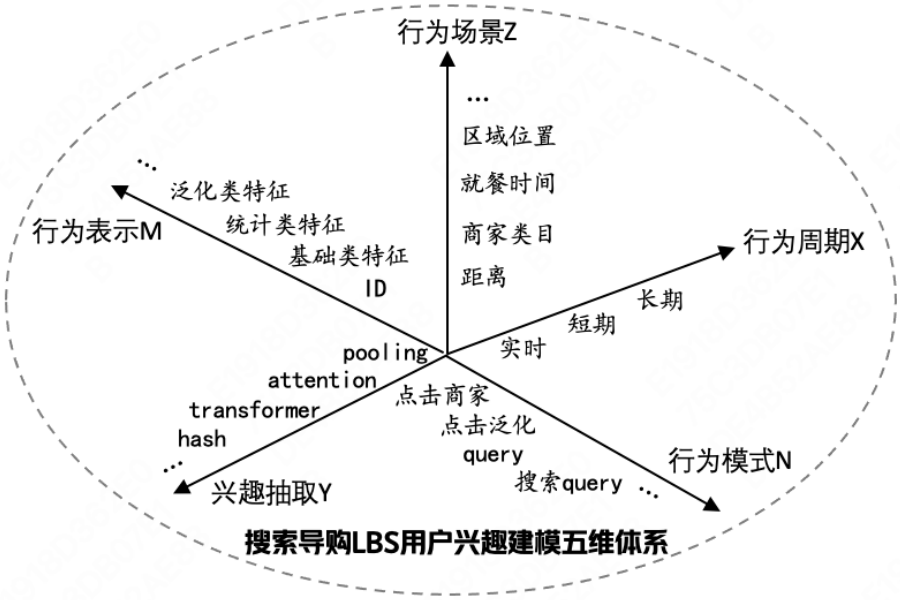


图 8　用户兴趣建模五维体系

首先，用户兴趣是随着时间动态变化的，为了更好地理解用户的兴趣偏好和需求，我们可以按照行为周期维度将用户的全周期兴趣划分为用户实时兴趣、短期兴趣和长期兴趣。

其次，外卖 LBS 特性导致用户在不同的 LBS 情景下具有不同的行为偏好，归结为行为情景维度。

再次，结合外卖本身业务特色，用户的主要行为包括搜索 / 浏览 / 点击 / 加购 / 下单等行为，将这一系列动作归结为行为模式维度。

最后，业界用户兴趣建模同样面临的问题，使用哪些行为特征来表示用户兴趣，将这归结为行为表示维度，以及如何对用户兴趣进行抽取得到最终兴趣向量，归结为兴趣抽取维度。可以说，“行为表示”决定了用户兴趣表示的上限，“兴趣抽取”决定了逼近用户兴趣表示上限的可能。

实时异构行为序列选取：与传统的商家推荐建模方式不同 [6]，我们在建模用户对商家的行为序列的同时，根据搜索导购的场景特点，将用户的实时搜索序列和商家泛化 Order-Query 等异构序列联合建模，以保留场景特色并取得更好的效果。为了帮助模

型从行为序列中更精准地捕捉到用户的兴趣，我们引入了序列的 Sideinfo 信息来辅助学习，在行为表示上，我们引入点击商家 ID 和 Query ID 以及基础类特征：商家名、商家 Tag、Query 分类、Query 分词等特征；在行为情景上，我们将 LBS 特性作为上下文特征：就餐时间、区域、距离。

异构行为序列建模：用户行为序列蕴含了非常丰富的用户兴趣和偏好信息，目前主流方法是将用户行为序列中的 Item 进行 Pooling 后的结果来表征用户兴趣，这种方式的假设是用户行为序列内的 Item 对用户兴趣的贡献是等价的，但容易存在噪声。Guorui Zhou 等提出的 DIN 模型 [7]，认为用户行为序列内不同的 Item 对用户当前的兴趣，贡献是不同的，通过 Attention 来刻画这种贡献。因此，我们将模型从 DNN 升级为 DIN，模型结构如下图 9 所示：

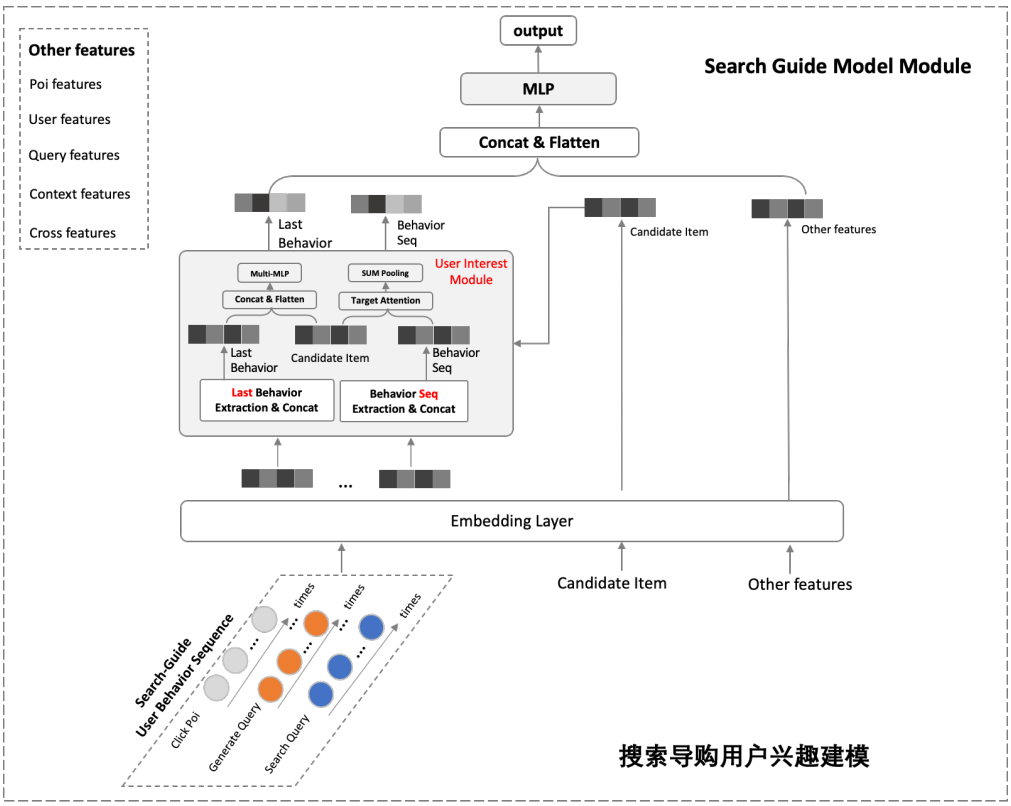


图 9　搜索导购用户兴趣建模

我们将收集到的用户行为序列以及其对应的 Sideinfo 喂到如下图所示的模型中，Behavior Seq 与 Target Item 进行 Target Attention 后进行聚合，得到用户的兴趣

序列表示，同时由于在早期的尝试中得到了用户 Last Click 行为十分高效的结论，所以我们将用户三个序列的 last 行为还进行了单独建模，将三个序列的 Last 行为及 Sideinfo 与 Target Item 进行聚合，然后经过一个 MLP，得到用户最实时的兴趣表示。将上述的两个兴趣表示与 Target Item 以及 Other features 聚合后经过 MLP 后，得到 Output。

我们对用户实时异构行为序列建模进行了多组离线的消融实验，离线效果如下表所示：

用户实时异构行为序列建模	框内词 CTR-AUC	框内词 CTR-GAUC	框下词 CTR-AUC	框下词 CTR-GAUC	搜索发现 CTR-AUC	搜索发现 CTR-GAUC
+ 24h内的商家点击序列 vs 无序列建模的模型	+1.06pp	+1.04pp	+1.71pp	+1.19pp	+1.34pp	+1.07pp
+ 24h内泛化Query序列 vs 无序列建模的模型	+0.80pp	+0.89pp	+1.08pp	+0.74pp	+0.79pp	+0.94pp
+ 24h内搜索Query序列 vs 无序列建模的模型	+0.92pp	+0.84pp	+0.95pp	+0.79pp	+0.82pp	+0.84pp
+ 24h内3个行为序列 vs 无序列建模的模型	+2.62pp	+2.34pp	+3.45pp	+2.39pp	+2.83pp	+2.59pp
+ 3个Last行为建模 vs 无序列建模的模型	+1.59pp	+1.64pp	+1.35pp	+1.72pp	+1.69pp	+1.54pp
+ 3个序列+3个Last行为vs 无序列建模的模型 **（推全版本）**	**+3.62pp**	**+3.62pp**	**+3.62pp**	**+3.62pp**	**+3.62pp**	**+3.62pp**

我们将离线最优的版本（3 个序列 +3 个 Last 行为）在线上进行了实验，效果如下表所示：

项目内容	导购服务			流量转化效率		
	框内词 UV_CTR	框下词 UV_CTR	搜索发现 UV_CTR	搜索 DAU	搜索UV-CXR	搜索UV-RPM
用户实时异构行为序列建模	+15.18%	+18.37%	+15.23%	+0.24%	+0.25%	+0.30%

3.2.4 导购联动搜索结果页全链路多目标效率优化

导购场景作为引流入口，外卖用户在导购的成单路径可以抽象为[1]导购 Query 曝

光 ->[2] 导购 Query 点击 ->[3] 搜索结果页曝光 ->[4] 搜索结果页点击 ->[5] 搜索结果页加购 ->[6] 搜索结果页成单。如果优化目标仅考虑导购场景内的信号“场景内的 CTR 作为目标”，却并未直接面向搜索场景的整体效率，即联动搜索结果页的用户反馈来优化，存在局部最优的问题。为此，我们引入词服务下游场景，即搜索结果页的用户反馈信号，并进行词服务 PV 维度的归因，构建 CXR 目标并进行建模。

基于以上背景，我们将单目标的导购精排模型升级为多目标模型，增加搜索结果页成单的目标，基于 Share-Bottom 的结构进行多目标联合建模。为了建模两个目标的共性，同时考虑到 CXR 是在 CTR 条件下发生的行为，于是我们采用了 Bayesian 的思想，将 CTR 网络的最后一层的输出 Concat 到 CXR 的上面，再计算 CXR 的 Logit。模型结构如图 10 所示：

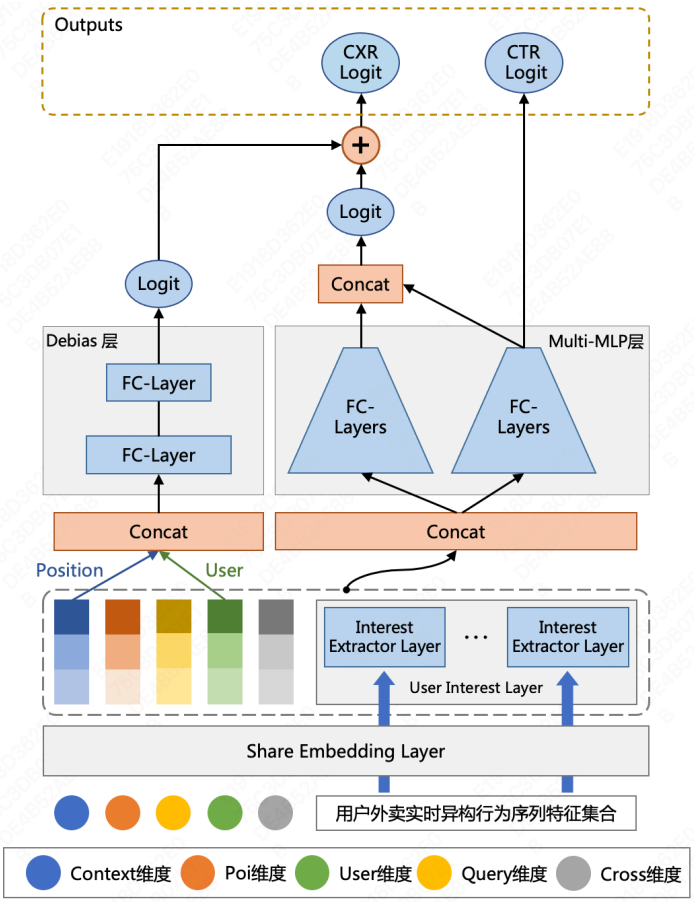


图 10　导购联动搜索结果页多目标建模

在机制上，我们通过机制公式融合 CTR 和 CXR 两个目标 $S \text { core }=w{1} * p C T R+w{2} * p C X R$。在通过多组实验验证，当时，在线效果达到最优，后续计划尝试离 / 在线多目标寻参。

我们对导购联动搜索结果页全链路多目标效率优化进行了多组离线的消融实验，离线效果如下表所示：

导购联动搜索结果页全链路多目标效率优化	框内词				框下词				搜索发现			
	CTR-AUC	CTR-GAUC	CXR-AUC	CXR-GAUC	CTR-AUC	CTR-GAUC	CXR-AUC	CXR-GAUC	CTR-AUC	CTR-GAUC	CXR-AUC	CXR-GAUC
+ 多目标优化 vs 单目标优化的模型	+1.28pp	+0.98pp	-	-	+2.27pp	+1.69pp	-	-	+1.38pp	+1.50pp	-	-
+ 多目标优化+CTR输出Concat到CXR vs 多目标优化（推全版本）	+0.38pp	+0.42pp	+0.55pp	+0.51pp	+0.68pp	+0.42pp	+0.62pp	+0.39pp	+0.31pp	+0.29pp	+0.48pp	+0.32pp

我们将离线最优的版本（多目标优化 +CTR 输出 Concat 到 CXR）在线上进行了实验，效果如下表所示：

项目内容	导购服务						流量转化效率		
	框内词 UV_CTR	框内词 UV_CXR	框下词 UV_CTR	框下词 UV_CXR	搜索发现 UV_CTR	搜索发现 UV_CXR	搜索DAU	搜索UV-CXR	搜索UV-RPM
导购联动搜索结果页全链路多目标效率优化	+5.96%	+7.25%	+7.86%	+12.80%	+4.55%	+8.01%	+0.07%	+0.18%	+0.10%

3.2.5 基于自监督学习的全场景统一模型

随着“陪伴型”导购场景在搜索各个模块的不断落地，Query 推荐场景已经扩展到了五个。然而，单独迭代每个场景的模型面临着如下挑战：

- **小场景和新场景训练数据不足。**多个 Query 推荐场景中，对于新场景（例如框内词智能刷新、猜你想搜）训练数据比较少，直接用单个场景反馈数据训练，会导致过拟合问题；
- **多个场景统一建模。**能否同时利用多个场景的数据，建模多个场景的共同点、不同点，提升多场景排序模型的效率呢?
- **反馈闭环问题** [8]。目前这种偏向效率的训练方式，会导致排序系统展示出来的 item 通常具有高频的特点，很多长尾的 item 没有曝光的机会，马太效应严重，不利于生态的长期健康成长；

- **降本增效。**对于多个相似场景，如果每个场景都单独迭代一个模型耗时耗力，同时也需要大量的机器资源。

为了解决以上问题，我们采用了一种基于自监督学习的全场景统一建模的方法[9]。这种方法采取了 Pre-training 和 Fine-tuning 两个阶段的方式。通过数据统计，我们发现用户历史搜索行为前后具有关联性。

因此，在预训练阶段，我们借鉴了语言模型 BERT[10] 中 Next Sentence Prediction（NSP）和 GPT[11] 中 Next Word Prediction（NWP）任务的思想，利用用户历史行为数据和用户主动输入的信息（SUG 以及主动搜索数据），学习预测用户下一个输入的 Query（Next-Query Prediction）。在 Fine-tuning 阶段，我们使用各场景的隐式反馈数据来 Fine-tuning 得到全场景统一模型。

由于在预训练阶段让模型见到了更多其他场景的 Query，我们可以在一定程度上缓解反馈闭环的问题。同时，在 Fine-tuning 阶段使用了全场景数据并对场景信息进行建模，所以我们的多个场景统一模型可以在缓解小场景和新场景过拟合问题的同时，还可以兼顾建模出不同场景的共性和特性。下面将分别介绍 Pre-training 和 Fine-tuning。

Pre-training 预训练：因为用户主动输入的 Query 正是 Query 推荐排序模型理想的结果，所以我们基于用户历史行为，预测下一个用户主动输入的 Query。下面我们分别从样本构建、特征 Matrix 以及预训练流程三部分介绍 Pre-training 的过程。

- **样本构建：**我们使用 SUG 以及主动搜索场景内数据，采用类似于 3.2.2 的思想。例如，当用户输入“和府捞面”时，SUG 曝光了该 Query，但是用户没有点击，而是直接点击搜索按钮主动搜索“和府捞面”，对于这类 SUG 曝光未点击，但是发生了主动搜索的情况，我们将主动搜索的点击数据归因到 SUG 数据中。



图 11　主动搜索点击行为归因到 SUG 场景案例

- **特征 Matrix：**在 Pre-Train 的模型中，挑选出 SUG 和词服务共用的特征进行预训练，包括用户兴趣建模方面的序列特征以及用户画像特征等。在 Fine-Tune 阶段，我们会预加载 Pre-Train 的特征，然后增加了 Dense Features 和 Scenario Features。
- **预训练流程：**我们设计的预训练任务是基于用户历史行为，预测下一个用户主动输入的 Query，因为用户主动输入的 Query 正是 Query 推荐排序模型理想的结果。由于简单地随机采样会使得负样本过于简单，因此我们采样用户主动搜索 Query 时 SUG 曝光未点击的 Query 作为困难负例 [7]。

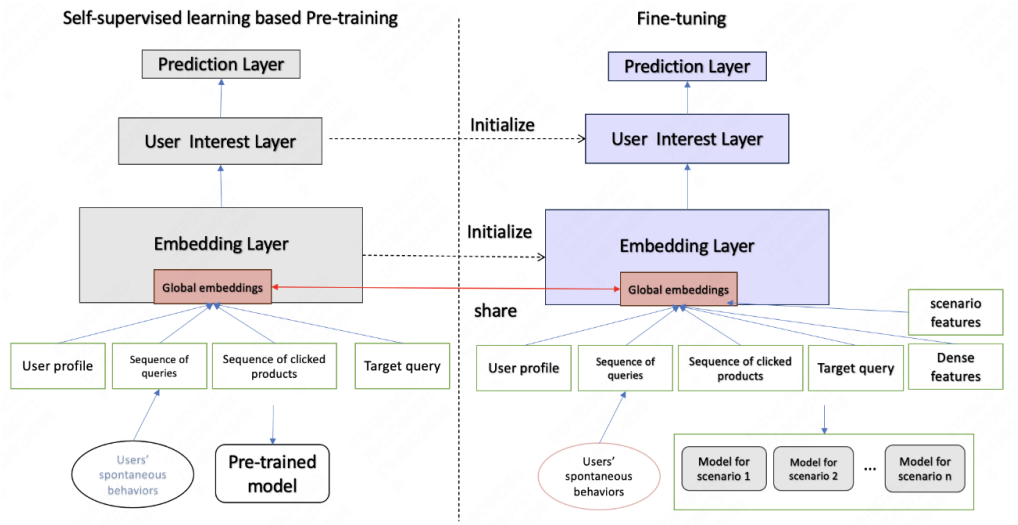


图 12　Pre-training 与 Fine-tuning 模型表示

Fine-tuning 微调：利用词服务场景数据微调预训练模型，学习多个场景之间的共性，同时加入场景信息，学习多个场景之间的不同。如上图 12 所示，由于预训练阶段没有 Dense Features 和 Scenario Features，因此 Prediction Layer 的输入维度不同，所以 Prediction Layer 的参数没有共享，两个阶段共享的是 Embedding Layer 和行为序列建模的部分。实验发现，Fine-tuning 阶段固定住 Query 和 POI Embedding 时，效果最好。因为这两个 Embedding 参数量为亿级，训练数据不充分时容易过拟合。同时，我们也通过对比，直接将用户主动行为数据融合到词服务数据中进行训练，实验结果证明了预训练的有效性。

打破反馈闭环：我们分析了在三个场景（框内词、框下词以及搜索发现）的数据，对比了有无预训练的模型半个月内给所有用户推荐出的 Unique Query 的个数，我们发现当加入预训练后，可以显著提升推荐出的 Query 多样性，三个场景分别增加 9.6%、13.6% 以及 10.8% 的 Unique Query 的个数，打破了推荐闭环。

后续计划：结合大模型预训练的全场景统一模型，借助大模型更强大的表征和泛化能力，我们计划后续利用大模型对 Query 的 Embedding 进行预训练，然后利用本场景内的数据进行 Fine-tuning，从而提高模型在导购全场景的推荐效果和泛化能力。

我们对基于自监督学习的全场景统一模型进行了多组离线的消融实验，离线效果如下表所示：

基于自监督学习的全场景统一模型	框内词				框下词				搜索发现			
	CTR-AUC	CTR-GAUC	CXR-AUC	CXR-GAUC	CTR-AUC	CTR-GAUC	CXR-AUC	CXR-GAUC	CTR-AUC	CTR-GAUC	CXR-AUC	CXR-GAUC
非预训练（直接将用户主动行为数据融合到词服务）	+0.22pp	+0.14pp	+0.23pp	+0.17pp	+0.82pp	+0.00pp	+0.48pp	+0.07pp	+0.29pp	+0.09pp	+0.27pp	+0.32pp
预训练+微调阶段更新Query和POI Embedding	+3.58pp	+2.92pp	+3.75pp	+3.04pp	+4.58pp	+2.45pp	+4.55pp	+2.68pp	+4.16pp	+3.35pp	+4.78pp	+4.62pp
预训练+微调阶段不更新Query和POI Embedding（推全版本）	**+3.83pp**	**+3.02pp**	**+3.89pp**	**+3.19pp**	**+4.77pp**	**+2.83pp**	**+4.65pp**	**+2.92pp**	**+4.32pp**	**+3.51pp**	**+4.90pp**	**+4.84pp**

我们将离线最优的版本（预训练 + 微调阶段不更新 Query 和 POI Embedding）在线上进行了实验，效果如下表所示：

项目	导购服务-词服务						搜索整体流量转化效率		
	框内词 UV_CTR	框内词 UV_CXR	框下词 UV_CTR	框下词 UV_CXR	搜索发现 UV_CTR	搜索发现 UV_CXR	搜索DAU	搜索UV-CXR	搜索UV-RPM
基于自监督学习的全场景统一模型	+3.86%	+6.03%	+5.20%	+8.14%	+6.54%	+7.55%	+0.08%	+0.05%	+0.21%

4. 总结与展望

本文介绍了我们在外卖智能陪伴型导购的探索与尝试，其中主要包括：

1. 借助 Alita 端智能的意图感知能力，智能判断何时需要针对性的刷新，将被动型导购变成主动型导购。
2. 为了追求和线下导购一样一对一极致的实时性的个性化体验，我们在对模型引入实时异构行为序列进行建模的同时，打造端云结合排序模型，捕获用户实时多变的兴趣。
3. 导购的优化除了满足用户体验之外，还需要为整体的转化目标负责，我们对全场景数据进行统一建模，打造基于自监督学习的统一模型，在对样本进行“无效”过滤以及归因的基础上，同时联动搜索结果页进行全链路的多目标效率优化。

目前智能陪伴型导购已经在外卖搜索全量，累计收获了如下收益：

- 搜索 DAU +1.02%、搜索 UV_RPM +1.70%。
- 搜索框内词 UV_CTR +187%、搜索框下词 UV_CTR +141%、搜索发现词 UV_CTR +121%。

未来，我们将从以下方向进行探索优化：

1. **智能刷新赋能更多场景：**在搜索结果页以及外卖首页 Feed 页，利用端智能的意图感知能力，判断用户兴趣发生收敛的时机，利用导购 Query 推荐的能力，响应用户更细粒度的需求表达，满足用户找同类商家、找同品牌商家、找用户感兴趣的商品或者品类的需求。
2. **借助大模型的能力赋能导购模型：**结合大模型预训练的全场景统一模型，借助大模型更强大的表征和泛化能力，我们计划后续利用大模型对 Query 的 Embedding 进行预训练，然后利用本场景内的数据进行 Fine-tuning，从而提高模型在导购全场景的推荐效果和泛化能力。

5. 本文作者

翔锟、营飞、关璐、泽钰、张聪、晓鹏、北海等，来自美团到家研发平台 / 搜索推荐技术部。
运祥、周全、张晶等，来自到美团外卖事业部 / 产品部。

6. 参考文献

[1] Gong Y, Jiang Z, Feng Y, et al. EdgeRec: recommender system on edge in Mobile Taobao[C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020: 2477-2484.
[2] 端智能在大众点评搜索重排序应用实践
[3] 交互式推荐在外卖场景的应用
[4] 美团外卖推荐情境化智能流量分发的实践与探索
[5] Zhou G, Zhu X, Song C, et al. Deep interest network for click-through rate prediction-[C]//Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 1059-1068.
[6] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the human-recommender system feedback loop in collaborative filtering. In WWW’19. 645–651.
[7] Gu Y, Bao W, Ou D, et al. Self-Supervised Learning on Users’ Spontaneous Behaviors for Multi-Scenario Ranking in E-commerce[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 3828-3837.
[8] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018).
[9] Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. “Improving language understanding by generative pre-training.” (2018).

招聘信息

搜索推荐算法工程师

1. 了解大规模深度学习、图学习等技术，利用注意力机制、记忆网络、关系网络等模块，从跨多个时空 VV 场景的海量数据中理解用户需求、发掘用户兴趣，优化点击率、转化率模型，为用户展示更合适、更感兴趣的美食和商品。
2. 了解强化学习、可解释深度学习、多模态学习、多目标优化等技术，优化重排、混排模型，智能调控流量分发，优化平台生态，实现消费者和商家共赢。
3. 能够使用知识图谱、计算视觉、自然语言生成等技术，面向用户兴趣，帮助商家自动化智能生成展示内容和文案，提升推广效率。
4. 追踪并研究人工智能前沿技术，探索技术在零售、医疗电商场景的应用。

感兴趣的同学可以将简历发送至：libeihai02@meituan.com。期待与大家一起共事，共创未来。

大规模食品图像识别：T-PAMI 2023 论文解读

作者：致岭 丽萍 晓明

1. 引言

视觉智能部与中科院计算所于 2020-2021 年度展开了《细粒度菜品图像识别和检索》科研课题合作，本文系双方联合在 IEEE T-PAMI2023 发布论文《Large Scale Visual Food Recognition》(Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, Shuqiang Jiang*) 的解读。IEEE T-PAMI 全称为 IEEE Transactions on Pattern Analysis and Machine Intelligence，是模式识别、计算机视觉及机器学习领域的国际顶级期刊，2022 年公布的影响因子为 24.314。



图 1　Food2K 数据集

食品计算 [1] 因能够支撑许多食品相关的应用得到越来越广泛的关注。食品图像识别作为食品计算的一项基本任务，在人们通过辨认食物进而满足他们生活需求方面发挥着重要作用，如食品营养理解 [2,3] 和饮食管理 [4] 等。此外，食品图像识别是细粒度视觉识别的一个重要分支，具有重要的理论研究意义。

现有的工作主要是利用中小规模的图像数据集进行食品图像识别，如 ETH Food-101[5]、Vireo Food-172[6] 和 ISIA Food- 500[7]，但是它们的数据规模不足以支撑更复杂更先进的食品计算模型的建立。考虑到大规模数据集已成为许多常规图像分类和理解任务发展的关键推动因素，食品计算领域也迫切需要一个大规模的食品图像数据集来进一步支撑各种食品计算任务，如跨模态食谱检索和生成 [8,9]。因此我们构建了一个新的大规模基准数据集 Food2K。该数据集包含 1,036,564 张食品图像和 2,000 类食品，涉及 12 个超类（如蔬菜、肉类、烧烤和油炸食品等）和 26 个子类别。与现有的数据集相比，Food2K 在类别和图像数量均超过其一个数量级。除了规模之外，我们还进行了严格的数据清理、迭代标注和多项专业检查，以保证其数据的质量。

在此基础上，我们进一步提出了一个面向食品图像识别的深度渐进式区域增强网络。该网络主要由渐进式局部特征学习模块和区域特征增强模块组成。前者通过改进的渐进式训练方法学习多样互补的局部细粒度判别性特征（如食材相关区域特征），后者利用自注意力机制将多尺度的丰富上下文信息融入到局部特征中，进一步增强特征表示。本文在 Food2K 上进行的大量实验证明了所提出方法的有效性，并且在 Food2K 上训练的网络能够改善各种食品计算视觉任务的性能，如食品图像识别、食品图像检索、跨模态菜谱 - 食品图像检索、食品检测和分割等。我们期待 Food2K 及在 Food2K 上的训练模型能够支撑研究者探索更多的食品计算新任务。本论文相关的数据集、代码和模型可从网站下载：http://123.57.42.89/FoodProject.html。

2. Food2K 数据集

Food2K 同时包含西方菜和东方菜，在食品专家的帮助下，我们结合现有的食品分类方法建立了一个食品拓扑体系。Food2K 包括 12 个超类（如“面包”和“肉”），每个超类都有一些子类别（如“肉”中的“牛肉”和“猪肉”），每种食品类别包含许多菜肴（如“牛肉”中的“咖喱牛肉”和“小牛排”），如下图 2 所示：

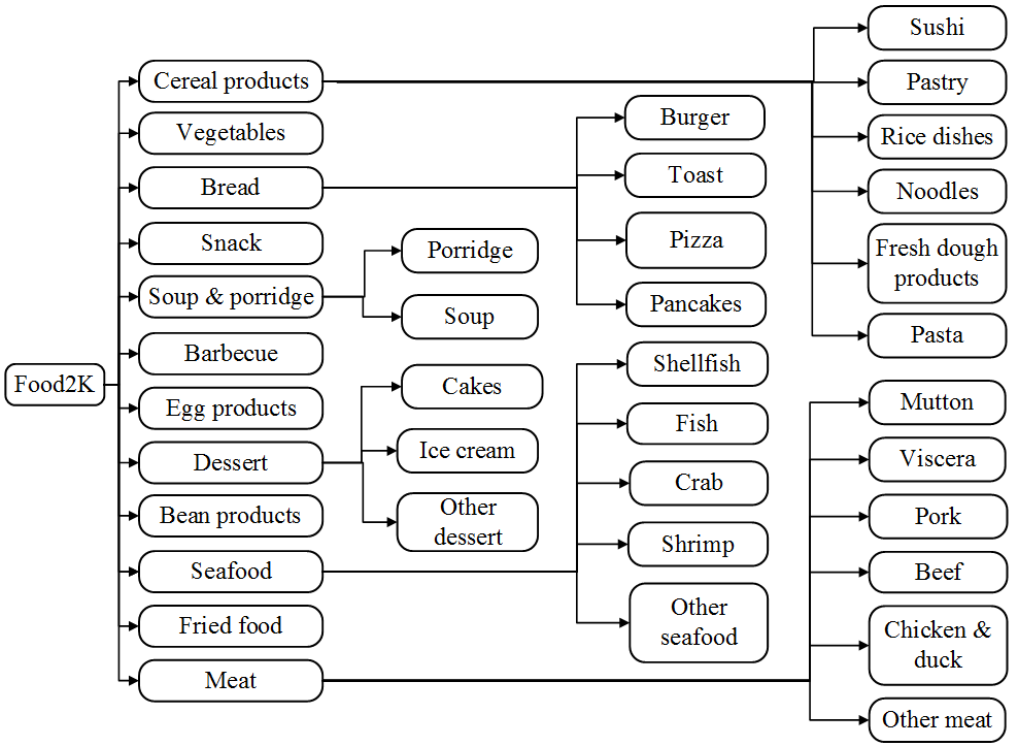


图 2　Food2K 分类体系

下图 3 展示了每个食品类别的图像数量，Food2K 中每个类别的图像数量分布在 [153,1999] 之间不等，呈现出明显的长尾分布现象，与类别不平衡性。

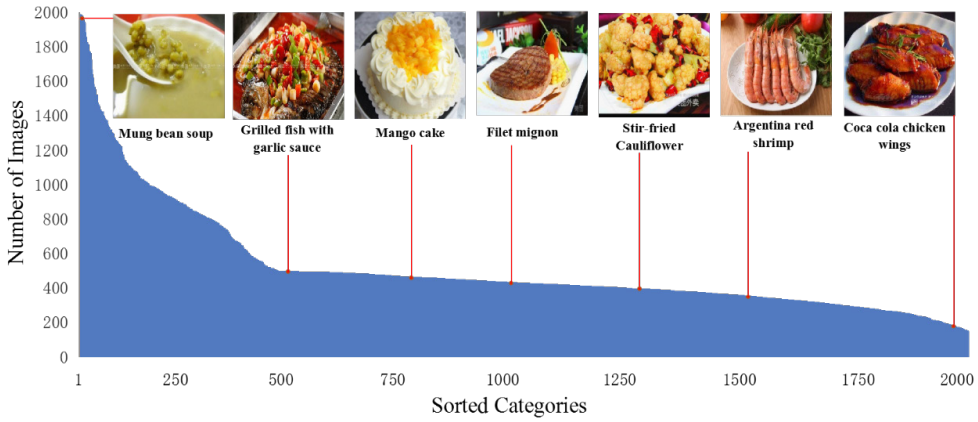


图 3　Food2K 各类别图像数量分布

下图 4 展示了 Food2K 与现有食品图像识别数据集的图像数量对比，可以看到 Food2K 在类别和图像数量上都远超过它们。

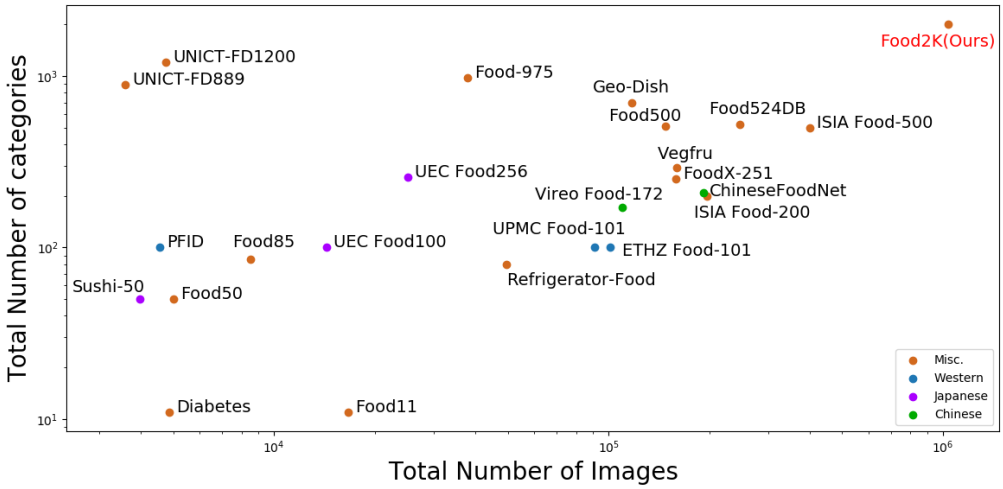


图 4　Food2K 与其它食品识别数据集图像数量对比

除此之外，Food2K 还具有以下特征：

1）**Food2K 涵盖了更多样化的视觉外观和模式。**不同食材组合、不同配饰、不同排列等都会导致同一类别的视觉差异。举例来说，新鲜水果沙拉因其不同的水果成分混合而呈现出不同的视觉外观。这些食品的独特特征导致了更高的类内差异，使大规模的食品图像识别变得更加困难。

2）**Food2K 包含更细粒度的类别标注。**以“Pizza”为例，一些经典的食品数据集，如 Food-101，只有较粗粒度的披萨类。而 Food2K 中的披萨类则进一步分为更多的类别。不同披萨图像之间的细微视觉差异主要是由独特的食材或同一食材的粒度不同引起的，这也导致了识别的困难。所有这些因素使 Food2K 成为一个新的更具挑战性的大规模食品图像识别基准，可以视为食品计算领域的“ImageNet”。

3. 方法

食品图像识别需要挖掘食品图像的本身特点，并同时考虑不同粒度的图像特征进行识别。通过观察我们发现，食品图像有着明显的全局特征和局部特征。

首先，食品图像明显有着全局的外观、形状和其他结构方面的特征，且该特征存在较大的类内差异。如下图 5 的“奶油核桃饼”明显有着变化不一的形状，炒面有着丰富多样的纹理。虽然当前已经有很多方法来解决这一问题，但大多数方法主要集中在提取某种

类型的特征，而忽略了其他类型的特征。



图 5　食品图像的全局特征与局部特征

其次，食品图像中有细微差别的细节信息，部分细节信息是其关键的局部特征。在许多情况下，现有的神经网络无法很好地挖掘那些具有判别性的细节特征。如图 5 中第 3 栏所示，仅仅依靠全局特征是不足以区分“玉米羹”和“鸡蛋羹”，必须进一步挖掘其食材信息的不同（图中黄色框内）。因此，如何更好地挖掘食品图像的全局特征和局部特征，对于提升食品图像特征表达能力来说尤为重要。

第三，如下图 6 所示，不同的食材在不同的食品类别中所占的权重也是不一样的。“香菜”在“香菜”拌牛肉中是一个关键性食材，必不可少，但是在“老醋海蜇”这道菜中仅仅是一个配料，并不总是出现在该类别的所有图像中。因此需要挖掘局部特征之间的关系，突出重点局部特征。进而提高食品识别的准确率。



图 6　不同食材在不同的食品图像中所占比重不同

针对上述这些特点，本文设计了深度渐进式区域特征增强网络，它可以共同学习多样化且互补的局部和全局特征。该网络结构如下图 7 所示，该网络主要由三部分组成：全局特征学习模块、渐进式局部特征学习模块和区域特征增强模块。

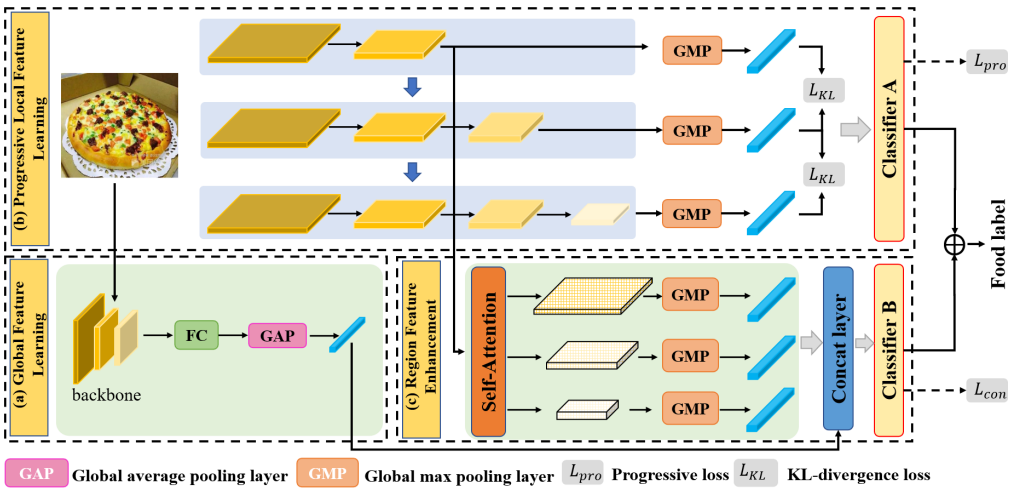


图 7　深度渐进式区域增强网络框架图

其中，渐进式局部特征学习主要采用渐进式训练策略来学习互补的多尺度细粒度的局部特征，例如不同的食材相关信息。区域特征增强使用自注意力机模块，将更丰富的多尺度上下文信息合并到局部特中，以增强局部特征表示。然后，我们通过特征融合层将增强的局部特征和来自全局特征学习模块的全局特征融合到统一的特征中。

此外，在模型训练时，本文逐步渐进式地训练网络的不同阶段，最后将训练整个网络，并在此过程中引入 KL 散度以增加各个阶段之间的差异性，以捕获更丰富多样化的局部信息。在推理阶段，考虑到每个阶段的输出特征和融合后的特征之间的互补性，我们将它们的预测结果结合起来得到最终分类得分。接下来，本文将详细介绍各个模块的计算原理。

全局 – 局部特征学习

食品识别可以看作是一个层次化的任务，在不同超类下的食品图像有着明显可分的视觉差异，因此可以基于他们的全局特征进行识别。

但是在同一超类下，不同子类的食品图像之间的差异却非常小。因此食品识别需要同时学习食品图像的全局特征和局部特征。因此，我们提取并融合了这两个特征，此过程可以使用两个子网络分别提取食品图像的全局和局部特征。这两个子网络可以是两个独立的网络，但是这部分工作为了提高效率并减小模型参数，本文中两个子网络共享基础网络的大部分层。

全局特征学习

对于全局特征学习，本文取最后一个卷积层的输出，使用全局平均池化（Global Average Pooling, GAP）来提取全局特征 f_{Glo}：

$$f_{Glo} = \mathbf{GAP}(f^g)$$

其中 f_g 为从主干网络最后一个卷积层中提取的特征。

渐进式局部特征学习

局部特征子网络旨在学习食品的区分性细粒度特征。由于食材和烹饪风格的多样性，食品图像的关键判别部分是多尺度和不规则的。作为本方法第一个创新点，我们提出了渐进式训练策略来解决这个问题。在这种策略中，我们首先训练具有较小感受野的网络较低阶段（可以理解为模型中较低的层），然后逐步扩大该局部区域范围，并引入新的层加入训练。

这种训练策略将迫使我们的模型提取更精细的判别性局部特征，例如与食材相关的特征。在此过程之后，我们从不同层提取特征以获得多尺度的特征表示。

具体来说，对于来自局部特征子网络的每一层的输出f^u，其中u是网络的不同阶段。我们使用一个卷积层和一个全局最大池化层（Global Maximum Pooling，GMP）来得到它们的局部特征向量f_{Loc}^u：

$$f_{Loc}^u = \mathbf{GMP}(f^u)$$

因此，该策略可以首先在网络较浅的层中学习更稳定的细粒度信息，然后随着训练的进行逐渐将注意力转移到在较深的层中学习粗粒度信息。具体来说，当具有不同粒度的特征被送入网络时，它可以提取出有判别性的局部细粒度特征，例如食材成分信息。然而，简单地使用渐进式训练策略不会得到多样化的细粒度特征，因为通过渐进式训练学习到的多尺度特征信息可能集中在食品图像中相似的区域。

而作为本方法第二个创新点，我们引入 KL 散度对来自不同阶段的特征进行离散化，以增加它们之间的差异性。我们通过最大化不同阶段特征之间的 KL 散度值，可以迫使网络模型在不同阶段关注不同区域的多尺度特征，这有助于网络捕获尽可能多的细节信息。

具体来说，我们将训练过程划分为S个步骤，每一步训练前$U-S+1$个阶段，其中U是网络的总阶段数。因为我们将在最后阶段（也称为特征融合阶段）将融合所有的全局和局部特征，因此总步骤数为$S+1$。在我们的方法中，对于渐进式训练的每一步训练的网络不同层的输出，使用F对该输出特征进行处理。F由一个卷积层、一个批归一化层和一个$ReLU$单元组成。然后我们可以得到局部特征表示c^u。

最后，对于来自不同阶段的这些多个局部特征，我们利用KL散度来增加阶段之间的差异，这有助于捕获尽可能多的细节。对每两个相邻阶段的输出，计算KL散度的方式如下，其中KL散度的reduction设置为batchmean。

$$L_{KL}(y_i, y_j) = \sum_{i=1}^{U} \sum_{j=U-i}^{U} y_i log(\frac{y_i}{y_j})$$

区域特征增强

不同于一般的细粒度任务中的识别对象，食品图像没有固定的语义信息。现有的大部分食品识别方法直接挖掘这些判别性局部特征，忽略了局部特征之间的关系。因此，我们采用自注意力机制来学习不同局部特征之间的关系。该策略旨在捕获特征图中同时出现的食品特征。

具体来说，借鉴于Non-Local Interaction的操作，我们在同一个特征图内操作实现自注意力增强，并输出与输入同尺度的特征图。我们首先提取所有U个阶段局部特征表示f_{Loc}^{u}，然后通过自注意力机制获得增强后的特征，计算方式如下：

$$q^{(i)} = \mathbf{Conv}(f_{Loc}^{(i)})$$

$$k^{(j)}, v^{(j)} = \mathbf{Conv}(f_{Loc}^{(j)})$$

$$S_{i,j} = \mathbf{Softmax}(q^{(i)}{k^{(j)}}^{T}/\sqrt{d_{k^{(j)}}})$$

$$\widehat{f_{IJ}} = S_{i,j}v^{(j)}$$

其中，$f_{Loc}^{(i)}$和$f_{Loc}^{(j)}$是f_{Loc}中位于i和j位置的像素特征。基于此我们可以得到增强特征$\hat{f_{IJ}}$。最后我们将这些相同大小的增强后的特征图融合起来，并使用卷积运算将它们转换为相同的维度。

最后，我们可以得到U个局部特征$f_{Loc}^{\hat{u}}$。通过这种策略我们的模型能够使局部特征实现跨空间和尺度的信息交互。在训练过程中，在获得全局特征和局部特征后，我们在特征融合阶段将它们融合后为最终表示为：

$$f' = \mathrm{Concat}(f_{Glo}, \widehat{f_{Loc}^{U-S+1}}, \ldots\ldots, \widehat{f_{Loc}^{U}})$$

训练和预测

在网络优化过程中，我们迭代更新网络的参数。首先，我们利用各 UU 个阶段的交叉熵损失，来反向传播以更新相应网络的参数。在此过程中，当前阶段的所有网络参数都将被优化，即使它们在之前的阶段已经被更新过。然后在特征融合阶段，我们利用另一个交叉熵损失函数来更新整个网络的参数。

此外，我们的网络以端到端的方式进行训练。在渐进式训练过程中，对于每个阶段的输出，我们采用以下交叉熵损失：

$$L_{pro}^{i} = -\sum_{x^m \in M} y_i^m ln y_i^m$$

其中M是一组训练数据，x^m表示m^{th}个样本，y_i^m是对应的类别标签。对于最后融合得到的特征表示，我们使用另一个交叉熵损失：

$$L_{con} = -\sum_{i=1}^{N} y'_i \times ln y'_i$$

其中y'是预测标签，N是训练样本的总数。

接下来，我们引入的KL散度作为一项损失加入到最终的损失函数中，以增加阶段之间的差异性进而以捕获更丰富的细节信息，所以最终损失函数为：

$$L = \alpha L_{con} + \beta L_{KL}$$

其中，α和β是平衡参数。

在模型推理阶段，来自最终特征融合征阶段输出的特征，被输入到分类器B中获得对应的预测输出。对于前几个阶段的输出，我们可以利用分类器A来预测相应的输出值，其中分类器A由两个全连接层和Batchnorm及Elu非线性单元组成。

考虑到不同阶段的预测输出和特征融合阶段最终输出之间的互补性，我们可以将所有输出组合在一起以提高识别性能。因此我们将所有阶段的预测得分（包括最终得分）相加，以相同的权重来预测对应的食品类别：

$$Y = \mathrm{Sum}(y', y_{U-S+1}, \ldots, y_U)$$

4. 实验

首先，我们在 Food2K 上对现有的食品识别方法和我们提出的方法进行了比较。然后，我们研究 Food2K 在食品识别、食品图像检索、跨模态菜谱 – 食品图像检索、食品检测和食品分割五个食品计算任务上的泛化能力。

方法验证与分析

本文使用在 ImageNet 上预训练的 ResNet[10] 作为基础网络。对于实验性能评估，使用 Top-1 准确率（Top-1）和 Top-5 准确率（Top-5）对食品类别识别进行评估。

在 Food2K 上的性能实验

表 1 展示了在 Food2K 上现有的食品识别方法和所提方法的性能比较。从表中可以看出，我们的方法在 Top-1 和 Top-5 准确率上分别高出主干网络（ResNet）2.24% 和 1.4%，以 ResNet101 为主干的网络超过了现有的食品识别方法，取得了较好的识别性能。同时，这也证实了结合渐进式训练策略和自注意力来增强局部特征表示的优势。

Methods	Top-1 Acc.	Top-5 Acc.
VGG16 [65]	78.96	95.26
Inception v4 [66]	82.02	96.45
ResNet50 [67]	80.79	95.74
ResNet101 [67]	81.28	95.99
ResNet152 [67]	81.95	96.57
Inception-ResNet-v2 [66]	82.07	96.74
WRN-50-2-bottleneck [68]	81.94	96.19
DenseNet161 [61]	81.87	96.53
SE-ResNeXt101_32x4d [62]	80.81	95.61
Inception v4 [66]	82.02	96.45
SENet154 [62]	83.62	97.22
NTS-NET(ResNet50) [69]	81.24	94.94
HBP(ResNet50) [70]	77.56	92.87
DCL(ResNet50) [71]	82.15	96.72
WS-DAN(ResNet50) [72]	81.37	96.27
Inception v4 @ 448px [73]	82.46	97.17
MOMN(ResNet50) [63]	80.84	96.02
PMG(ResNet50) [64]	81.29	96.12
DLA [74]	80.14	96.37
PAR-Net(ResNet101) [9]	80.93	96.61
PRENet(ResNet50)	83.03	97.21
PRENet(ResNet101)	**83.75**	**97.33**

表 1　现有方法在 Food2K 上性能对比

在 Food2K 上的消融实验

本文在消融实验中主要探讨了以下几个问题：

（1）网络不同组件的有效性：从下图 8(a) 中可以看出，渐进式策略（PL）的引入能够带来识别性能增益，且与区域特征增强（RE）相结合后进一步提高了性能。这说明我们提出的 PL+RE 的学习策略能够有效地学到更具判别性的局部特征。

（2）渐进式学习阶段的数量 U：从下图 8(b) 中可以看出，当 U 从 1 到 3 时，我们的方法在 Food2K 上分别取得了 81.45%、82.11% 和 83.03% 的 Top−1 分类准确率。模型的分类性能连续提高了 0.66% 和 0.92%。然而，当 U = 4 时，精度开始下降，可能的原因是浅层网络主要关注类别无关的特征。

（3）不同学习阶段的效果：为了更好地验证每个学习阶段和最终的连接阶段的贡献，我们还通过分别使用每个阶段的预测分数来进行评估。从下图 8© 中可以看出，相比于使用单一阶段进行预测，联合每个阶段的得分进行预测能够带来性能提升。此外，我们的方法将每个阶段的预测分数和联合特征的预测分数进行组合，能够实现最佳的识别性能。

（4）平衡参数 α 和 β：我们还研究了公式 $L=\alpha Lcon+\beta L_{KL}$ 中平衡参数对性能的影响。我们发现，当 $\alpha=0$，$\beta=1$ 时，即总损失仅包括 KL 散度时，此时模型无法收敛。当 $\alpha=1$，$\beta=0$ 时，即仅使用交叉熵损失进行优化，模型的性能明显下降。当 $\alpha=0.8$，$\beta=0.2$ 时，模型取得了最佳的识别性能，这说明联合渐进式训练和 KL 散度的策略能够提高多样化局部细节特征的学习能力。

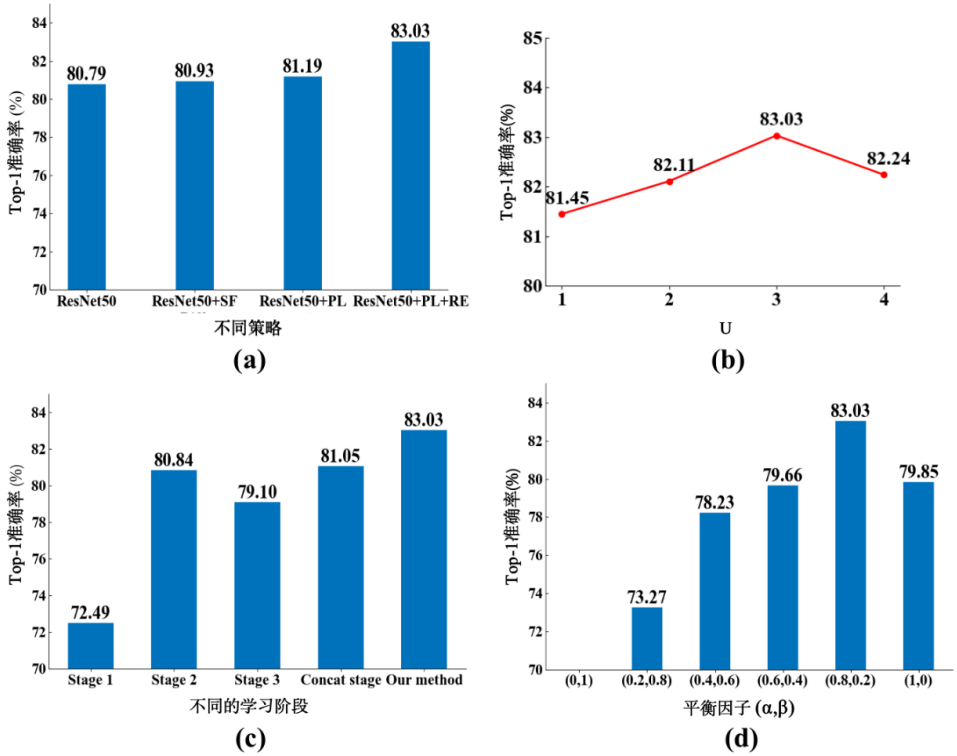


图 8　我们方法在 Food2K 上的消融实验

可视化

我们使用 Grad-CAM 来进行可视化分析。如下图 9 所示，以“Wasabi Octopus”为例，基线方法仅能获得有限的信息，不同的特征图倾向于关注相似的区域。相比之下，我们的方法在第一阶段更关注“Vegetable Leaf”，而第二阶段主要关注“Octopus”。而在第三阶段，我们的方法可以捕获该食品的总体特征，所以全局和局部特征都能被用于识别。

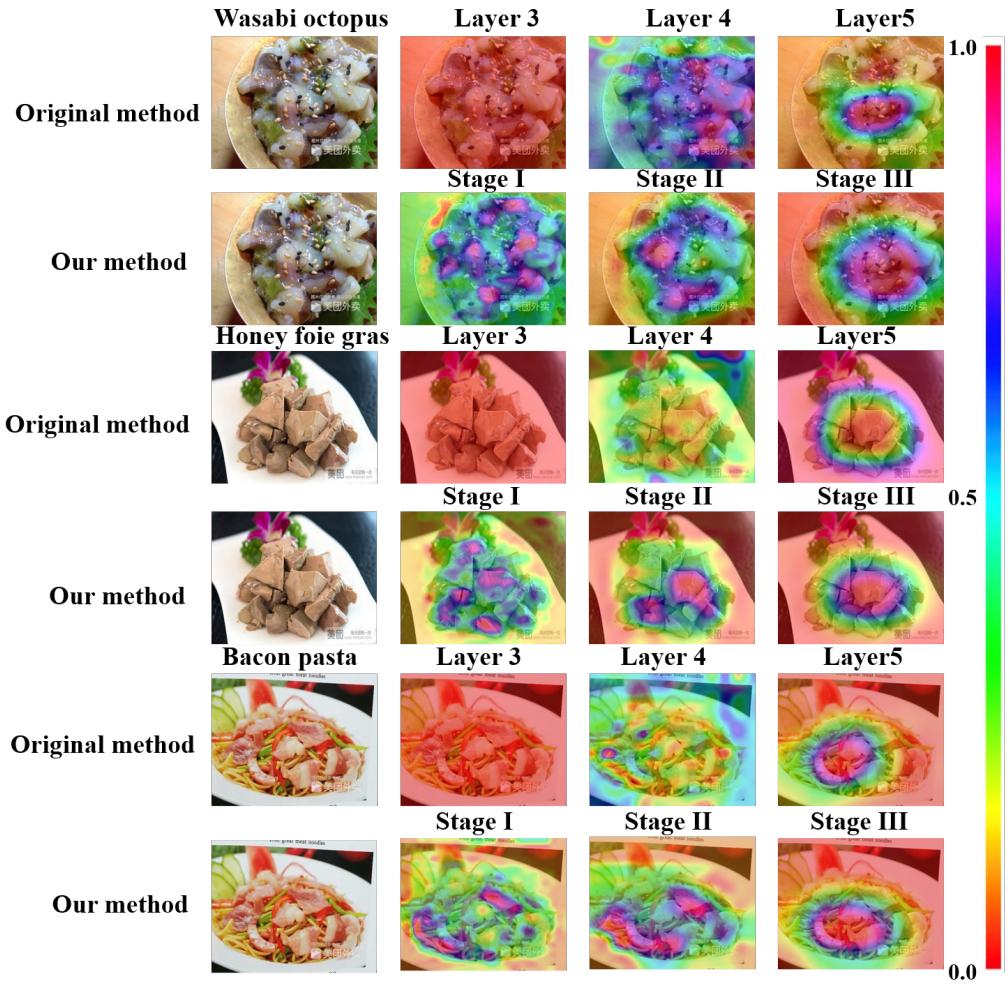


图 9　来自 Food2K 一些样本的可视化结果

基于 Food2k 的泛化实验

食品图像识别

我们评估了在 Food2K 上预训练过的模型在 ETH Food-101、Vireo Food-172 和 ISIA Food-500 上的泛化能力。从表 2 中我们可以看出，使用 Food2K 进行预训练后所有方法都取得了一定程度的性能提升，这说明我们的数据集在食品图像识别任务上具有良好的泛化能力。

Models	ETH Food-101		Vireo Food-172		ISIA Food-500	
	Top-1 Acc.	Top-5 Acc.	Top-1 Acc.	Top-5 Acc.	Top-1 Acc.	Top-5 Acc.
VGG16 [65]	79.02	93.78	80.41	94.59	55.22	82.77
+Fine-tuned on Food2K	**80.68**	**94.45**	**87.74**	**97.57**	**56.59**	**83.65**
ResNet50	84.50	96.18	87.32	97.01	56.75	82.80
+Fine-tuned on Food2K	**85.89**	**96.66**	**89.46**	**97.84**	**57.97**	**83.40**
ResNet152 [67]	86.61	96.95	86.86	97.11	57.03	83.80
+Fine-tuned on Food2K	**87.58**	**97.28**	**89.44**	**97.99**	**62.51**	**87.67**
Inception V3 [87]	84.15	96.11	87.58	97.39	56.03	83.42
+Fine-tuned on Food2K	**87.61**	**97.25**	**89.89**	**98.20**	**62.48**	**87.71**
DenseNet161 [61]	86.94	97.03	86.93	97.17	60.05	86.09
+Fine-tuned on Food2K	**88.22**	**97.57**	**89.54**	**97.84**	**62.46**	**87.75**
SENet154 [62]	88.62	97.57	88.71	97.74	63.83	88.61
+Fine-tuned on Food2K	**89.68**	**98.08**	**91.45**	**98.62**	**65.18**	**89.49**
NTS-NET (ResNet50) [69]	89.40	97.80	89.24	97.91	64.61	89.45
+Fine-tuned on Food2K	**90.78**	**98.17**	**90.33**	**98.37**	**65.34**	**89.53**
Inception v4 @ 448px [73]	90.00	-	90.23	97.86	65.49	88.35
+Fine-tuned on Food2K	**90.33**	**98.18**	**90.89**	**98.52**	**66.62**	**89.87**
PAR-Net (ResNet101) [9]	89.30	-	89.60	-	63.98	88.76
+Fine-tuned on Food2K	**89.93**	**97.72**	**90.92**	**98.74**	**64.72**	**89.23**
PMG(ResNet50) [64]	86.93	97.21	89.78	97.45	57.53	84.18
+Fine-tuned on Food2K	**87.45**	**97.67**	**90.32**	**97.51**	**63.21**	**87.94**
DCL(ResNet50) [71]	88.90	97.82	89.13	97.82	64.10	88.77
+Fine-tuned on Food2K	**89.57**	**97.85**	**90.80**	**98.41**	**64.37**	**88.90**

表 2　基于 Food2K 微调的模型在食品图像识别任务上的性能

食品检测

我们评估了 Food2K 数据集对食品检测任务的泛化能力，评估任务为检测食品托盘中的食品。为了进行比较，我们还对在 ETH Food-101 上进行预训练的模型进行了评估。从表 3 中可以看出，使用 Food-101 和 Food2K 能够提升所有方法的 mAP 和 AP75 指标，且 Food2K 所带来的性能增益要超过 Food-101。这说明我们的方法在食品检测任务上表现出良好的泛化性能。

Models	UNIMIB2016			Oktoberfest		
	mAP	AP50	AP75	mAP	AP50	AP75
SSD(VGG16) [93]	64.1	78.4	65.2	52.4	79.8	54.8
+Fine-tuned on ETH Food-101	64.3	76.5	73.2	54.8	80.0	62.1
+Fine-tuned on Food2K	**65.0**	**86.6**	**76.7**	**55.2**	**80.2**	**63.4**
RetinaNet(ResNet101) [94]	66.8	80.2	67.2	55.4	82.5	57.2
+Fine-tuned on ETH Food-101	67.3	76.5	70.9	55.7	78.2	58.9
+Fine-tuned on Food2K	**69.1**	**80.9**	**76.1**	**55.8**	**84.7**	**59.0**
Faster-RCNN(ResNet101) [95]	55.2	80.7	59.8	49.1	77.7	52.5
+Fine-tuned on ETH Food-101	55.5	80.5	60.3	51.8	83.0	52.9
+Fine-tuned on Food2K	**55.7**	**82.8**	**60.4**	**57.6**	**83.6**	**64.0**
PAN(ResNet101) [96]	58.3	88.6	66.3	50.4	77.8	56.5
+Fine-tuned on ETH Food-101	63.5	85.2	72.1	52.8	**81.8**	**58.3**
+Fine-tuned on Food2K	**64.1**	**87.4**	**74.6**	**52.9**	79.0	**58.3**
Cascade-RCNN(ResNet101) [97]	73.1	87.3	82.1	59.7	85.6	59.6
+Fine-tuned on ETH Food-101	73.2	86.1	82.7	61.2	85.9	70.1
+Fine-tuned on Food2K	**73.7**	**87.5**	**83.2**	**61.7**	**87.8**	**70.2**
Dynamic R-CNN(ResNet101) [98]	68.8	90.6	77.9	57.3	85.6	59.8
+Fine-tuned on ETH Food-101	69.2	88.3	79.8	58.1	86.1	65.4
+Fine-tuned on Food2K	**70.6**	**91.2**	**80.1**	**58.5**	**86.3**	**65.7**

表 3　基于 Food2K 和 ETH Food-101 微调的模型在食品检测任务上的性能比较

食品分割

我们还评估了 Food2K 在食品分割任务上的性能。从表 4 中可以看出，对于所有使用 Food2K 进行预训练的模型均能带来性能的提升。这也证明了我们的数据集在分割任务上具有良好的泛化表现。

Models	UECFoodPix Complete	
	mIoU	PixAcc(PA)
FCN8(ResNet50) [101]	55.8	70.2
+Fine-tuned on ETH Food-101	56.5	**71.4**
+Fine-tuned on Food2K	**56.6**	71.2
SegNet(ResNet50) [102]	57.6	71.7
+Fine-tuned on ETH Food-101	58.6	72.1
+Fine-tuned on Food2K	**60.3**	**73.0**
PSPnet(ResNet50) [103]	70.9	81.7
+Fine-tuned on ETH Food-101	72.2	82.0
+Fine-tuned on Food2K	**74.5**	**84.1**
DUC_HDC(ResNet101) [104]	70.9	81.1
+Fine-tuned on ETH Food-101	70.4	81.4
+Fine-tuned on Food2K	**72.7**	**82.5**
Deeplabv3+(ResNet101) [106]	71.2	82.4
+Fine-tuned on ETH Food-101	71.6	82.5
+Fine-tuned on Food2K	**73.4**	**83.3**

表 4　基于 Food2K 和 ETH Food-101 微调的模型在食品分割任务上的性能比较

食品图像检索

我们进一步在食品图像检索任务上验证 Food-2K 的泛化能力。具体来说，我们在 ETH Food-101、Vireo Food-172 和 ISIA Food-500 数据集上实验，并使用与前文相同的数据集划分方式。测试集的每张图片依次作为查询项，其余的图片作为检索库。我们分别使用交叉熵损失函数和以 Contrastive loss 和 Triplet loss 为代表的度量学习损失函数来微调 ResNet101 网络，并使用 mAP 和 Recall@1 指标评估方法的性能。

表 5 的结果展示了在 Food-2K 数据集上预训练后微调的网络取得了不同程度的性能增益。具体来说，在 Vireo Food-172 数据集上取得了最优性能，并在三个数据集上分别取得了 4.04%, 5.28% 和 4.16% 的性能增益。值得注意的是，当使用额外的 ETH Food-101 数据集预训练，以及在度量学习损失函数方法上微调的方法并没有取得性能增益，但使用 Food2K 数据集预训练仍然取得了性能增益，这是因为食品图像检索任务对目标数据集之间的差异较为敏感（ETH Food-101 和 Vireo Food-172），并间接表明来自 Food2K 的图像类别和尺度的多样性提升了食品图像检索任务的泛化性。

Models	ETH Food-101		Vireo Food-172		ISIA Food-500	
	mAP	Recall@1	mAP	Recall@1	mAP	Recall@1
Cross-entropy loss(ResNet101)	52.13	70.35	54.20	81.54	7.88	36.58
+Fine-tuned on ETH Food-101	-	-	56.36	80.08	10.66	38.91
+Fine-tuned on Food2K	**53.33**	**78.10**	**68.23**	**85.87**	**11.20**	**40.04**
Contrastive loss(ResNet101) [88]	71.31	80.02	82.33	86.62	16.77	46.98
+Fine-tuned on ETH Food-101	-	-	76.62	83.53	16.42	46.02
+Fine-tuned on Food2K	**77.91**	**83.30**	**83.61**	**87.07**	**21.52**	**49.64**
Triplet loss(ResNet101) [89]	73.85	80.79	83.14	86.74	22.66	48.90
+Fine-tuned on ETH Food-101	-	-	77.68	83.66	20.17	46.88
+Fine-tuned on Food2K	**78.17**	**83.05**	**83.65**	**87.31**	**27.08**	**51.03**

表 5　基于 Food2K 和 ETH Food-101 微调的模型在食品图像检索任务上的性能比较

跨模态菜谱 – 食品图像检索

我们还在跨模态菜谱 – 食品图像检索任务上进一步验证 Food2K 的泛化能力。具体来说，我们在 Recipe1M[11] 数据集上验证方法的性能，并使用与之相同的数据集划分方法。与此同时，我们使用 MedR 和 Recall@K 指标来评估。表 6 展示了我们使用不同的网络主干，并分别通过 ImageNet、ETH Food-101 和 Food2K 数据集预训练的结果。结果发现使用 ETH Food-101 和 Food2K 数据集预训练后在目标数据集上微调都取得了性能的增益，使用我们的 Food-2K 数据集取得了更大的性能增益。

Size	Methods	Image-to-Recipe				Recipe-to-Image			
		MedR	Recall@1	Recall@5	Recall@10	MedR	Recall@1	Recall@5	Recall@10
1k	JE [47]	5.2	24.0	51.0	65.0	5.1	25.0	52.0	65.0
	+Fine-tuned on ETH Food-101	4.8	24.3	52.1	**66.2**	4.6	24.7	53.6	66.6
	+Fine-tuned on Food2K	**4.3**	**25.8**	**53.3**	**66.2**	**4.1**	**26.4**	**54.5**	**66.9**
	VSE [91]	3.9	26.6	54.8	68.2	3.8	26.0	55.7	67.3
	+Fine-tuned on ETH Food-101	3.4	26.9	55.7	69.3	3.4	27.2	57.1	68.4
	+Fine-tuned on Food2K	**2.8**	**29.1**	**58.5**	**70.6**	**2.9**	**28.6**	**58.6**	**70.5**
	PWC++ [92]	3.3	25.8	54.5	67.1	3.5	24.8	55.0	67.1
	+Fine-tuned on ETH Food-101	2.8	26.3	55.1	68.2	2.9	26.1	55.6	68.2
	+Fine-tuned on Food2K	**2.6**	**27.4**	**56.3**	**69.3**	**2.7**	**27.1**	**56.7**	**68.8**
	AdaMine [92]	2.5	39.8	69.0	77.4	2.1	40.2	68.1	78.7
	+Fine-tuned on ETH Food-101	2.3	38.8	70.2	77.7	2.0	39.7	68.6	79.3
	+Fine-tuned on Food2K	**1.9**	**41.2**	**71.3**	**78.5**	**1.8**	**40.5**	**69.6**	**79.9**
	ACME [48]	1.0	51.8	80.2	87.5	1.0	52.8	80.2	87.6
	+Fine-tuned on ETH Food-101	1.0	51.9	80.8	87.3	1.0	53.3	79.8	87.6
	+Fine-tuned on Food2K	**1.0**	**52.3**	**81.3**	**88.9**	**1.0**	**53.7**	**80.9**	**88.3**
10k	JE [47]	41.9	-	-	-	39.2	-	-	-
	+Fine-tuned on ETH Food-101	39.6	-	-	-	38.1	-	-	-
	+Fine-tuned on Food2K	**37.5**	-	-	-	**35.2**	-	-	-
	VSE [91]	28.3	7.4	21.5	29.6	28.1	7.6	22.4	31.2
	+Fine-tuned on ETH Food-101	27.1	8.6	22.3	30.7	27.3	8.5	24.1	32.7
	+Fine-tuned on Food2K	**26.8**	**9.2**	**23.1**	**32.5**	**26.6**	**9.0**	**24.8**	**34.3**
	PWC++ [92]	27.5	7.4	20.3	29.9	27.6	7.7	21.3	30.3
	+Fine-tuned on ETH Food-101	26.4	7.9	21.8	30.8	26.8	8.3	22.4	31.4
	+Fine-tuned on Food2K	**25.8**	**8.5**	**22.4**	**33.8**	**25.2**	**8.9**	**24.0**	**33.2**
	AdaMine [92]	13.2	14.9	35.3	45.2	12.2	14.8	34.6	46.1
	+Fine-tuned on ETH Food-101	12.5	15.2	36.1	46.9	11.7	15.4	35.7	46.7
	+Fine-tuned on Food2K	**11.4**	**15.8**	**37.3**	**48.2**	**10.9**	**16.5**	**36.5**	**48.2**
	ACME [48]	6.7	22.9	46.8	57.9	6.0	24.4	47.9	59.0
	+Fine-tuned on ETH Food-101	6.5	23.2	47.1	58.3	5.4	23.9	48.0	59.8
	+Fine-tuned on Food2K	**6.2**	**23.5**	**47.7**	**59.4**	**5.2**	**25.6**	**48.3**	**60.7**

表 6　基于 Food2K 和 ETH Food-101 微调的模型在跨模态菜谱 – 食品图像检索任务上的性能比较

5. 未来工作

本文全面的实验证明了 Food2K 对于各种视觉和多模态任务具有较好的泛化能力。基于 Food2K 的一些潜在研究问题和方向如下。

（1）大规模鲁棒的食品图像识别：尽管一些细粒度识别方法（如 PMG[12，22]）在常规细粒度识别数据集中获得了最佳性能，但它们在 Food2K 上表现欠佳。虽然也有一些食品图像识别方法（如 PAR-Net[13]）在中小规模食品数据集上取得了较好的性能，但它们在 Food2K 大规模食品图像识别数据集上也并不能获得更好的性能。

我们推测，随着食品数据的多样性和规模的增加，不同食材、配饰和排列等因素产生了更复杂的视觉模式，以前的方法不再适用。因此，基于 Food2K 有更多的方法值得进一步探究。例如 Transformers[14，23] 在细粒度图像识别方面产生了巨大的影响，其在大规模数据集上的性能高于 CNNs。Food2K 可以提供足够的训练数据来开发基于 Transformers 的食品图像识别方法来提高性能。

（2）食品图像识别的人类视觉评价：与人类视觉对一般物体识别的研究相比，对食品图像识别进行评价可能更加困难。例如，食品具有较强的地域和文化特征，因此来自不同地区的人对食品图像识别会有不同的偏见。最近的一项研究 [15] 给出了人类视觉系统和

CNN 在食品图像识别任务中的比较。为了避免信息负担过重，需要学习的菜肴数量被限制在 16 种不同类型的食物中。更有趣的问题，值得进一步的探索。

（3）跨模态迁移学习的食品图像识别：我们已经验证了 Food2K 在各种视觉和多模态任务中的推广。未来我们可以从更多的方面来研究迁移学习。例如，食物有独特的地理和文化属性，可以进行跨菜系的迁移学习。这意味着我们可以使用来自东方菜系的训练模型对西方菜系进行性能分析，反之亦然。经过更细粒度的场景标注，如区域级甚至餐厅级标注，我们可以进行跨场景迁移学习来进行食品图像识别。此外，我们还可以研究跨超类别迁移学习的食品图像识别。例如，我们可以使用来自海鲜超类的训练模型来对肉类超类进行性能分析。这些有趣的问题也都值得深入探索。

（4）大规模小样本食品图像识别：最近，有一些基于中小型食品类别的小样本食品图像识别方法 [16,17] 研究。LS-FSFR[18] 是一项更现实的任务，它旨在识别数百种新的类别而不忘记以前的类别，且这些数百种新的食品类别的样本数很少。Food2K 提供了大规模的食品数据集测试基准来支持这项任务。

（5）更多基于 Food2K 的应用：本文验证了 Food2K 在食品图像识别、食品图像检索、跨模态菜谱 – 食品图像检索、食品检测和分割等各种任务中具有更好的泛化能力。Food2K 还可以支持更多新颖的应用。食品图像生成是一种新颖而有趣的应用，它可以通过生成对抗网络（GANs）[19] 合成与现实场景相似的新的食品图像。例如，Zhu 等人 [20] 可以从给定的食材和指令中生成高度真实和语义一致的图像。不同的 GANs，如轻量级的 GAN [21]，也可以用于生成基于 Food2K 的食物图像。

(6) 面向更多任务的 Food2K 扩展：基于训练的 Food2K 模型可以应用于更多与食物计算任务中。另外，考虑到一些工作 [6] 已经表明食材可以提高识别性能，我们计划扩展 Food2K 来提供更丰富的属性标注以支持不同语义级别的食品图像识别。我们还可以在 Food2K 上进行区域级和像素级标注使其应用范围更广。我们还可以开展一些新的任务，如通过在 Food2K 上标注美学属性信息，对食品图像进行美学评估。

6. 总结及展望

在本文中，我们提出了具有更多数据量、更大类别覆盖率和更高多样性的 Food2K，它可以作为一个新的大规模食品图像识别基准。Food2K 适用于各种视觉和多模态任务，

包括食品图像识别、食品图像检索、检测、分割和跨模态菜谱 – 食品图像检索。

在此基础上，我们进一步提出了一个面向食品图像识别的深度渐进式区域增强网络。该网络主要由渐进式局部特征学习模块和区域特征增强模块组成。渐进式局部特征学习模块通过改进的渐进式训练方法学习多样互补的局部细粒度判别性特征，区域特征增强模块利用自注意力机制将多尺度的丰富上下文信息融入到局部特征中以进一步增强特征表示。在 Food2K 上进行的大量实验证明了该方法的有效性。

美团本身有着丰富的食品数据及业务应用场景，如何利用多元化数据进行食品图像细粒度分析理解，解决业务痛点问题是我们持续关注的方向。目前，美团视觉智能部持续深耕于食品细粒度识别技术，并成功将相关技术应用于按搜出图、点评智能推荐、扫一扫发现美食等不同的业务场景中，不仅提升了用户体验，还降低了运营成本。

在技术沉淀层面，我们围绕此食品计算技术不断推陈出新，目前申请专利 20 项，发表 CCF-A 类会议或期刊论文 4 篇（如 AAAI、TIP、ACM MM 等）；我们还参加了 2019 年和 2022 年 CVPR FGVC 细粒度识别比赛，并取得了一冠一亚的成绩；同时在 ICCV 2021 上也成功举办了以 LargeFineFoodAI 为主题的视觉研讨会，为推动食品计算领域的发展贡献了一份绵薄之力。

未来，我们计划进一步围绕这条主线，探索多模态信息融入、多任务学习等技术路线，不断沉淀经验教训，并将相关技术推广到更多、更远、更有价值的生活服务场景中，从而更好地服务好社会。

7. 参考文献

[1] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, “A survey on food computing,” ACM CSUR, vol. 52, no. 5, pp. 1 – 36, 2019.

[2] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy, “Im2Calories: towards an automated mobile vision food diary,” in ICCV, 2015, pp. 1233 – 1241.

[3] Q. Thames, A. Karpur, W. Norris, F. Xia, L. Panait, T. Weyand, and J. Sim, “Nutrition5k: Towards automatic nutritional understanding of generic food,” in CVPR, 2021, pp. 8903 – 8911.

[4] Y. Lu, T. Stathopoulou, M. F. Vasiloglou, S. Christodoulidis, Z. Stanga, and S. Mougiakakou, “An artificial intelligence-based system to assess nutrient intake for hospitalised patients,” IEEE TMM, pp. 1 – 1, 2020.

[5] L. Bossard, M. Guillaumin, and L. Van Gool, “Food-101 – mining discriminative components with random forests,” in ECCV, 2014, pp. 446 – 461.

[6] J. Chen and C.-W. Ngo, “Deep-based ingredient recognition for cooking recipe retrieval,” in ACM MM, 2016, pp. 32 – 41.
[7] W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, and X. Wei, “ISIA Food- 500: A dataset for large-scale food recognition via stacked globallocal attention network,” in ACM MM, 2020, pp. 393 – 401.
[8] J. Mar´ın, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, “Recipe1M+: A dataset for learning cross-modal embeddings for cooking recipes and food images,” IEEE T-PAMI, vol. 43, no. 1, pp. 187 – 203, 2021.
[9] H. Wang, G. Lin, S. C. H. Hoi, and C. Miao, “Structure-aware generation network for recipe generation from images,” in ECCV, vol. 12372, 2020, pp. 359 – 374.
[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in CVPR, 2016, pp. 770 – 778.
[11] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, “Learning cross-modal embeddings for cooking recipes and food images,” in CVPR, 2017, pp. 3020 – 3028.
[12] R. Du, D. Chang, A. K. Bhunia, J. Xie, Z. Ma, Y. Song, and J. Guo, “Fine-grained visual classification via progressive multigranularity training of jigsaw patches,” in ECCV, 2020, pp. 153 – 168.
[13] J. Qiu, F. P.-W. Lo, Y. Sun, S. Wang, and B. Lo, “Mining discriminative food regions for accurate food recognition,” in BMVC, 2019.
[14] Dosovitskiy, Alexey, et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. “ In ICLR, 2020.
[15] P. Furtado, M. Caldeira, and P. Martins, “Human visual system vs convolution neural networks in food recognition task: An empirical comparison,” in CVIU, 2020, pp. 102878.
[16] H. Zhao, K.-H. Yap, and A. Chichung Kot, “Fusion learning using semantics and graph convolutional network for visual food recognition,” in WACV, 2021, pp. 1711 – 1720.
[17] S. Jiang, W. Min, Y. Lyu, and L. Liu, “Few-shot food recognition via multi-view representation learning,” ACM TOMM, vol. 16, no. 3, pp. 87:1 – 87:20, 2020.
[18] A. Li, T. Luo, Z. Lu, T. Xiang, and L. Wang, “Large-scale few-shot learning: Knowledge transfer with class hierarchy,” in CVPR, 2019, pp. 7212 – 7220.
[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in NIPS, vol. 27, 2014.
[20] B. Zhu and C. Ngo, “CookGAN: Causality based text-to-image synthesis,” in CVPR, 2020, pp. 5518 – 5526.
[21] B. Liu, Y. Zhu, K. Song, and A. Elgammal, “Towards faster and stabilized gan training for high-fidelity few-shot image synthesis,” in ICLR, 2020.
[22] Zhu, H., Ke, W., Li, D., Liu, J., Tian, L., & Shan, Y. Dual cross-attention learning for fine-grained visual categorization and object re-identification. In CVPR 2-22(pp. 4692-4702).
[23] He, J., Chen, J. N., Liu, S., Kortylewski, A., Yang, C., Bai, Y., & Wang, C. . Transfg: A transformer architecture for fine-grained recognition. In AAAI 2022 (Vol. 36, No. 1, pp. 852-860).

8. 本文作者

致岭、丽萍、君实、晓明等，均来自美团基础研发平台 / 视觉智能部。

斩获 CVPR 2023 竞赛 2 项冠军｜美团街景理解中视觉分割技术的探索与应用

作者：金明 旺旺等

1. 问题背景

街景数据通过不同设备进行采集获取，以各种摄像头采集的视频图像及各种雷达采集的点云为主要载体。其中，摄像头采集的视频图像具有低成本、易获取的特点，是最为常见的街景数据类型，而本文处理的街景数据主要为视频图像数据。街景视频图像数据作为室内外场景的重要信息载体，是计算机视觉众多任务的关键研究对象。

为了从视频图像数据中解析有效的街景信息，计算机视觉各项技术融汇互补，实现对交通道路、室内外空间等街景进行深度全面的理解，这个过程通常被称为街景理解。街景理解相关技术的发展在计算机视觉技术演进中扮演着非常重要的角色，同时也对众多下游任务（例如场景重建、自动驾驶、机器人导航等）发挥着重要的作用。

总的来说，街景理解技术融汇了众多计算机视觉技术，从不同技术的识别结果的表示形式上，可以划分为四个层级：点级、线级、面级、体级，以及每个层级内、不同层级间要素的逻辑关系。其中：

1. 点级提取技术用于解析各种与“点”相关的信息，以提取坐标及特征描述子为主，包括通用特征点、语义关键点等各种点级信息的提取技术，处理对象包括各种要素，用于表征要素的位置、特征等信息。
2. 线级提取技术用于解析各种与“线”相关的信息，以提取线条为主，包括车道线、地平线、各类曲线 / 直线等各种线级信息的提取技术，处理对象包括各种线条，用于表征要素的位置、矢量、拓扑等信息。
3. 面级提取技术用于解析各种与“面”相关的信息，以提取区域为主。街景视频图像数据由于透视投影的成像方式，所有信息均展示在二维平面上，该平面根据不同语义、不同实例被划分为不同区域，这些区域表征了要素的二维位置、轮廓、语义等信息。本层次能力包括语义分割、实例分割等提取技术。
4. 体级技术用于解析各种与“体”相关的信息，以提取三维结构为主，包括深度估

计、视觉显式 / 隐式三维重建等提取技术，用于表征场景及要素的三维结构信息。

5. 逻辑关系提取技术基于以上技术的提取的要素及场景信息，通过时序信息融合及逻辑推理，提取不同层级或同一层级要素间的逻辑关系，包括点的匹配关系、线的拓扑关系、要素的多帧跟踪及位置关系等。

具体到现实场景，点级、线级、面级提取技术的的识别结果，如下图 1 所示：

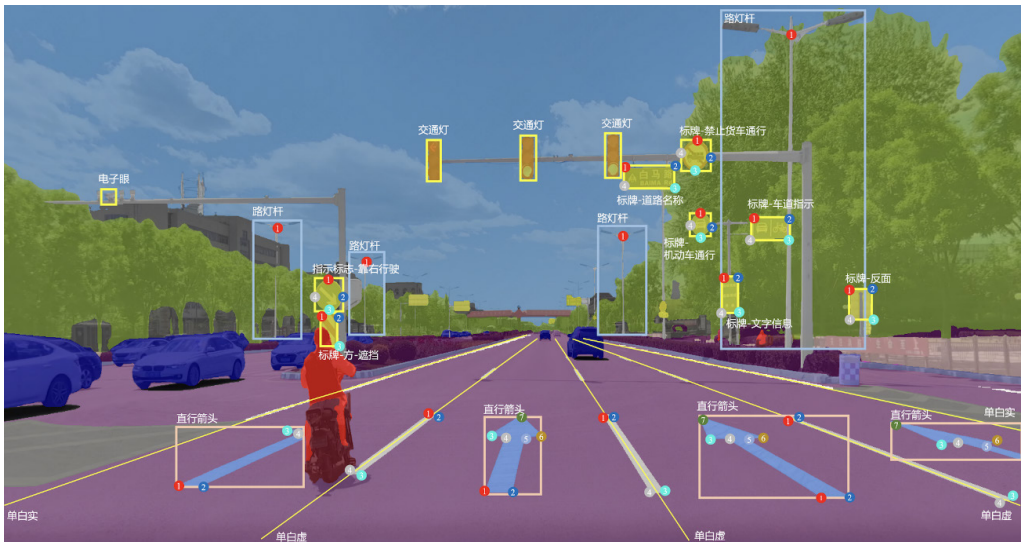


图 1　街景解析结果示例

在街景理解中，各类视频图像分割技术是“面级”提取和“逻辑关系”提取中的关键技术，实现对街景二维信息的像素级表征。在街景分割中，由于实际场景的复杂性，面临众多难题。

首先，街景分割的突出难点是要素的形状、尺寸差异大，如图 2 第一列所示（图像示例来自于数据集 [1]）。由于现实场景中各种目标的多样性以及视频图像成像的局限性，采集数据中目标存在各种异型或不完整问题。此外，由于透视成像的问题，远处与近处的相同目标在图像中大小差异极大。这两个问题要求街景分割算法具备鲁棒的复杂目标精准分割能力。

其次，街景分割的另一难点是恶劣自然条件带来的干扰，如图 2 第二、三列所示（示例来自于数据集 [2]）。由于实际场景中恶劣天气或极端光照条件是经常出现的，采集数据中目标往往受到自然条件的影响，存在可见度低、遮挡或模糊等问题。这要求街景分割

算法具备困难目标的发现与识别能力。

此外，由于街景理解中需要利用视频 / 图像等不同数据形式不同结果表征的分割技术，如何构建高效率迭代的分割技术？如何保证不同分割算法间相互配合、性能互补，同时保证多种算法在有限的计算资源与维护成本下共存？如何将分割任务与其他任务结合，变得更加统一开放？也是街景分割亟需解决的难题。



图 2　街景图像示例

为了解决以上难题，美团街景理解团队在分割技术上做了大量探索，构建了一系列真实复杂场景下的高精度分割算法，实现了复杂目标精准分割及困难目标的发现识别。同时，也对高效率分割技术进行了实践，实现了分割模型的高效迭代与应用，并探索了统一开放的的街景视觉分割技术。最终，提出的相关技术在街景理解中取得了明显的效果，相关算法被 CVPR 2023 接收为 Workshop 论文，并且在国际竞赛中取得了 2 项冠军 1 项季军的成绩。

2. 研究现状

2.1 分割技术体系

分割作为计算机视觉三大基础任务（分类、检测、分割）之一，对目标进行像素粒度的位置和轮廓信息表示。计算机视觉进入深度学习时代之后，分割任务根据不同的应用场景进一步细分发展出各种子任务，按照数据形式的不同分为两大类：图像分割和视频分割，如下图 3 所示（图片来自 [3][15] 等文献）：

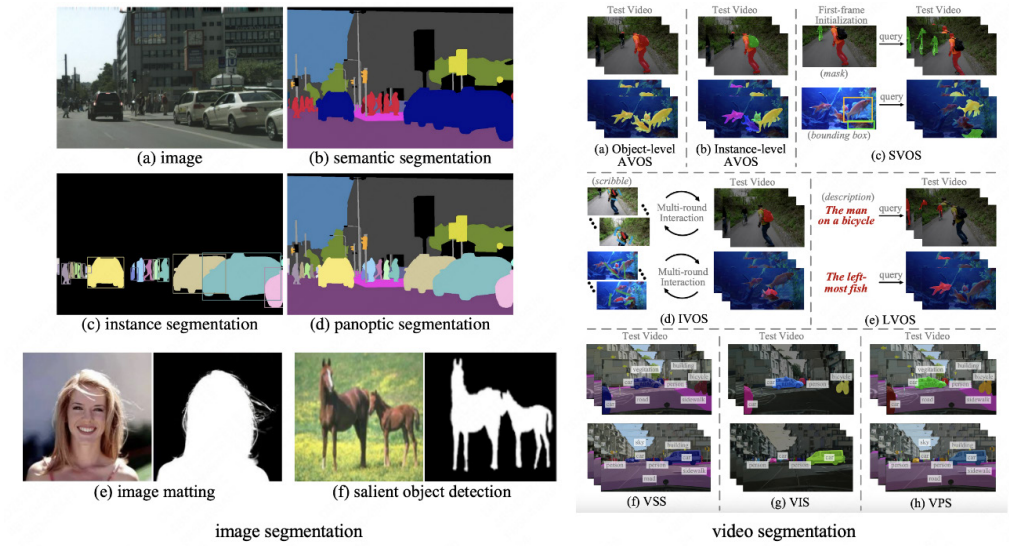


图 3　不同分割任务示例

图像分割任务的处理对象是单张图像，根据输出结果表示形式的不同，逐渐发展出语义分割、实例分割、全景分割等。其中，语义分割将图像中每个像素分配到对应的语义类别，代表工作有 FCN[4]、U-Net[4]、DeepLab[6]、OCRNet[7]、SegFormer[8]等。实例分割的目标是将图像中每个实例进行分割，并识别出每个实例的语义类别，代表工作有 Mask-RCNN[9]、YOLACT[10]、SOLO[11] 等。全景分割的目标是将图像中所有像素分配到对应的语义类别中，并区分不同实例，代表工作有 EfficientPS[12]、Panoptic-DeepLab[13]、UPSNet[14] 等。除了以上分割任务以外，还有一些其他图像分割技术在许多场景有应用需求，如抠图、显著性目标分割等图像分割任务。

视频分割任务的处理对象具有时序关系的视频序列，其不仅关注单帧的空间维度信息，同时关注多帧间的时序维度信息。按照输入和输出结果的表示形式的不同，视频分割分为视频目标分割、视频语义分割、视频实例分割、视频全景分割等；视频目标分割根据输入形式的不同，进一步分为自动视频目标分割、半自动视频目标分割、交互式视频目标分割、语言引导的视频目标分割等子任务。自动视频目标分割仅利用视频帧作为输入，分割出视频序列目标所在区域，代表工作有 OSVOS[16]、MATNet[17] 等。半自动视频目标分割除了输入视频序列以外还输入指定所要分割的目标，代表工作有 SiamMask[18]、STM[19]、AOT[20] 等。交互式视频目标分割是指在用户的交互下，对视频前景目标进行分割，代表工作有 MiVOS[21] 等。语言引导的视频目标分割通过输入文

本来指定所要分割的目标，代表工作有 CMSANet[22] 等。

除了按照以上方式对分割任务进行分类以外，也可以根据模型学习范式或者监督程度来进行分类，根据训练数据标注信息受限程度的不同，分为强监督分割、无监督分割、半监督分割、弱监督分割等。当分割模型实际运行时，许多应用场景下设备的计算资源往往是受限的，例如一些移动端设备提供的计算资源十分有限，然而实际需求又要求模型具备一定的实时性，这就要求分割模型在模型架构设计上具备高效性。一些工作聚焦于此进行研究，如 BiSeNet[23]、STDCNet[24] 等。此外，真实世界中的语义类别复杂多样，各个类别的数据分布也不均匀，因此也衍生出了开集分割、长尾分割等研究方向。

2.2 街景分割现状

针对街景分割中存在的问题，许多方法提出了相应解决方案。

为了解决复杂目标的精准分割问题，PSPNet[25] 提出金字塔池化模型，利用不同尺度上下文信息实现不同尺度物体的分割。OCRNet[7] 引入物体上下文特征表示模块用于上下文信息增强的特征，预测每个像素的语义类别。SegFormer[8] 提出一种新的层次结构 Transformer 编码器输出多尺度特征，同时引入多层感知机解码器聚合来自不同层的信息，得到不同目标的分割结果。此外，也有一些基于视频的方法，例如 TMANet[26] 通过利用相邻帧的特征对当前帧进行增强，解决目标精准分割问题。目前已有的方法中，基于图像的方法主要从上下文特征增强的角度出发，但由于单帧单视角的信息对于场景的信息刻画不够完整，因此对于复杂目标的精准分割是困难的。同时，基于视频的方法主要利用周围帧的特征增强当前帧的特征，但由于多帧特征的对齐与融合存在困难，因此复杂目标的精准分割仍难以解决。

为了解决困难目标的发现与识别问题，难样本挖掘技术通用被用于增强特征的辨别能力。为了减少识别难样本的计算开销，已有的工作主要在两个方向进行了探索：每个 batch 内的精确搜索、整个数据集的近似搜索。例如，OHEM[27] 基于损失函数的反馈在 batch 中自动选择困难样本，以使训练更有效和更高效，减少了繁重的启发式搜索做法和超参数。此外，UHEM[28] 通过分析视频序列上经过训练的检测器的输出来自动获得大量困难的负样本。此外，SCHEM[29] 使用类别签名在训练期间以较小的额外计算成本在线跟踪特征嵌入，通过使用该签名识别困难负样本。目前已有的方法中，通过优化训练策略的方式增强难样本的学习能力，从而实现困难目标的发现与识别，但由于

这些方案都是通过约束模型训练从而间接达到目标发现的目的，因此并不能完全解决困难目标的发现与识别问题。

为了解决分割模型的高效率迭代问题，研究人员也做出了许多努力。为了提高分割任务的标注效率，ScribbleSup[30]、FocalClick[31]、MiVOS[21] 等通过交互式分割的方式加快对图像或视频目标的像素级标注。此外，由于长尾分布的存在，CANet[32]、PADing[33] 等通过少样本学习和零样本学习的方法减少对稀少类别数据样本的依赖，还有一些工作通过重采样、类别平衡损失函数等方法在训练过程中缓解长尾问题。另外，在模型结构的设计上需要注重效率，例如 BiSeNet[23]、STDCNet[24] 等模型通过多分支网络结构获取更好的实时性，ShuffleNet[34]、MobileNet[35] 等通过特殊算子和模块减少模型的计算量和参数量。

随着分割技术的发展，衍生出了许多不同的分割子任务，如语义分割、实例分割、全景分割等，其在数据标注形式、输出表征方式、模型结构设计等方面都存在着一定的差异。在不同的分割算法间找到统一的解决方案，充分利用不同形式的分割标注数据等都是重要的研究问题。MaskFormer[36]、OneFormer[37] 等方法提出了通用的分割模型结构，其将语义分割、实例分割、全景分割等任务统一起来，并且能较为容易地由图像分割推广到视频分割。近期提出的 Segment Anything Model[38] 则作为一个零样本基础分割模型，能够“分割一切”，基于 SAM 可以进一步开发出许多的分割下游应用，如语义分割、实例分割等、视频目标分割等。

3. 核心技术

本部分针对街景理解分割任务存在的问题，从真实复杂场景下的高精度分割、分割模型的高效率迭代以及通向统一开放的街景分割等三个方面介绍相应的解决方案。

3.1 复杂场景下的高精度分割

3.1.1 基于时空对齐的复杂目标精准分割

复杂目标的精准分割问题，表现为复杂目标中经常存在部分区域的预测结果具有高不确定性，导致该区域难以精准分割甚至分割错误。通常，通过融合周围帧的预测信息，可以提高当前帧目标分割的确定性，从而提高该目标的分割精度。为解决复杂目

标的精准分割问题，街景理解团队提出了一种基于时空对齐的复杂目标精准分割框架（Motion-State Alignment Framework，简称 MSAF），如下图 4 所示：

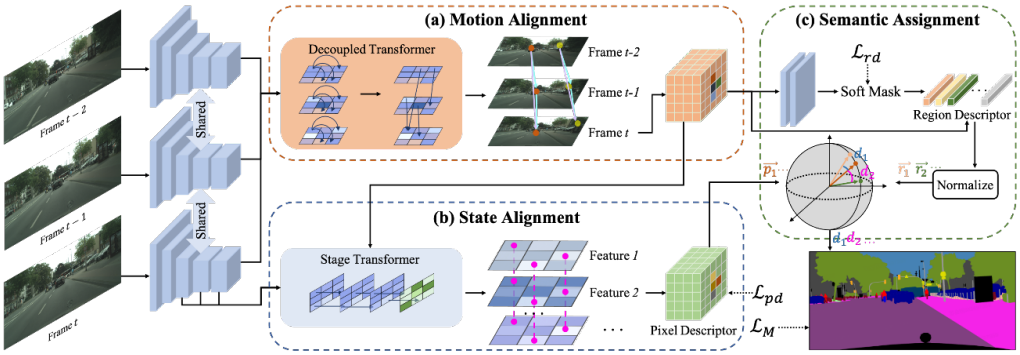


图 4　基于时空对齐的复杂目标精准分割框架

MSAF 重新思考了视频信号带来的信息增益：视频中包含的语义信息可分为动态语义和静态语义，动态语义存在于连续多帧的时序关系中，能够增益目标区域的位置信息和特征描述；静态语义存在于确定时刻的单帧图像中，能够有效恢复目标区域的细节信息。动静态语义能够为分割模型带来不同方面的确定性信息增益。

MSAF 首先提取视频相邻多帧的特征，通过动态特征对齐机制及静态特征对齐机制，分别对当前帧的动静态语义特征进行增强，接着从动态语义特征中提取目标区域描述子、从静态语义中提取目标像素描述子，然后求解像素描述子与区域描述子的特征距离，为每个像素指定准确的区域类别，实现复杂目标的精准分割。

最终，与主流的图像、视频级分割方法相比，基于时空对齐的复杂目标精准分割方法在 Cityscapes[1]、CamVid[39] 等数据集上均达到领先的精度，并具有较快的推理速度。

3.1.2 基于样本自动挖掘的困难目标发现

为了解决困难目标发现与识别的问题，街景理解团队提出了一种基于样本感知、挖掘、净化的困难目标发现框架（Perceive, Excavate and Purify，简称 PEP），如下图 5 所示：

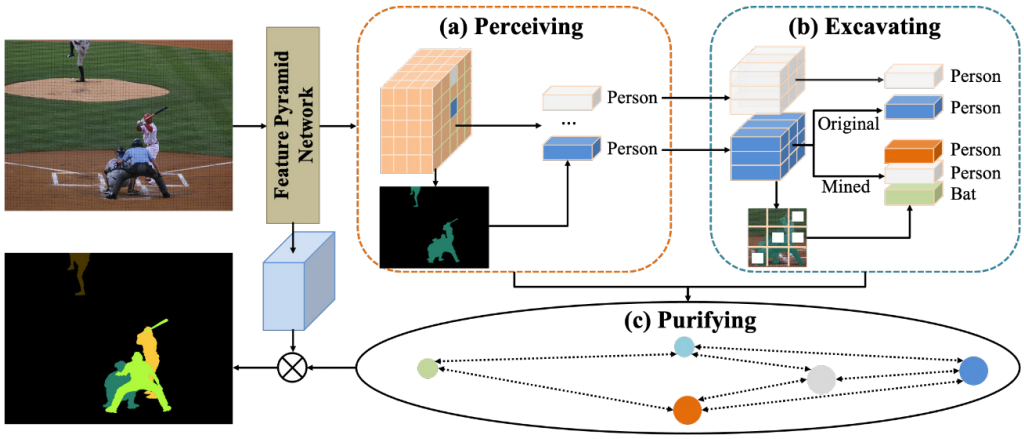


图 5　基于样本自动挖掘的困难目标发现框架

首先，利用特征金字塔主干网络提取不同尺度的特征，然后将不同尺度特征分别输入到三个分支：实例感知分支、实例描述分支、特征学习分支。

实例感知分支对主干特征每个像素点进行分类，初步确定像素点位置是否存在实例；实例描述分支学习不同实例的原始特征描述子，同时利用样本挖掘子网络来挖掘困难目标，并表征为提取描述子。此外，引入实例关联子网络来提高相同实例的相似性、降低不同实例的相似性，实现目标净化，进一步提高分割性能。最后，原始的和挖掘的实例描述子与特征学习分支的通用特征实施卷积操作得到各个目标的分割结果。

最终，与主流的分割方法相比，基于样本自动挖掘的困难目标发现方法在 COCO[40] 等数据集上达到领先的精度。

3.2 分割模型的高效率迭代

为了更好地适应街景理解中变化多样的真实场景，满足各种不断新增的实际业务需求，街景分割模型需要不断进行迭代。因此，建立一种高效率的迭代方式是十分必要的。街景理解团队经过长期探索，构建了一套面向分割任务的高效数据 – 模型闭环，能够以有限的成本积累大量的高质量有标签分割数据集，不断提升分割模型的性能，还能够高效地完成模型迭代，以满足实际业务场景的定制化需求。数据 – 模型闭环的整体流程图如下图 6 所示：

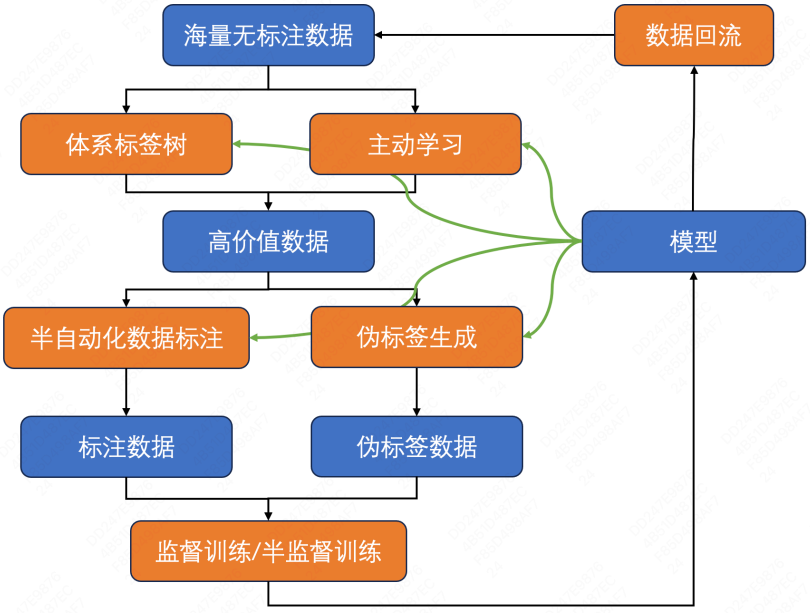


图 6　数据 – 模型闭环流程

实际的街景理解业务场景中通过数据回流可以获取海量的无标注数据，这些无标注数据经过众多街景理解模型的推理预测可以得到丰富多样的标签属性，由此可以构建能够覆盖各种复杂场景、层次结构丰富的街景理解体系标签树。在新增的业务需求到来时，依靠体系标签树可以及时高效地获取大量与需求相关性较高的数据。此外，模型在数量众多的无标注数据上通过主动学习也可以筛选挖掘出不确定性更高、对模型迭代更加有价值的数据。

面对高价值的数据，通过模型与标注人员高效协同作业的半自动化数据标注，可以获得标注质量更有保证的数据，也可以通过伪标签技术，基于已有的模型获取大量带有伪标签的数据，然后通过有监督或半监督训练完成模型迭代。迭代之后，性能更加优异、能力更加丰富的新模型不仅能够在业务场景中更好地赋能，还能够更好地助力数据 – 模型闭环中的各个环节。由此，数据 – 模型闭环实现不断迭代、循环。

在分割模型的实际部署应用时，需要平衡好业务侧的识别精度和模型的计算资源。为此，构建了包括轻量级、中等量级、重量级的分割模型族。轻量级的分割模型凭借其较小的参数量和极高的吞吐量，常用于计算资源有限的端侧部署或者调用量极大的服务端部署等场景；中等量级的分割模型则用于对精度要求更高，调用量规模中等的场景；重

量级的分割模型则凭借其极高的模型容量和识别精度，通过模型蒸馏、伪标签生成、预标注生成等手段助力更加轻量的模型的性能提升和数据 – 模型闭环，使其优势能够发挥到实际业务场景的前线。

此外，通过模型量化、高性能部署等手段，可以进一步提高模型的执行效率，降低计算成本，实现分割模型的高效率应用。

3.3 通向统一开放的街景分割

近期，随着 ChatGPT[41]、Stable Diffusion[42] 等自然语言处理、多模态大模型的火爆，人们对大模型、统一模型的关注度持续提升，统一的视觉大模型的热度也不断上升，Segment Anything Model[38]、UniAD[43] 等的出现也显示出了通用统一的基础模型在视觉领域的潜力。街景理解团队在分割统一大模型上也在不断探索。

在图像分割上，探索涵盖语义分割、实例分割、全景分割、边缘检测等多任务的统一分割模型结构，并通过多任务训练充分发挥多种类型分割标注数据的潜力，保证不同任务间相互配合，取得增益。在视频分割上，也在探索涵盖视频语义分割、视频实例分割、视频全景分割、视频目标分割等任务的统一分割模型结构，在视频分割标注困难的情况下充分利用已有的各种视频分割标注及图像标注数据。此外，将图像和视频的各自分割任务中学习到的知识迁移到另一个任务中也是十分重要的研究内容。

另外，分割任务与其他视觉任务的相互融合、相互配合也是一个十分重要的方向，其在街景理解技术体系中有着重要的作用。如分割任务与分类、检测任务的融合，不仅能够在计算资源有限的场景下降低资源的占用，提高系统整体的吞吐，还能充分发挥不同视觉任务监督信息的潜力。

除了视觉任务内各个任务之间的融合和统一，与分割任务相关的跨模态研究也有着重大潜力，如与文本模态结合的开集分割任务、文本引导的指向性分割任务等，其不仅能够将分割任务推广到更加开放的真实环境中，还能够通过文本这个桥梁提高人与分割模型之间的交互能力，使其能够更加快速、精准地实现定制化的分割需求。基于分割任务的更高层次的语义推理研究也有着重要的价值，在分割技术这样细粒度的基础场景理解、语义解析能力的加持下，辅以知识先验、逻辑推理能力强大的大语言模型，在未来也能够在街景理解场景中产生巨大的应用价值。

总之，通向更加统一、更加开放的街景视觉分割模型已经成为了现在以及未来的重要方向，街景理解团队将会继续实践、沉淀，探索视觉分割模型的未来。

4. CVPR 2023 技术成果

基于街景理解中的分割技术积累，街景理解团队在 CVPR 2023 发表了 2 篇 Workshop 论文，并在相关竞赛中取得了 2 项冠军 1 项季军的成绩，同时相应技术成果已经申请多项国家专利。

4.1 论文发表

面向分割任务的基于时空对齐的复杂目标精准分割方法、基于样本自动挖掘的困难目标发现方法完善为 2 篇学术论文：《Motion-state Alignment for Video Semantic Segmentation》[44]、《Perceive, Excavate and Purify: A Novel Object Mining Framework for Instance Segmentation》[45]，被 CVPR 2023 接收为 Workshop 论文（8 页长文）。

4.2 恶劣天气街景理解竞赛双赛道冠军

在自动驾驶场景中，恶劣天气（如雾、雨、雪、弱光、夜间、曝光、阴影等）会给感知系统带来很大的干扰。为了保证自动驾驶汽车在恶劣天气的情况下平稳运行，感知系统需要具备处理极端天气的能力。虽然计算机视觉技术在街景理解等场景中的表现正在快速发展，但是已有的评测基准主要集中在晴朗的天气条件下（好天气、有利的照明），即使是当前性能最好的算法，在恶劣天气条件下也会出现严重的性能下降。为此，ACDC Challenge 提出了专门针对恶劣天气的评测基准，用于促进对不利天气和光照条件下鲁棒视觉算法设计的研究。

本次竞赛中，街景理解团队获得了语义分割、全景分割两个赛道的冠军。

4.3 视频理解竞赛视频全景分割赛道季军

像素级场景理解是计算机视觉中的基本问题之一，其目的是识别给定图像中每个目标的语义类别和准确位置。由于现实世界实际上是动态而非静态的，因此面向视频的全景分割对于现实应用具有合理性和实用性。因此，PVUW Challenge 提出了自然场景下具有实用且具有挑战的大规模视频全景分割数据集并举办竞赛，用于促进视频全景分割的

算法研究。

本次竞赛中，街景理解团队获得了视频全景分割赛道的季军。

5. 业务应用

美团街景理解中的分割技术在多个业务场景得到了广泛应用，其中主要包括地图、自动配送、门店实景化等业务。

地图是美团本地生活服务必不可少的基础设施，地图数据自动化生产是地图业务的重要环节，该环节主要从图像等数据中进行各类交通要素的提取与处理，分割技术在其中发挥着重要作用。首先，分割技术作用于低质图像过滤，提取道路、车辆、镜头遮挡物等类别的准确区域，识别并防止了道路拥堵、镜头遮挡等低质图像影响地图数据生产流程。同时，分割技术作用于交通要素提取，有效提取各类车道线（例如单白实线、双黄虚线、四线等）、各类物体隔离物（例如栅栏、水马、混凝土石墩等）的位置信息、轮廓信息、语义信息、实例信息等，用于后续要素生产。

此外，分割技术作用于道路结构提取，解析主辅路及路口结构等，获取道路结构，同时用于交通要素的位置信息提取。另外，分割技术也作用于卫星影像识别，如自动化提取建筑楼栋用于地图前端的渲染等。如下图 7 所示：

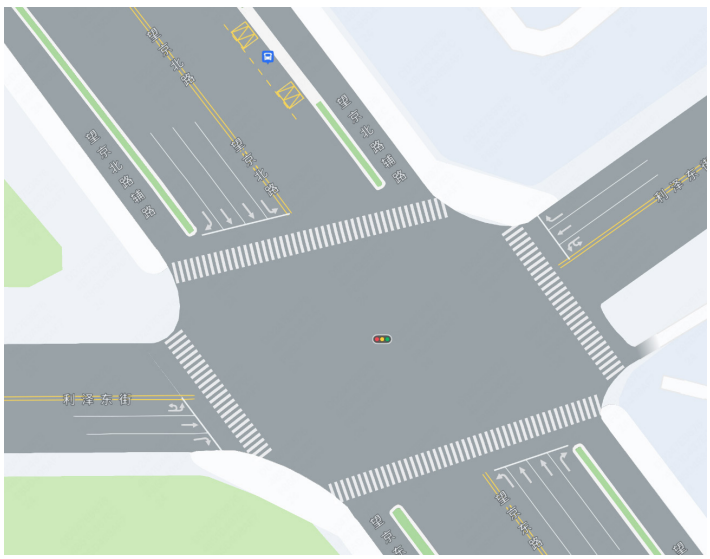


图 7　美团地图北京望京路段前端渲染效果

自动配送业务围绕美团外卖、跑腿等核心业务，提升配送效率和用户体验。高精地图是自动驾驶的核心基建，为了保证低成本、高鲜度，高精地图的生产过程中视频图像数据的分割发挥着重要作用。其中，各类车道线、地面箭头、交通标牌等各类要素分割为交通要素的提取提供了重要信息，杆体、公交站台等交通设施的分割为交通设施提取提供了重要信息，提高了高精地图生产中要素提取的自动化率和精度。

此外，高效率的语义分割模型为高精地图的定位图层提供了 70+ 类的语义分割，支持了定位图层的生产，高精度语义分割支持了感知模型的半监督学习。另外，视频图像分割技术为视觉高精地图建图提供了各类要素的提取及跟踪能力，提高了建图的成功率、自动化率和精度。

门店实景化业务涉及广泛的线下室内场景，目的是提供基于以视觉为主的室内建图与渲染解决方案，提供对线下场景的几何、语义、物体等内容的理解以及渲染能力，分割技术在其中发挥着重要作用。在门店实景化业务中，分割技术提高了重建精度和渲染成功率，有效支持了门店布局估计、营业面积计算、关键设施自动化计数等。

此外，分割技术在智能标注、数据生成等应用中也发挥着重要作用，同时也在为街景理解中其他技术进行赋能。

6. 总结与展望

分割技术在街景理解中占有重要地位，同时也面临众多挑战。为了应对这些挑战，美团街景理解团队构建了一套兼顾精度与效率的分割技术体系，在业务应用中取得了显著效果。

随着人工智能技术的发展，街景理解中的分割技术也将更加精确、更加通用、更加智能。结合多源数据与先进模型的自动化迭代，分割效果将会越来越准确；结合语言及视觉大模型等先进技术，分割技术将逐渐实现开放世界的“万物分割”；结合大规模语言模型的成功与经验，分割技术也将通向更高层语义关系的建模与推理。

未来，美团街景理解团队将不断推动视觉技术在街景理解中的应用与演进，为场景重建、自动驾驶、机器人导航等应用场景提供更加高效便捷的技术支持。

7. 作者简介

金明、旺旺、祎婷、兴岳、钧峰等，均来自美团基础研发平台 / 视觉智能部。

8. 参考文献

[1] Cordts, Marius and Omran, Mohamed and Ramos, Sebastian and Rehfeld, Timo and Enzweiler, Markus and Benenson, Rodrigo and Franke, Uwe and Roth, Stefan and Schiele, Bernt. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016.

[2] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding. In ICCV, 2021.

[3] Kirillov, Alexander and He, Kaiming and Girshick, Ross and Rother, Carsten and Doll{\ ‘a}r, Piotr. Panoptic segmentation. In CVPR, 2019.

[4] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[5] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, 2015.

[6] Chen, Liang-Chieh and Papandreou, George and Kokkinos, Iasonas and Murphy, Kevin and Yuille, Alan L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI, 2017.

[7] Yuan, Yuhui, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In ECCV, 2020.

[8] Xie, Enze and Wang, Wenhai and Yu, Zhiding and Anandkumar, Anima and Alvarez, Jose M and Luo, Ping. SegFormer: Simple and efficient design for semantic segmentation with transformers. In NeurIPS, 2021.

[9] He, Kaiming and Gkioxari, Georgia and Doll{\ ‘a}r, Piotr and Girshick, Ross. Mask r-cnn. In ICCV, 2017.

[10] Bolya, Daniel and Zhou, Chong and Xiao, Fanyi and Lee, Yong Jael. Yolact: Real-time instance segmentation. In ICCV, 2019.

[11] Bolya, Daniel and Zhou, Chong and Xiao, Fanyi and Lee, Yong Jae. Solo: Segmenting objects by locations. In ECCV, 2020.

[12] Mohan, Rohit and Valada, Abhinav. Efficientps: Efficient panoptic segmentation. IJCV, 2021.

[13] Cheng, Bowen and Collins, Maxwell D and Zhu, Yukun and Liu, Ting and Huang, Thomas S and Adam, Hartwig and Chen, Liang-Chieh. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In CVPR, 2020.

[14] Xiong, Yuwen and Liao, Renjie and Zhao, Hengshuang and Hu, Rui and Bai, Min and Yumer, Ersin and Urtasun, Raquel. Upsnet: A unified panoptic segmentation network. In CVPR, 2019.

[15] Zhou, Tianfei and Porikli, Fatih and Crandall, David J and Van Gool, Luc and Wang, Wenguan. A survey on deep learning technique for video segmentation. TPAMI, 2022.

[16] Caelles, Sergi and Maninis, Kevis-Kokitsi and Pont-Tuset, Jordi and Leal-Taix{\ ‘e}, Laura and Cremers, Daniel and Van Gool, Luc. One-shot video object segmentation. In

CVPR, 2017.
[17] Zhou, Tianfei and Li, Jianwu and Wang, Shunzhou and Tao, Ran and Shen, Jianbing. Matnet: Motion-attentive transition network for zero-shot video object segmentation. TIP, 2020.
[18] Wang, Qiang and Zhang, Li and Bertinetto, Luca and Hu, Weiming and Torr, Philip HS. Fast online object tracking and segmentation: A unifying approach. In CVPR, 2019.
[19] Oh, Seoung Wug and Lee, Joon-Young and Xu, Ning and Kim, Seon Joo. Video object segmentation using space-time memory networks. In ICCV, 2019.
[20] Yang, Zongxin, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In NeurIPS 2021.
[21] Cheng, Ho Kei, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In CVPR, 2021.
[22] Ye, Linwei and Rochan, Mrigank and Liu, Zhi and Zhang, Xiaoqin and Wang, Yang. Referring segmentation in images and videos with cross-modal self-attention network. TPAMI, 2021.
[23] Yu, Changqian and Wang, Jingbo and Peng, Chao and Gao, Changxin and Yu, Gang and Sang, Nong. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In ECCV 2018.
[24] Fan, Mingyuan and Lai, Shenqi and Huang, Junshi and Wei, Xiaoming and Chai, Zhenhua and Luo, Junfeng and Wei, Xiaolin. Rethinking bisenet for real-time semantic segmentation. In CVPR, 2021.
[25] Zhao, Hengshuang and Shi, Jianping and Qi, Xiaojuan and Wang, Xiaogang and Jia, Jiaya. Pyramid scene parsing network. In CVPR, 2017.
[26] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In ICIP 2021.
[27] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In CVPR, 2016.
[28] SouYoung Jin, Aruni RoyChowdhury, Huaizu Jiang, Ashish Singh, Aditya Prasad, Deep Chakraborty, and Erik Learned- Miller. Unsupervised hard example mining from videos for improved object detection. In ECCV, 2018.
[29] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In CVPR, 2019.
[30] Lin, Di and Dai, Jifeng and Jia, Jiaya and He, Kaiming and Sun, Jian. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In CVPR, 2016.
[31] Chen, Xi and Zhao, Zhiyan and Zhang, Yilei and Duan, Manni and Qi, Donglian and Zhao, Hengshuang. Focalclick: Towards practical interactive image segmentation. In CVPR, 2022.
[32] Zhang, Chi and Lin, Guosheng and Liu, Fayao and Yao, Rui and Shen, Chunhua. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In CVPR, 2019.
[33] He, Shuting, Henghui Ding, and Wei Jiang. Primitive generation and semantic-related alignment for universal zero-shot segmentation. In CVPR, 2023.
[34] Zhang, Xiangyu and Zhou, Xinyu and Lin, Mengxiao and Sun, Jian. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In CVPR, 2018.

[35] Howard, Andrew G and Zhu, Menglong and Chen, Bo and Kalenichenko, Dmitry and Wang, Weijun and Weyand, Tobias and Andreetto, Marco and Adam, Hartwig. Mobilenets: Efficient convolutional neural networks for mobile vision applications. Arxiv, 2017.

[36] Cheng, Bowen, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In NeurIPS, 2021.

[37] Jain, Jitesh and Li, Jiachen and Chiu, Mang Tik and Hassani, Ali and Orlov, Nikita and Shi, Humphrey. Oneformer: One transformer to rule universal image segmentation. In CVPR, 2023.

[38] Kirillov, Alexander and Mintun, Eric and Ravi, Nikhila and Mao, Hanzi and Rolland, Chloe and Gustafson, Laura and Xiao, Tete and Whitehead, Spencer and Berg, Alexander C and Lo, Wan-Yen and others. Segment anything. Arxiv, 2023.

[39] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In ECCV, 2008.

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll ar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014.

[41] OpenAI.https://openai.com/blog/chatgpt. 2022.

[42] Rombach, Robin and Blattmann, Andreas and Lorenz, Dominik and Esser, Patrick and Ommer, Bj{\ “o}rn. High-resolution image synthesis with latent diffusion models. In CVPR, 2022.

[43] Hu, Yihan and Yang, Jiazhi and Chen, Li and Li, Keyu and Sima, Chonghao and Zhu, Xizhou and Chai, Siqi and Du, Senyao and Lin, Tianwei and Wang, Wenhai and others. Planning-oriented autonomous driving. In CVPR, 2023.

[44] Su, Jinming and Yin, Ruihong and Zhang, Shuaibin and Luo, Junfeng. Motion-state Alignment for Video Semantic Segmentation. In CVPRW, 2023.

[45] Su, Jinming and Yin, Ruihong and Chen, Xingyue and Luo, Junfeng. Perceive, Excavate and Purify: A Novel Object Mining Framework for Instance Segmentation. In CVPRW, 2023.

美团外卖推荐关于用户新颖体验优化的技术探索

作者：亚成 若静 俊洁等

1. 背景介绍

1.1 为什么要优化用户新颖性?

2022 年初，美团外卖搜索推荐技术团队通过内部访谈、外部用研，发现用户对外卖首页 Feed 推荐的新颖性问题诟病较多：首页 Feed 推荐了过多的复购复点商家，无法满足用户尝新的诉求。

我们通过分析，发现主要原因是：历史上策略优化[1]目标都聚焦在提升交易效率（RPM）和全局 GMV 上，对用户兴趣的探索不足，因而在外卖首页 Feed 的前几位，都是为用户重复曝光了过多复购复点商家，而现有策略也只是对历史购买商家进行降权排序，效果极其有限。随着业务的发展，如果首页 Feed 推荐的优化目标仍旧只关注交易效率而忽视用户新颖体验，长此以往用户难以跳出“信息茧房”[2]，其浏览体验也会变得越来越差。因此，我们将用户**新颖性**加入了外卖首页 Feed 推荐的优化目标中。

1.2 外卖场景下新颖性推荐的重要性

业界很多电商平台的推荐系统，候选池动辄几亿 ~ 几十亿不等，待分发内容非常多，重点都是做相关性优化以提升推荐准确度，往往直接过滤掉用户曾经看过、买过的内容商品，新颖性问题并不突出。但在外卖场景下，候选商家少、用户复购 / 下单频次高，决定了“直接过滤用户看过、下单过”的策略既不现实、也不合理。

一方面，受 LBS 约束，外卖候选商家量级在几百 ~ 几千，曝光商家既要匹配用户饮食习惯，又要考虑用户复购的需求，因而难免重复曝光部分商家；另一方面，外卖用户下单频次很高，不希望一天或者几天之内反复品尝同样的食物，用户有着很强的尝新诉求。故而新颖性推荐是用户体验的核心问题之一，平衡好用户的复购、尝新意图，显得极为关键。

2. 问题和挑战

新颖性推荐的目标是优化用户新颖体验，和首页 Feed 一直以来的优化目标（交易效率 RPM、全局 GMV）区别较大。具体到餐饮场景，优化新颖性主要的问题如下：

1. 对用户来说，什么是新颖性？即如何将用户的直接感受，量化为准确的新颖性定义。
2. 新颖性的优化目标如何设计？即如何将新颖定义转化为推荐系统的观测指标，再进一步，如何量化系统观测指标和用户新颖感受之间的变化关系？
3. 新颖性问题是否存在解决的终态？即对比持续迭代的效率问题，新颖性问题是能被彻底解决，还是需要持续迭代。

3. 新颖性概述

美团外卖推荐系统评估指标，长期以来主要包含效率指标（UV_CTR、UV_CXR、UV_RPM 等）、生态指标（主题曝光占比、商品曝光占比、闪购曝光占比等）、用户体验指标（多样性）等多个方面。但是要优化新颖体验，必须要有科学合理、策略可干预的新颖性指标，而这又依赖于“新颖”的定义。

3.1 什么是用户新颖性

首先，我们要回答“对用户来说，什么是新颖的”。外卖首页 Feed，承载了商家、商品、主题卡片等多种异构流量的分发，其中用户看到的卡片 90% 左右都是商家卡片，为此，我们认为用户新颖性意味着其看到的商家是否新颖。

如何判断一个商家是否新颖？一方面，业界[3][4][5][6]给出的新颖定义主要有：新颖性（Novelty/Unexpectedness）、惊喜度（Serendipity）。新颖性衡量的是给用户推荐没有见过的物品，而惊喜度（Serendipity）衡量的是推荐和用户历史兴趣不相似、却满意的推荐。受外卖 LBS 供给少的限制，要求“用户没见过”不太合理。

与此同时，为用户带来更好选购体验，直觉上看，新颖商家至少应该是让用户感到惊喜的推荐结果，看上去惊喜度的定义更满足要求。另一方面，在用户调研中发现：引起用户反感的在于“一直给用户推荐同几家”商家，导致其没有机会看到更多“没见过但是适合自己口味的新商家”。所以，惊喜度要求“历史兴趣不相似”也不完全满足实际业务需求。

为此，综合考虑外卖候选商家少、用户高复购高下单频次的特性，并结合了业界惊喜度、新颖性的定义和用研结论，我们认为：新颖商家，应该满足“用户近一段时间没见过、没吃过，但却愿意为推荐买单”的要求。出于业务经验，实际采用的新颖商家定义是：用户在外卖全局最近 7 天没有见过、或者 30 天没有点击、或者 90 天没有完单的商家。可以看出，我们的定义和传统新颖性（ Novelty/Unexpectedness）有着显著的差异。

3.2 新颖性评估体系

从上一章节的分析看，新颖性推荐要首先解决在短时间内反复推荐“来来回回那几家商家”的问题，为用户曝光更多新颖商家以满足其尝新探索需求，进而提升其推荐满意度。

“曝光 -> 点击 -> 下单”选购链路中的曝光环节，直接决定了用户感知“推荐结果好坏”的第一印象，覆盖的用户也更全，因此我们将“曝光新颖性 @Top N”（下文简称**曝光新颖性**），而不是点击 / 完单新颖性作为新颖性推荐的核心优化指标，其物理意义是用户在首页 Feed 列表前 N 位看到的新颖商家占比。再进一步，我们认为列表前几位对用户体验影响更加明显，因此“前 N 个 Item”能更好地体现用户兴趣区域的新颖性变化，实际中 N=10。

虽然，曝光新颖性避免了无法衡量非点击 / 完单用户的新颖性变化、点击 / 完单环节的漏斗过深带来干预效果不明显的问题，但如果只关注曝光漏斗，极端情况就是给用户全部曝光新颖商家，一旦新颖性商家质量很差那就是“反感”而不是“惊喜”，无法满足上文中要求新颖商家中“用户愿意为推荐买单”的约束。为此，我们引入了点击新颖性、完单新颖性（点击、完单商家中新颖商家的占比）作为辅助观测指标，并将列表 UV_RPM（千人曝光 GMV）作为核心约束指标（要求列表 UV_RPM 不降或微降）。

对我们来说，“曝光新颖性”为策略优化明确了方向，并能很好的衡量业务经验上认为的新颖性变化。如果该指标提升明显，用户的新颖性感受仍一成不变，那也并不合理。为此，实际中我们通过用研对用户投放新颖性问卷，并引入了新颖好评率（用户对推荐新颖性由低到高打分 1-5 分，表示 >=4 分用户的占比）、新颖性 Case 率（问卷中主动反馈新颖体验 Bad Case 的比例）2 个指标来观测“曝光新颖性”改善后用户新颖感受变化。

4. 新颖商家推荐方案

首页 Feed 的推荐链路（如下图 1 所示）包含“召回 –> 精排 –> 混排”3 个阶段，其优化目标都以效率为主，模型泛化性、策略的探索性不足，对曝光新颖性造成了不同程度的损害。所以，我们在推荐链路的各个阶段都进行了新颖推荐优化，其中：召回和精排阶段的目标在于优化模型的泛化性；混排阶段重点尝试优化推荐策略的探索性，同时因为其靠近业务最上游、受策略干预更为直接故而是策略优化的重心。在前端展示阶段，为了更直接地提升用户新颖体感，我们借助交互式推荐 [7] 的能力做了尝试。

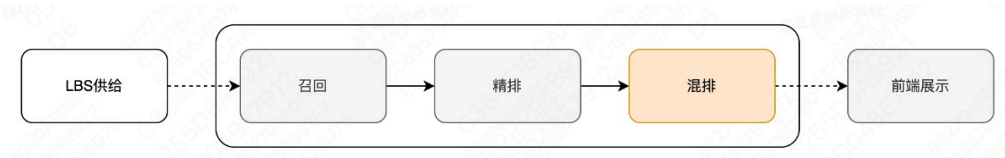


图 1　美团外卖首页 Feed 的推荐架构

新颖商家推荐整体方案如下图 2 所示。首先，我们围绕召回扩充新颖供给、精排提升模型泛化性、混排新颖商家个性化排序和浮动插卡、前端提升交互体验 4 个方面，系统性地构建了美团外卖首页 Feed 的新颖商家推荐方案。其次，我们也将业界常用的“直接过滤用户下单过”的策略升级为历史成单商家降位排序、实时点击成单商家分数重校准排序，以实现对新颖性负向体验的 Bad Case 兜底。因受限于篇幅限制，部分方案在本文中并未详述。

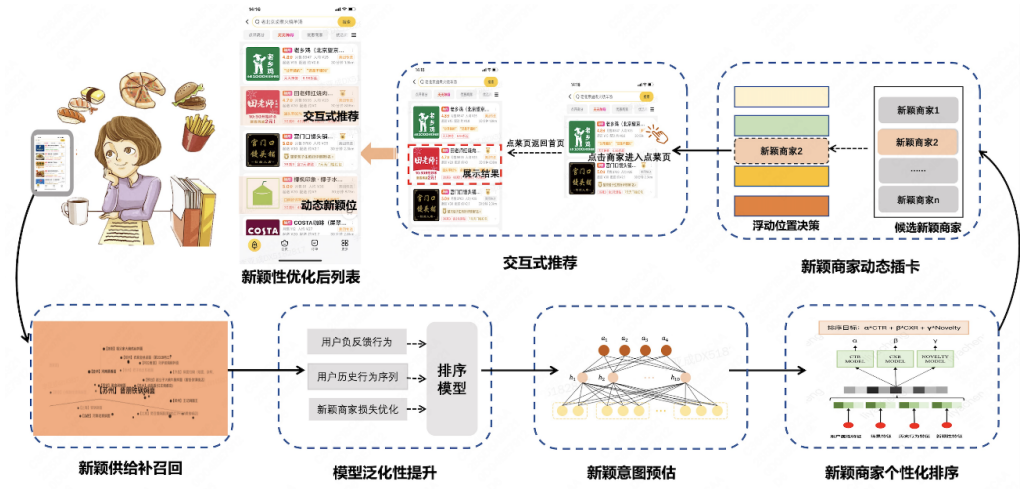


图 2　美团外卖首页 Feed 的新颖商家推荐方案

4.1 新颖商家供给补召回

召回阶段的新颖商家推荐和冷启动、长尾等问题有共通之处，常见冷启动解决思路有：a）利用对比学习 [8]、元学习 [9] 等方案使得模型在小样本上具备快速学习能力的；b）利用包括图学习 [11] 等方式挖掘和补充实体信息以获得更好的 Embedding 表征等。

在外卖场景下，新颖商家曝光高达平台商家曝光流量的 40%，因此无实体信息、样本少的问题在新颖推荐任务中并不突出。我们的问题更多是模型泛化性不足，高估了用户近期有过交互的商家因而推荐了过多“复购”商家，所以策略重心放在了通过优化 Embedding 表征来提升旁路召回的多样性上，具体工作包括：

首先，我们优化了 I2I 旁路召回。现有 I2I 召回的商家相似性计算依赖的是 Word2Vec 向量，主要存在 2 个问题：a）使用一段时间内的交互数据构建 I2I 的 Trigger，这类样本天然受到地理范围的限制，不能很好地捕捉跨地域的商家相似性；b）只考虑了商家序列，未利用用户、商家侧丰富的 Side-information，对新用户、长尾商家不友好。

而主召回链路的双塔 [11] 模型在训练中引入了丰富的商家侧 Side-information，通过对商家侧顶层 Embedding 表征进行聚类分析、Case 分析发现：商家侧的顶层 Embedding 表征在向量空间上有着显著的商家品类聚类效果（如图 3 左图所示），以及挖掘跨地域相似商家的能力（如图 3 右图所示，深圳的猪脚饭检索到遵义的猪脚饭），所以我们尝试用双塔模型的商家 Embedding 替换原有基于 Word2Vec 实现的商家 Embedding（Trigger 使用的是用户历史点击、完单商家）。同时，我们利用双塔商家 Embedding 向量，额外新增了一路 I2I 新颖商家召回以提升新颖商家的供给。

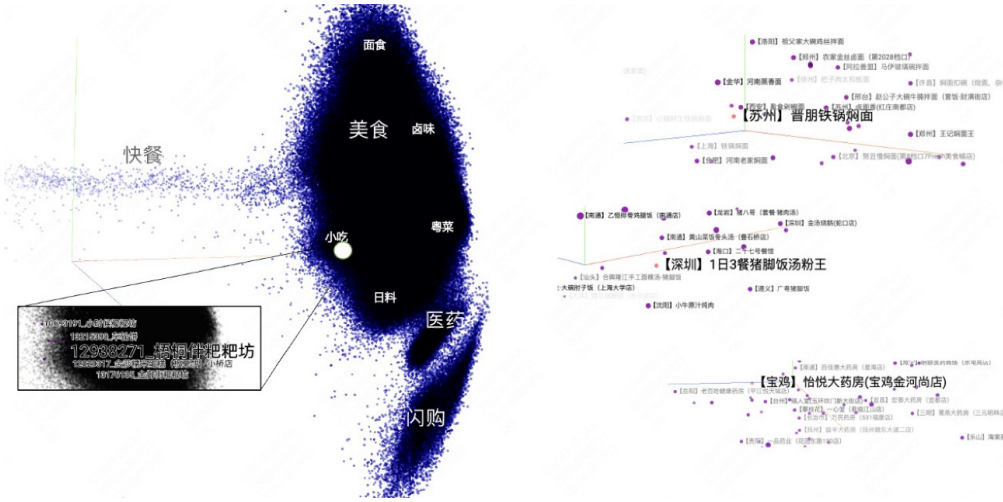


图 3　商家 Embedding PCA 压缩结果视图 && 相似商家检索结果

其次，我们新增了 GCN 旁路召回，利用 GCN 网络挖掘用户、商家之间的高阶关联性改善推荐长尾问题。实践中尝试了建模“用户 – 商家”关系的 U2I 召回和建模“Query—商家”关系的 I2I 召回。

这里以 U2I 召回为例介绍 GCN 建模：模型结构引入了用户行为构图、新颖性构图两部分以平衡效率、新颖性，损失函数如公式 1（r 表示完单、点击、新颖性任务）所示。用户行为构图包含用户点击、下单 2 个主任务；而新颖性构图部分，为解决无差别引入新颖商家导致模型噪音过大的问题，对完单、点击新颖商家进行了加权采样。最后，将新颖性采样子图作为辅助任务融入 2 个主任务的训练过程中，模型在构建高效的用户—商家表示的同时，也能具备发现新颖商家的能力。

$$Loss = \alpha_c \cdot L_c + \alpha_o \cdot L_o + \alpha_n \cdot L_n = \sum_{r \in R} \alpha_r (- \sum_{(u,v) \in r^+} log\sigma(y_{u,v}) - \sum_{(u,v') \in r^-} log\sigma(y_{u,v'})) \quad \text{（公式1）}$$

最后，线上实验发现，I2I 旁路召回 Embedding 向量表征优化带来了线上 1% 左右的曝光新颖性收益。而后续的 I2I 新颖商家召回、GCN 召回的方案，其新颖性收益都不足 0.5%，分析和外卖的 LBS 供给较少导致召回侧收益空间相对较低有关。初期从召回链路的改造成本考虑，我们的策略重点在旁路召回，但是随着迭代的进行，围绕双塔主召回也从 T+1 更新、模型损失等方向做了探索并取得了线上收益，在此不做详细的展开。

4.2 精排模型泛化性提升

首页 Feed 的排序模型主要以 UV_CXR 为导向，重点关注用户点击、完单等反馈，容易对历史点击成单商家高估、对长尾的新颖商家低估。为了提升模型对长尾新颖商家的预估准度，可以从模型特征、损失、结构等多个方面入手。

其中，相对模型特征和损失来说，结构优化和新颖性的关系存在更大的不确定性，为在短期内给用户体感带来明显的变化，优先投入到了引入泛化特征、优化模型损失两个方向上，线上用户曝光新颖性累计提升 3%+。

4.2.1 正负反馈特征

一方面，为更好地捕捉用户尝新、复购意图的变化，我们重点优化了负反馈特征和长序列特征。首先，在模型中引入负反馈特征以便于模型学习更好捕捉“负反馈模式”。其次，我们通过行为数据，来更好捕捉用户尝新、复购意图等多个兴趣的周期变化，为了降低长序列带来的训练和推理的压力，除工程的优化外，我们参考了 SIM[12] 的工作。

首先，我们将候选商家的品类 ID 当作 Query，从用户的行为序列中检索出相同品类的商家，再进行兴趣建模。具体的，我们结合外卖的 LBS、就餐特点，设计了 Distance（用户到商家的距离小于 xx Km）、Mealtime（将一天分为早餐、午餐、下午茶、晚餐和宵夜）和 Tag_id（商家品类，比如奶茶、甜点等）这 3 种检索路径，从用户历史行为中检索出与当次访问最相关性的行为，以精确刻画当前情境下的用户偏好。最终，线上实验 UV_RPM 显著增加，用户曝光新颖性 +1.0%。另一方面，我们也在探索头图、菜品属性等多模态特征对于提升模型泛化性的帮助。

4.2.2 新颖商家预估分纠偏

非凸优化导致 DNN 模型存在不可重复性 [13][14] 的问题：使用同样的架构 + 数据 + 超参，训练两版模型，其统计指标比如 AUC 可能接近，但是具体到每个样本的预测结果，却可能存在较大差异。

造成这个现象的原因有很多，比如随机初始化、分布式训练本来就自带随机性、数值误差等 [15]。具体到新颖性问题，表现在：虽然我们训练得到的 DNN 模型的准确率十分接近，但对于特定的新颖性样本，其预估分很可能相差较大（明显偏高或偏低），因此通过设计新的优化目标，在保证模型预估准确率不变情况下，使得模型到达的局部最优状

态偏向“新颖商家预估分比较高”。这样，就可以在在不损失列表 UV_RPM 的前提下，提升新颖商家的曝光量。

在实际中，我们在现有的交叉熵损失的基础上，结合新颖商家在曝光、点击、成单行为上的差异，对新颖商家引入如公式 2 所示的损失。线上实验发现，列表 UV_RPM 持平的情况下，曝光新颖性 +1.95%。

$$Loss = \begin{cases} -0.1log(p+1) & if \text{ 曝光未点击} \\ -0.2log(p+1) & if \text{ 点击未成单} \\ -0.4log(p+1) & if \text{ 成单} \end{cases} \quad \text{（公式2）}$$

4.3 混排个性化排序和动态插卡策略

混排阶段，我们的目标是通过公式加权排序、固定位插卡、实时 / 历史成单降权排序等多种手段优化推荐系统的探索性以提升用户新颖性。这几种手段的优点是可解释性强，但是转化率都比较低。

业务上认为：每个用户都有尝新诉求，差异在于其尝新需求的强弱，新颖性推荐应该和用户尝新意愿相匹配。如果调整后的新颖推荐结果无法精准匹配用户尝新诉求，反而很容易带来更差的用户体验。为此，我们根据预估的用户新颖意图强弱，实现了基于 ES 模型的新颖商家个性化排序策略，并通过动态位置分配实现了新颖商家的浮动插入。混排阶段的整体优化，用户曝光新颖性累计提升 15%+。

4.3.1 新颖意图预估

初期，通过数据分析找到和用户新颖性较为相关的特征，利用人工设计的公式计算每个用户的新颖意图分，线上实验发现新颖性有较为明显的提升。但是，人工设计公式的方式迭代效率、准确率都较低，无法即时捕捉用户在不同时段场景下的新颖需求变化，长此以往，难以有效支持整个新颖推荐策略的迭代。为此，我们构建了新颖性意图预估模型，通过模型网络抽象出多个行为特征之间的交叉信息，预估用户在当前场景下的即时意图分，以满足不同用户在不同情境下的新颖商家浏览、消费需求。

模型设计思路简述如下：首先，模型 Label 采用的是用户在当前请求下是否点击新颖商家，我们主要通过实验比较了新颖点击、完单等多种方案确定了该 Label，这也从侧面

印证了选择漏斗更为靠前“曝光新颖性”作为优化指标的合理性。其次，模型特征上除用户统计特征、上下文场景特征外，主要引入了完单新颖商家占比、新颖商家曝光点击等新颖性相关的特征，损失函数使用了交叉熵损失。最终，考虑线上系统性能影响，模型使用了简单的 MLP 网络，线上服务阶段，为降低系统耗时，采用了模型并行调用的方案。线上实验在 UV_RPM 持平的情况下，用户曝光新颖性 +2.28%。

4.3.2 新颖商家个性化排序

在混排阶段，外卖推荐为了灵活兼顾效率、生态、业务规则等多种目标参考了 Linkedin[16]、Facebook[17] 相关的工作，从帕累托最优的角度出发主要通过形如“a*pctr+b*pcxr+c*other”的公式对候选内容进行异构排序。

同时，为了实现对不同场景下用户的差异化、个性化排序，我们基于 ES 算法 [18] 针对 pCTR、pCXR 等核心效率目标建设了个性化的超参数的方案，其核心思想是：通过扰动神经网络参数，搜集用户的反馈计算 Reward 并确定模型参数更新的方向和比例，并输出排序参数应用于对应的请求，实现个性化的排序策略。

初期为提升迭代速度，我们直接采用“在排序公式中使用人工设置的固定系数加权”的方案对新颖商家排序，取得了一定收益，但是 UV_RPM 损失较大，原因在于无法根据用户在不同时段、场景下的个性化新颖消费需求进行差异化排序。为此，我们在后续迭代中借鉴已有的 ES 个性化超参数方案，实现了对新颖商家“千人千面”个性化排序。

考虑到外卖推荐系统的业务场景复杂度、用户行为易受多方面因素影响（优惠券、业务干预等）等问题，基于新颖性优化的 ES 模型方案具体如下：

首先，在 Reward 方面，为引导模型学习效率、新颖性之间的关系，在原有 Reward 的基础上增加了新颖性，最终 Reward 为 ：a * cxr + b* 新颖商家曝光占比，既提升新颖商家曝光占比指标，又能尽可能减少对列表 UVRPM 的损伤。进一步的，为了更好地衡量模型动作带来的净收益，在 Reward 中引入了优势函数 [19]V(s, a) = Q(s, a) − V(s)，即在原本的动作状态价值 Q(s, a) 的基础上，减去 Batch 内 Reward 的均值，使模型学习到动作相较于平均而言带来的优势 V(s, a)，进一步为解决 Batch 内均值无法消除人群状态价值偏差的问题，我们通过模型预估了用户个性化的状态价值代替了原有的“Batch 内 reward 均值”方案。

其次，在模型特征方面，我们在原有特征基础上，通过决策树模型筛选出与新颖性相关度较高的用户场景特征、新颖性特征、历史行为特征等。最后，在模型网络方面参考已有经验选择结构简单的 MLP 网络。经过一周左右的在线学习后，ES 模型决策 Action 分布逐渐趋向稳定状态，如下图 4 所示：

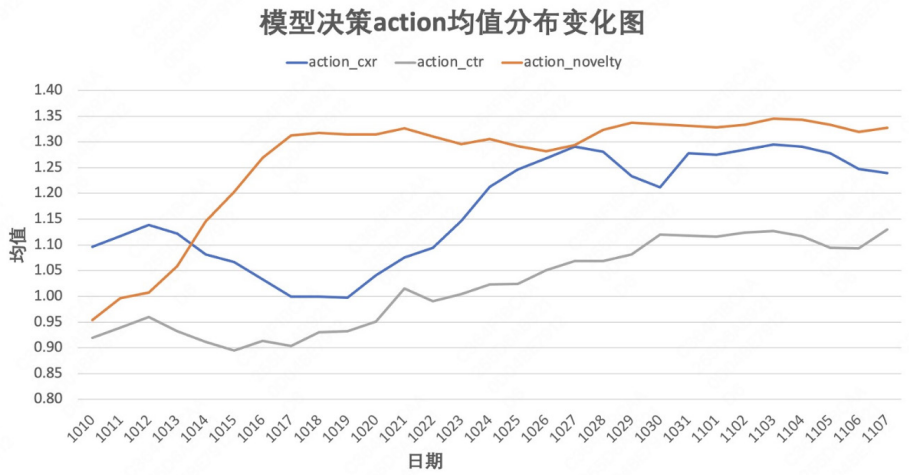


图 4　ES 模型决策 action 均值变化趋势

分析发现，ES 模型在兼顾效率的前提下，实现了新颖商家的个性化排序。如图 5 所示的 Case，展示了一个实际中有明显新颖偏好的用户：首先相较于上线第 1 天，模型在第 20 天已经能稳定、有效地学习到该用户新颖倾向；其次，模型 action_novelty 在不同时段下，也学习到了该用户新颖偏好的变化。最终，线上实验在 UV_RPM 持平的情况下，用户曝光新颖性累计提升 5+%。

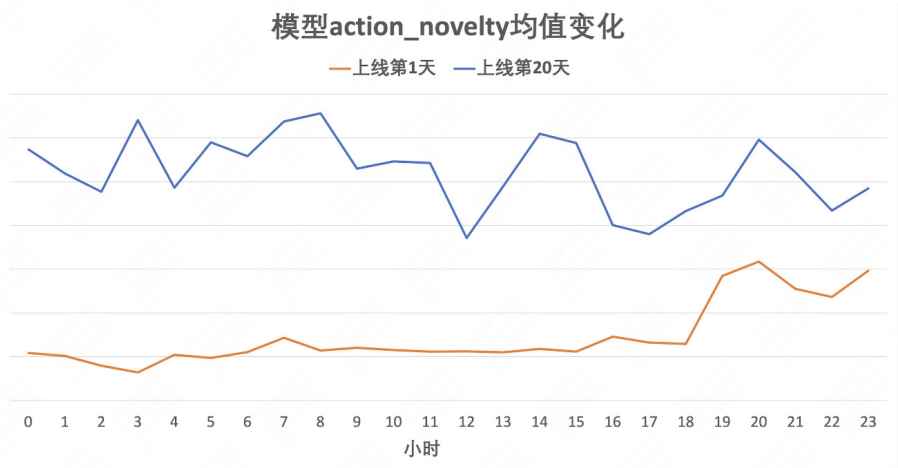


图 5　模型新颖性商家加权系数均值随时段变化趋势图

4.3.3 新颖商家动态插卡

在上一章节，新颖商家个性化排序的方案容易出现新颖商家“扎堆”的现象，用户体感不好，简单的方案是采用“固定位”的方案插入新颖商家。为了解决固定位流量转化效率低、不考虑用户浏览深度而导致曝光效率低的问题，我们设计了个性化的新颖商家浮动插入方案：依据用户新颖意图，在合适的位置，插入合适的新颖商家。

初期，我们通过两种简单策略验证动态插卡的可行性：方案 1 是“按位置均匀插卡”，用户在前 N1、N2 个位置至少要有 M1、M2 个新颖商家，如果没有，就应该在 N1、N2 个位置进行插入。方案 2 是“结合用户浏览深度设计个性化位置”，结合用户新颖意图、用户浏览深度设计了“意图越强，插卡数量越多”的策略，其中插卡位置根据用户历史浏览深度的分位数计算得出。实验证明，这两种策略对新颖性都有明显的收益。

但是，这两种决策插卡位置的策略，对用户个性化信息利用不足，既没有考虑用户在不同位置的新颖偏好、也没有考虑上下文对当前插入新颖商家的影响。所以，我们迭代了模型化方案。

在模型选型上，选择了强化学习的方案，其原因在于：一方面，基于监督学习的 ML 算法往往是通过拟合用户历史行为数据得到预估结果，难以捕捉外卖场景下快速转换的用户尝新、复购意图（高频用户一天可能转化多次）；另一方面，监督学习往往基于贪心策略给出单次最优结果，无法关注用户长期收益的变化。而强化学习通过不同

状态决策不同的动作，始终建模累计收益，能够有效的进行 E&E（Exploration and Exploitation），因此更符合新颖商家动态插卡的要求。

首先，我们参考了广告流量分配 [20] 问题，将新颖商家动卡插卡的问题简化建模如公式 3 所示的业务问题：在 UV_RPM、流量占比、用户体验的约束下，通过决策哪些位置放入新颖商家，以最大化用户新颖性。其中效率约束表示插卡前后列表 GMV 降幅不高于 P，流量约束表示整体新颖商家曝光占比，不高于某个阈值以避免极端问题。此外，还有插卡位置打散、插卡位置上限等体验约束。

$$
\begin{aligned}
&Max \sum_{k}\sum_{x_{i,k}} \frac{I(x_{i,k}=1)}{I(x_{i,k})} \\
&s.t.\text{效率：} \sum_{k}\sum_{x_{i,k}} gmv >= p * \sum_{k}\sum_{x_{j,k}} gmv \\
&\text{流量：} \sum_{k} \frac{I(x_{i,k}=1)}{I(x_{i,k})} <= r \\
&\forall i,k \quad x_{i,k} = \begin{cases} 1 \text{ if } x_{i,k} = \text{新颖商家} \\ 0 \text{ if } x_{i,k} = \text{非新颖商家} \end{cases}
\end{aligned}
\qquad \text{（公式3）}
$$

其次，这个问题属于典型的多臂老虎机（Multi-Armed Bandit）问题：无论当前（列表前 10 位）位置是否可以插入新颖商家，我们都希望列表 RPM 期望最大，而 MAB 就可以基于 RPM 期望对当前是否插入新商家的决策进行 Explore 或者 Exploit，也最小化列表 RPM 损失。常见的 MAB 算法有 LinUCB、Thompson 采样等。

我们基于 Thompson 的方案做了尝试，验证传统的强化学习方案在位置决策上的效果：具体的，利用 Beta 分布拟合用户在每个位置的新颖、非新颖商家完单期望，每次只需要选择对用户 RPM 的收益最大的方式（该位置下，插入一个新颖商家的完单期望，高于原有非新颖商家的完单期望），来决策当前位置是否插入新颖商家。其原理在于 Thompson 采样通过 Beta 分布维护用户在当前位置关于新颖、非新颖商家的完单期望。

Beta 分布（见如下公式 4）我们通过统计用户在每个位置的历史新颖、非新颖商家的曝光 / 完单（或者点击，用于解决完单数据稀疏的问题）历史数据拟合用户偏好，其期望就是对应位置下新颖商家、非新颖商家的 CXR。每次决策的时候，只需要比较新颖商

家、非新颖商家的 Beta 分布完单期望（见公式 5），新颖商家期望更大则在当前位置插入新颖商家，同时这种多臂赌博机的思想能根据用户反馈不断拟合用户逐位置的新颖偏好。其中 Intentionscore 表示 4.3.1 节的新颖性意图分。

$$Beta_{novelty} = Beta(View_{novelty}, Order_{novelty})$$
$$Beta_{non_novelty} = Beta(View_{non_novelty}, Order_{non_novelty}) \quad \text{（公式4）}$$

$$Intention_{score} * Beta_{novelty} > \alpha * Beta_{non_novelty} \quad \text{（公式5）}$$

为解决 Thompson 采样等算法泛化性不足的问题，探索了深度强化学习的方案。考虑到 DQN 实现简单、其 Off-policty 非常适合推荐场景的优点，我们最开始就参考了 Cross DQN[21]、DEAR[22] 的模型结构，通过 D3QN 模型决策当前用户可以插卡新颖商家的位置：模型输出的是 N 维的 Multi-hot vector（1 表示当前位置插卡新颖商家），State 考虑了用户和商家历史统计特征、列表上下文的信息等，奖励函数考虑了完单金额、曝光新颖性以及用户体验。该方案仍在优化当中，不再展开介绍。

最终，线上实验，UV_RPM 跌幅不超过 0.5% 的情况下，用户曝光新颖性提升 8%+。

4.4 交互推荐优化选购体验

交互式推荐是一种实时推荐产品模块，主要利用用户的实时行为、以实时互动的方式进行推荐：用户从首页 Feed 进入商家详情页并退出之后，动态地插入新的内容到推荐列表中。优势在于能够实时感知用户行为并判断用户意图，进而动态插卡反馈，增强用户交互体验、新颖感受。

在不损失用户 UV_RPM 的前提下，我们在现有的效率（pCXR）排序的基础上，引入了新颖性目标，进一步优化用户新颖体验。线上实验用户曝光新颖性提升 +1.20%。

5. 总结与展望

本文介绍了外卖推荐团队 2022 年至今的用户新颖体验整体优化方案，主要包括：

1. 从“什么是新颖性”问题出发，出于业务经验设计了外卖餐饮场景下的“新颖商家”的定义，确定了“曝光新颖性”的优化指标和与之相关的用户体验指标

“新颖好评率”、“主动反馈率”等，用于指导新颖性推荐的策略优化。

2. 围绕“如何优化新颖性”的问题，从用户新颖意图预估出发，构建了“优化旁路召回扩充新颖商家供给，在模型中引入用户正负反馈行为和新颖性目标提升新颖商家预估准度；利用强化学习，实现新颖商家个性化排序和千人千面的动态插卡；引入交互式推荐，以实时互动的方式给用户带来又新又好选购体验”的方案；系统地提升了外卖推荐的泛化性、发现性，以更好匹配用户尝新的意图。

用户新颖体验提升明显：用户前 10 位平均比之前多看到 0.6 个新颖商家，其中业务指标收益如下：

1. **系统优化指标**：曝光新颖性累计提升 19%+，完单新颖性提升 25%+，用户平均浏览深度提升 1%+，用户平均曝光次数提升 1.5%+。
2. **用户直接体验指标**：新颖性五星好评率提升 7%+，一星差评率降低 4%+，用户主动反馈率降低 18%+。

优化过程中我们也看到：

1. 当前大部分效率优化策略会损失新颖性：在较长的一段时间内，需要持续关注并优化用户新颖体验，以避免随着效率的持续优化导致用户新颖性退化到之前水平。
2. 对比低频用户，中高频用户（月访问频次 >=13）其尝新意愿和推荐新颖性的匹配程度低 7%+，主动反馈新颖性的 Bad Case 的比例高 4 倍多，新颖性优化后的 DAU 变化更为正向：不同频次的用户群体新颖性诉求存在明显差异，其优化确实应以匹配用户尝新意图为准。

尽管我们做了很多优化，但仍存在较多问题亟待解决，未来我们将从以下几方面展开：

1. **新颖商家定义迭代**：继续探索和迭代什么样的新颖商家，更加科学合理地进行推荐，使用户愿意为之买单。
2. **新颖性推荐策略优化**：利用因果推断优化用户意图预估准度，从样本、模型结构、多模态等角度提升精排模型表征能力，探索强化学习下更为情境化的“一人多位”灵活位次插入能力，以及新颖体验优化和效率优化如何协同并进。
3. **交互形式迭代**：利用交互式推荐、重排序、视频内容推荐等产品形式为用户带来更好的交互体验。

一直以来，美团外卖都在努力为用户（实时性、多样性、满意率等）、商家（流量公平性分配等）、骑手（履约等）提供更好的体验而努力。我们希望帮大家吃得更好，生活更好。

6. 作者简介

亚成，若静，俊洁，小洋，国峰，李睿，昊雨，子相，彬彬，姜飞，北海等，来自到家事业群 / 到家研发平台 / 搜索推荐技术部。

7. 参考文献

[1] 美团外卖推荐情境化智能流量分发的实践与探索
[2] Li, Pan , et al. “PURS: Personalized Unexpected Recommender System for Improving User Satisfaction.” RecSys ‘20: Fourteenth ACM Conference on Recommender Systems ACM, 2020.
[3] Kaminskas, Marius , and D. Bridge . “Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems.” The ACM Transactions on Interactive Intelligent Systems 7.1(2016):1-42.
[4] Adamopoulos, Panagiotis , and A. Tuzhilin . On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected. ACM, 2015.
[5] Ge, Mouzhi , C. Delgado-Battenfeld , and D. Jannach . “Beyond accuracy: evaluating recommender systems by coverage and serendipity.” Acm Conference on Recommender Systems ACM, 2010.
[6] Chen, Li , et al. “How Serendipity Improves User Satisfaction with Recommendations? A Large-Scale User Evaluation.” The World Wide Web Conference 2019.
[7] 交互式推荐在外卖场景的应用
[8] Zhang, Yin , et al. “A Model of Two Tales: Dual Transfer Learning Framework for Improved Long-tail Item Recommendation.” (2020).
[9] Zhu, Yongchun , et al. “Learning to Warm Up Cold Item Embeddings for Cold-start Recommendation with Meta Scaling and Shifting Networks.” ACM (2021).
[10] Li, Feng , et al. “Graph Intention Network for Click-through Rate Prediction in Sponsored Search.” (2021).
[11] Hamilton, William L , R. Ying , and J. Leskovec . “Inductive Representation Learning on Large Graphs.” (2017).
[12] Pi Q, Zhou G, Zhang Y, et al. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. CIKM 2020.
[13] Shamir, Gil I. , D. Lin , and L. Coviello . “Smooth activations and reproducibility in deep networks.” (2020).
[14] Snapp, Robert R. , and G. I. Shamir . “Synthesizing Irreproducibility in Deep Networks.”, 10.48550/arXiv.2102.10696. 2021.
[15] Rohan Anil, Sandra Gadanho, Da Huang. “On the Factory Floor: ML Engineering for Industrial-Scale Ads Recommendation Models. “(2022)
[16] Yan, J., Xu, Z., Tiwana, B., and Chatterjee, S. (2020). Ads allocation in feed via constrained optimization. In Proceedings of the 26th ACM SIGKDD International

Conference on Knowledge Discovery & Data Mining, 3386 - 3394.
[17] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quinonero Candela. Practical lessons from predicting clicks on ads at facebook. In ADKDD, pages 5:1 - 5:9, 2014.
[18] Salimans, Tim , et al. "Evolution Strategies as a Scalable Alternative to Reinforcement Learning." (2017).
[19] Baird, L. C . "Advantage Updating." Advantage Updating (1993).
[20] Xie R, Zhang S, Wang R, et al. "Hierarchical reinforcement learning for integrated recommendation."(2021).
[21] Liao, Guogang , et al. "Cross DQN: Cross Deep Q Network for Ads Allocation in Feed." (2021).
[22] Zhao, Xiangyu , et al. "DEAR: Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems." (2019).

低延迟流式语音识别技术在人机语音交互场景中的实践

作者：正坤 鸿雨 李敏 飞飞 丁科 广鲁

1. 前言

1.1 语音识别技术简介

人机交互一直都是人工智能大背景下的“热门话题”，语音交互作为人机交互的一个重要分支，具有广泛的应用价值，也被应用到美团的多个业务场景中，如智能客服、电话营销和电话满意度反馈等。而流式语音识别技术是整个交互链条的入口，对交互体验影响巨大。

常见的语音识别大多都是非流式语音识别技术，它是指模型在用户说完一句话或一段话之后再进行识别。这意味着模型需要等待用户停顿或结束说话才能开始识别，并且只能在用户停顿或结束说话后才能输出完整的识别结果。这样做的缺点是会导致较长的延迟和不连贯的交互。例如，在会议场景中，如果使用非流式语音识别技术，就可能出现会议参与者说了很长时间的话才显示出他们所说的内容，而且可能因为网络延迟或其他原因导致内容显示不全或错误。这样就会影响会议参与者之间的沟通和理解，并且降低会议效率和质量。

而与之对应的是流式语音识别技术，它是指可以在处理音频流的过程中，支持实时返回识别结果的一类语音识别模型。这意味着模型不需要等待用户说完整句或整段话就可以开始识别，并且可以随着用户说话的进度逐渐输出识别结果。这样做的好处是能够大大减少人机交互过程中语音识别的处理时间，提高用户体验和交互效率。例如，在智能客服场景中，使用流式语音识别技术，就可以实现用户说一句话很快就能获得机器人响应，而不是等到用户说完一段话才给出回答。这样就可以让用户更快地得到满意的解决方案，并且减少用户的等待时间和不满情绪，提升用户满意度。在美团内部的众多业务场景中广泛使用了流式语音识别技术。

本文将详细阐述团队在语音交互场景中的低延迟流式语音识别方案，目前以该方案形

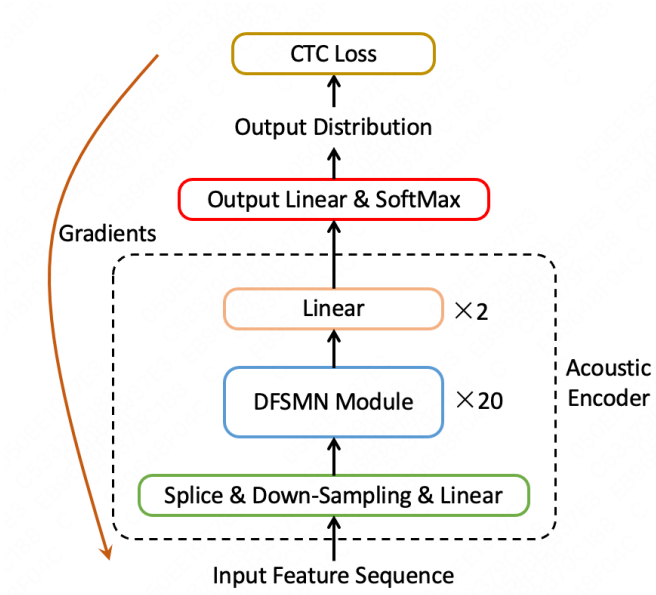
成的技术论文《Peak-First CTC: Reducing the Peak Latency of CTC Models by Applying Peak-First Regularization》已经被语音领域国际顶级会议 ICASSP 2023 收录。

1.2 问题与挑战

对一个好的流式语音识别系统而言，不仅仅需要高的识别准确率，还应该具有很低的延迟。在交互场景中，低延迟可以提高用户体验和满意度，让用户感觉不到语音识别的处理时间，更加自然和流畅地进行对话和问答。低延迟也可以减少通话交流中的误解和冲突，让用户能够及时地收到反馈结果，避免重复或打断对方的说话。此外，低延迟还可以增加语音应用的可用性和灵活性，让用户能够在各种场景下通过说话来完成任务（例如在线游戏、语音助手、智能家居等），节省下来的延迟也可以用于在语音服务的上下游部署更加复杂的模型结构，从而进一步完善交互链路等。

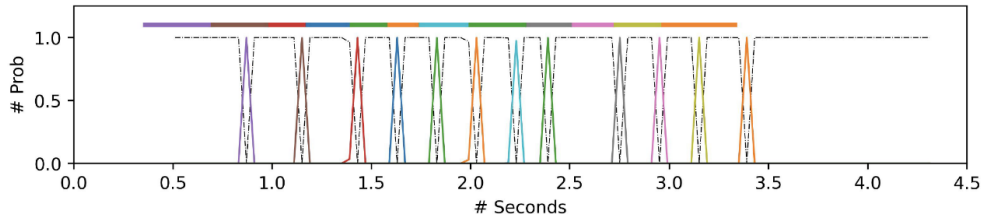
在美团的交互场景中，广泛使用联结时序分类模型（Connectionist Temporal Classification，CTC ）作为基础模型来构架流式语音识别系统。CTC 模型由于其优雅的模型结构、卓越的模型表现以及良好的扩展性受到了广泛的青睐。目前已经广泛应用在语音识别（Automatic Speech Recognition, ASR）、语音翻译（Speech Translation, ST）以及 光学字符识别（Optical Character Recognition, OCR）等领域。

下图展示了一种典型的 CTC 模型结构，其依赖 DFSMN 网络结构搭建，仅包含声学编码器（Acoustic Encoder）和输出线性映射层两部分。声学编码器用来将输出的声学特征序列转变成声学编码序列，而输出线性映射层则负责将利用声学编码表示，计算得到模型预测出不同文本标记的概率值。对比其他流式语音识别模型，CTC 模型不需要复杂的编码解码（Encoder-Decoder）结构或者注意力机制（Attention Mechanism）就能实现两个不等长序列之间的转换（对于语音识别而言是从声学特征序列转换到目标文本序列）。



基于 DFSMN-CTC 的语音识别模型结构

基于 CTC 的流式语音识别系统对于延迟也有着非常高的要求。从用户发音结束到系统识别出对应文字之间的时间差被称之为出字延迟。出字延迟越低则意味着 ASR 系统吐字的速度越快，用户体验越好。下图展示了 CTC 模型的输出概率分布，其中顶部色块表示用户说的每个文本的发声范围，而底部对应颜色的尖峰则表示系统识别出的文本所在的位置。出字延迟则对应着色块尾部与概率尖峰位置之间的时间差。本文所展示的工作就聚焦于如何降流式 CTC 语音识别系统的出字延迟。



CTC 模型的输出概率分布

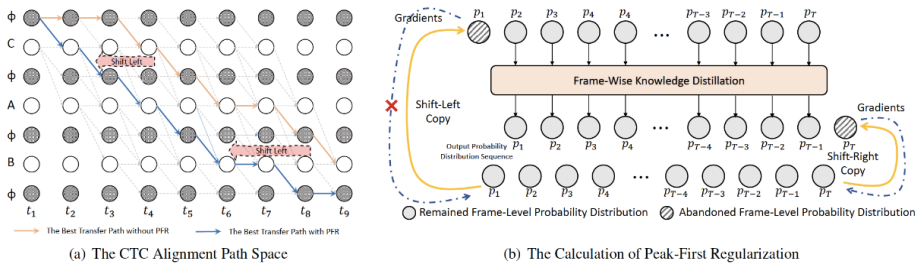
2. 尖峰优先正则化方法

2.1 CTC 模型基础

CTC 模型能够直接建模了声学序列到文本序列的转换关系，而不需要注意力机制等结构。由于文本序列的长度远远小于声学特征序列（通常情况下声学特征序列以帧作为单位，相邻两帧之间间隔为 10ms，时长为 1s 的语音就可以被划分为 100 帧），而在模型预测过程中，每帧特征都有一个预测标签。CTC 损失计算过程中引入了空格标记 ϕ 来作为填充标记符，以使得文本序列与声学序列的长度相等。

以下图 (a) 中所展示的 CTC 路径空间为例，其中横轴表示声学特征序列，纵轴表示目标文本序列。一条语音预测出文本序列“CAB”的概率可以被描述为后验概率 $P(CAB|X)$，为了方便计算损失，需要使用空格标记 φ 对文本标记填充，填充之后会出现与目标序列对应的多条路径（对应图 3(a) 中的实线与虚线，从图中左上角开始空格标记或者非空格标记开始，沿着线段转移至右下角空格或者非空格标记的路径均是一条可能的解码路径），所有路径的概率和等于后验概率 $P(CAB|X)$。为了避免路径穷举导致的计算爆炸问题，CTC 损失计算过程实际上使用了基于动态规划思想的前后向算法，来对所有可行的解码路径进行概率求和，并最终以负对数概率作为最终损失函数来进行优化。

2.2 尖峰优先正则化方法描述



(a) The CTC Alignment Path Space　　(b) The Calculation of Peak-First Regularization

(a)CTC 对齐路径空间 (b) 尖峰优先正则化计算过程

由于 CTC 的输出概率中蕴含着海量的可行解码路径，为了降低输出延迟，我们对所有的解码路径进行了仔细的分析和观察，如上面 (a) 图所示，网格中包含与文本“CAB”对应的多条可能的路径，以橙色和蓝色实线连接的路径为例，显然两条路径的转移位置存在明显区别，蓝色路径分别在 t_2，t_4 和 t_6 位置预测出字符 "C"，"A"，"B"；而橙色路

径则在 t_4，t_6 和 t_9 位置才预测出对应的字符。因此蓝色路径相对橙色路径具有更低的出字延迟，其从时间轴上来看，蓝色路径相对橙色路径更加靠左。基于这个观察，我们可以得出结论：具有低延迟的路径在时间轴上的非空格标记概率尖峰的位置会更加靠前一些。因此，我们提出了一个猜想，可以通过将 CTC 输出的概率分布整体左移的方式来降低模型的出字延迟。

基于这个假设，本文提出了一个简单的正则化方法 – 尖峰优先正则化方法（Peak-First Regularization, PFR），来使得 CTC 模型的模型输出的概率分布实现整体左移以降低出字延迟。PFR 方法巧妙使用了知识蒸馏的方法，迫使 CTC 输出概率分布的每一帧概率都学习其邻近下一帧的信息。如上图 (b) 所示，利用逐帧的知识蒸馏函数，使得每一帧的概率分布 都学习其后一帧，随着迭代的进行，模型实现了整体分布的左移。其损失计算过程可以被表述为以下形式：

$$\mathcal{L}_{PFR} = -\sum_{t=1}^{T-1} p_{t+1} \log \frac{p_{t+1}}{p_t}$$

该损失函数仅作为正则项使用，整体损失函数可以被描述为：

$$\mathcal{L} = \mathcal{L}_{\mathcal{CTC}} + \lambda \mathcal{L}_{PER}$$

其中 λ 作为权重，用来平衡两个损失之间的关系，避免输出概率持续移动最终导致训练崩溃的问题。PFR 正则项在学习过程中实际上呈现损失数值上升的态势，当输出分布不再移动的时候，损失值也趋于平衡。

虽然模型在训练过程中仅学习后面一帧（约等于 40 ms）的内容，但是随着训练结果的累积，可以获得远超 40ms 的延迟降低。这样设置有三方面的考虑：

- 首先，模型在完全学习到下一帧内容后，整体分布已经左移了 40ms。再继续学习后一帧内容，可以实现延迟效果的累加。
- 其次，考虑到 CTC 的输出概率分布是稀疏的，如果学习后面第 N 帧的内容，有非常大的可能性第 N 帧是空格标记，起不到学习效果，甚至学不到时序移动的关系。

- 最后，仅学习后面一帧的内容会降低训练难度，这种情况下经过平滑后的相邻两帧之间的概率分布的相似程度比较高，比较容易直接学习。如果直接学习后面第 N 帧的内容，也容易使得模型初始情况下面临更加困难的学习环境。

2.3 梯度分析

虽然本文通过间接的方法来实现降低延迟的效果，但是其仍然具有一定的解释性。可以对其梯度进行分析，假如 CTC 在第 t 时刻预测出第 k 个标记的概率是，则其梯度可以被描述为：

$$\frac{\partial \mathcal{L}}{\partial p_k^t} = -\frac{G(t,k) + \lambda p_k^{t+1}}{p_k^t}$$

其中 CTC 损失部分梯度为：

$$\frac{\partial \mathcal{L}}{\partial p_k^t} = -\frac{G(t,k)}{p_k^t}$$

通过公式可以知道概率和它邻近下一帧的概率紧密相关，如果下一帧预测出同一个标记 k 的概率很大（即下一帧是一个概率尖峰），则会促使当前帧梯度发生较大变化，进而实现概率分布左移，而如果下一帧的概率很小，则对当前帧梯度影响不大，不会产生概率分布位移。

3. 相关工作

伴随着智能交互技术的发展，大家对于交互体验的要求越来越高，如何降低语音识别系统的出字延迟再次成为了研究热点，各种思路层出不穷。整体来看方法可以被归结为以下四类。

3.1 强制对齐方法

强制对齐（Force Alignment）方法依托外部模型提前生成强制对齐标注信息。这些信息中包含用户发音与标注文本之间的准确对齐关系，在 CTC 或者 Transducer 模型损失计算过程中对路径的延迟进行限制，对具有高延迟的路径施加惩罚，以此来实现降低延迟的目的 [3,5]。这一方案将延迟作为约束引入到损失函数的计算过程中，需要修改损

失函数以及梯度的计算环节。

3.2 路径分解方法

路径分解方法以 FastEmit 方法为代表 [4]，主要应用到 RNNT 模型上，其对 RNNT 损失计算过程中的每个节点进行了路径分解，在损失函数的计算过程中，对低延迟路径赋予更高的权重，进而达成了鼓励模型在空格标记和非空格标记中优先预测非空格标记来降低出字延迟的目的。

基于这种逻辑训练出来的模型具有较低的延迟。虽然该方法摆脱了对于强制对齐的依赖，可以使得模型在训练过程中自然而然得地学习到低延迟路径，但是这种方法仍然需要修改损失函数前向计算环节和修改梯度的计算公式，具体实现相对复杂。

3.3 最小贝叶斯风险方法

最小贝叶斯风险方法（Bayes Risk CTC）方法将延迟作为贝叶斯风险值加入到损失函数的计算过程中 [6]。为了避免大量的计算，使用了分而治之的策略来将路径分组，同组内指定相同的延迟风险值。本方法虽然能够获得延迟降低，但是需要修改损失函数和梯度的计算过程，甚至需要模型方面的改动，增加了延迟优化问题的复杂度。

3.4 自对齐方法

自对齐方法（Self-Alignment）方法不需要修改损失函数的计算过程，而是从上一轮的模型的解码结果中选择低延迟路径，并将其作为正则项添加到本轮模型的优化过程中 [7]，这种方法虽然简化了计算量，但是面临新的问题，这种在线解码的方法需要消耗大量的解码时间，当面临海量数据的时候，在线解码会严重训练的进度，延缓训练流程。

综上所述，本文提出的延迟优化策略最为简单，不需要复杂的损失与梯度计算，也不依赖外部强制对齐结果，且在小数据和大规模生产数据上同样有效。

4. 评价指标

4.1 字错误率

字错误率（Character Error Rate, CER）用来衡量标注文本与识别文本之间的编辑距离。字错误越低则语音识别结果越好。计算公式如下：

$$CER = \frac{N_{ins} + N_{del} + N_{sub}}{N}$$

其中 N_{ins}，N_{del} 和 N_{sub} 分别表示识别文本与标注文本之间发生插入、删除和替换操作的次数，则表示标注文本中所包含的文本标记的总数。

4.2 平均尖峰延迟（平均出字延迟）

平均尖峰延迟（Average Peak Latency, APL）是统计的每个解码正确的概率尖峰的首帧与通过强制对齐方法获得的每个文本标签人声范围的尾帧之间的时间差的平均值。这一指标反映了系统的平均延迟水平。平均尖峰延迟越低则意味着出字延迟越低，ASR 识别系统反馈识别结果的速度越快。

4.3 PR50/PR90

由于真实的 CTC 预测的出字延迟分布具有长尾分布的特点，所以引入了 尖峰延迟的 50 分位数和 90 分位数来衡量延迟分布的特点。其计算方法是根据每句话的平均出字延迟进行从小到大进行排序，以整个分部中第 50% 和 90% 条句子的平均出字延迟作为指标。PR50/PR90 越低表示整个长尾分布的尾巴越短，长尾分布中尾巴部分的数据延迟越低，比例越小。

5. 实验与分析

5.1 实验与模型搭建

本文基于开源中文语音识别数据集 AISHELL-1 进行实验，并采用了流式和非流式两种模型进行比较验证。两个模型均为 Transformer 模型结构，包含两层 2D 卷积构建的前端特征处理模块，以及 12 层 Transformer 编码层构建的编码模块以及一个输出线性映射层。其中流式模型依赖 510ms 的声学下文。

5.2 出字延迟比较

Table 1. Comparison of different models and different PER weights λ. APL denotes the average peak latency.

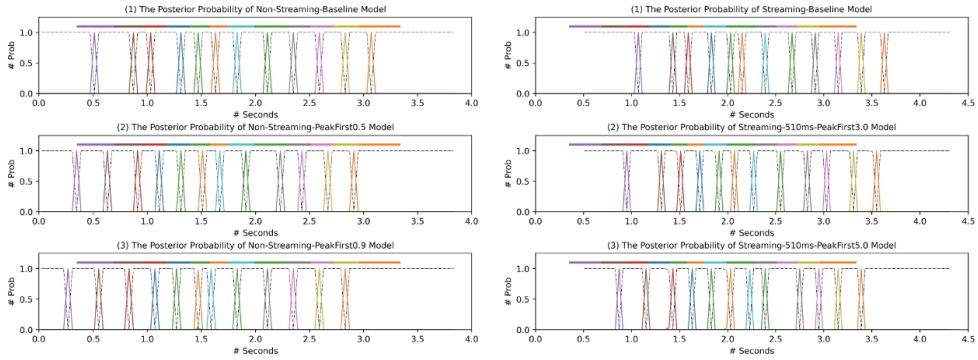
Model	Weight λ	Dev				Test			
		CER	APL(ms)	PR50(ms)	PR90(ms)	CER	APL(ms)	PR50(ms)	PR90(ms)
Non-Streaming Models	Baseline	5.38	-187.24	-300	-230	5.85	-202.44	-300	-220
	0.1	5.28	-214.82	-330	-250	5.71	-230.01	-330	-240
	0.3	5.36	-275.45	-390	-310	5.79	-289.88	-390	-300
	0.5	5.41	-337.35	-460	-370	5.83	-351.48	-450	-350
	0.7	5.43	-355.61	-470	-380	5.97	-369.58	-470	-360
	0.9	5.59	-403.33	-510	-420	6.00	-419.29	-510	-400
Streaming Models (510ms)	Baseline	6.43	400.23	280	360	7.03	384.95	280	360
	0.5	6.25	361.16	240	320	6.86	346.09	240	330
	1.0	6.16	343.39	220	300	6.81	328.44	220	310
	2.0	6.17	326.23	210	280	6.85	311.62	210	290
	3.0	6.21	297.10	180	260	6.84	283.22	180	270
	5.0	6.52	217.81	90	190	7.22	206.44	90	200

实验结果

上图中分别展示了流式识别模型与非流式识别模型在开源测试集上的字准确率和延迟结果。从实验结果很容易发现，无论是非流式模型还是流式模型，采用本文提出的尖峰优先正则化方法均能够降低各种延迟指标，并且延迟的变化值与权重的设置关系密切。通过调节权重设置可以显著改变延迟的大小，权重设置越大，延迟越低。在字错误率（CER）不降低的条件下，非流式模型和流式模型在测试集中分别可以获得 149 毫秒和 101 毫秒的延迟降低，在 CER 波动的可接受范围内，延迟甚至可以降低 200 毫秒以上。

当权重设置比较小的时候，模型甚至能够同时获得 CER 和平均延迟的降低，我们猜想造成这种现象的原因是正则化使得模型在学习邻近帧的时候同时学到了更长的声学下文信息。随着权重的变大，模型的识别错误率可能出现上升，此时权重的设置已经破坏了两个损失之间的平衡关系，模型在训练过程中会更激进地倾向于选择低延迟的路径，这种情况下会损失更多的声学下文信息，造成识别结果的衰退。

5.3 可视化分析



(a) Comparison of Different Non-Streaming Models (b) Comparison of Different Streaming Models

实验结果

本文随后通过可视化的方式对系统的延迟变化进行分析。上图中左侧三幅图表示非流式模型的输出概率分布，右侧三幅图则表示流式模型的输出概率分布。图上方的色块与概率尖峰一一对应，便是每个标记的发声范围，而下面的概率尖峰则表示 CTC 预测到对应标记的位置与概率。

很容易发现图中非流式模型中原本每个尖峰的位置就处于其对应的发声范围中，引入尖峰优先策略后尖峰的位置甚至能够提前其发声范围。而流式模型的概率尖峰也往往滞后于其发声范围，而引入尖峰优先策略后同样可以获得较大的延迟降低效果。通过图中不同权重的参数设置也可以发现，使用较大的权重能够更大程度地降低系统延迟。

6. 总结与展望

本文通过对 CTC 的输出概率分布进行分析，将 CTC 的出字延迟问题转化为一个知识蒸馏过程。通过知识蒸馏方法将 CTC 的输出概率分布沿着时间轴左移，从而有效地降低 CTC 模型的出字延迟。本文提出的方法简单有效，不需要强制对齐标注信息，也不需要复杂的损失和梯度计算方法。此外该方法也具有一定的扩展空间，或许可以扩展到 Transducer 等语音识别模型上。

7. 本文作者

正坤、鸿雨、李敏、飞飞、丁科、广鲁等，均来自美团平台 / 语音交互部。

8. 参考文献

[1] Alex Graves, Santiago Fern Landez, Faustino Gomez, and J Nurgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 369 - 376.

[2] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., “Deep speech 2: End-to-end speech recognition in english and mandarin,” in International conference on machine learning, 2016, pp. 173 - 182.

[3] Andrew Senior, Has.im Sak, F Lelix de Chaumont Quitry, Tara Sainath, and Kanishka Rao, “Acoustic modelling with cd-ctcsmbr lstm rnns,” in 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015, pp. 604 - 609.

[4] Jiahui Yu, Chung-Cheng Chiu, Bo Li, Shuo-yiin Chang, Tara N Sainath, Yanzhang He, Arun Narayanan,Wei Han, Anmol Gulati, Yonghui Wu, et al., “Fastemit: Low-latency streaming asr with sequence-level emission regularization,” in ICASSP 2021- 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 6004 - 6008.

[5] Yusuke Shinohara and Shinji Watanabe, “Minimum latency training of sequence transducers for streaming end-to-end speech recognition,” in Proc. Interspeech 2022, 2022, pp.2098 - 2102.

[6] Jinchuan Tian, Brian Yan, Jianwei Yu, Chao Weng, Dong Yu, and Shinji Watanabe, “Bayes risk ctc: Controllable ctc alignment in sequence-to-sequence tasks,” arXiv preprint arXiv:2210.07499, 2022.

[7] Jaeyoung Kim, Han Lu, Anshuman Tripathi, Qian Zhang, and Hasim Sak, “Reducing streaming asr model delay with self alignment,” arXiv preprint arXiv:2105.05005, 2021.

[8] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., “Espnet: End-to-end speech processing toolkit,” arXiv preprint arXiv:1804.00015, 2018.

ICDE 2023 | 多场景多任务学习在美团到店餐饮推荐的实践

作者：周杰 先帅 文豪 薄琳 张琨

随着推荐算法技术的不断发展，跨场景学习已经受到了越来越多的研究人员的关注。美团到餐算法团队受到业界相关技术的启发，不断探索到店餐饮多场景推荐的优化问题，在多场景多任务学习的推荐领域中积累了较多的应用经验。团队使用到店餐饮全域推荐场景数据训练统一的多场景多任务学习模型，减少了重复性开发，并在多个到店餐饮推荐场景进行落地，取得了较为显著的效果。

本文详细阐述了美团到店餐饮业务中多场景多任务学习的解决方案，基于该方案形成的学术论文《HiNet: Novel Multi-Scenario & Multi-Task Learning with Hierarchical Information Extraction》已经被国际数据工程会议 ICDE 2023 收录。

1. 背景

随着网络信息和服务的爆炸式增长，推荐系统已经成为为用户提供高质量个性化决策建议和体验的关键组件。传统的推荐系统，模型服务通常需要为特定场景单独进行定制化的开发，以适配不同场景下数据分布和特征空间的差异。然而在美团等工业互联网平台中通常存在多种多样的推荐场景（例如首页信息流、垂类子频道等）作用于用户访问的决策链路，同时基于每个场景的个性化推荐模型再对展示项目进行排序最终呈现给用户。

在美团到店餐饮（以下简称到餐）平台中，伴随业务精细化的发展趋势，越来越多的场景需要对推荐系统进行定制化的建设，以满足用户到店就餐的个性化需求。如下图 1 所示，现实中用户往往会在多个不同场景之间进行浏览、点击，并最终成交。

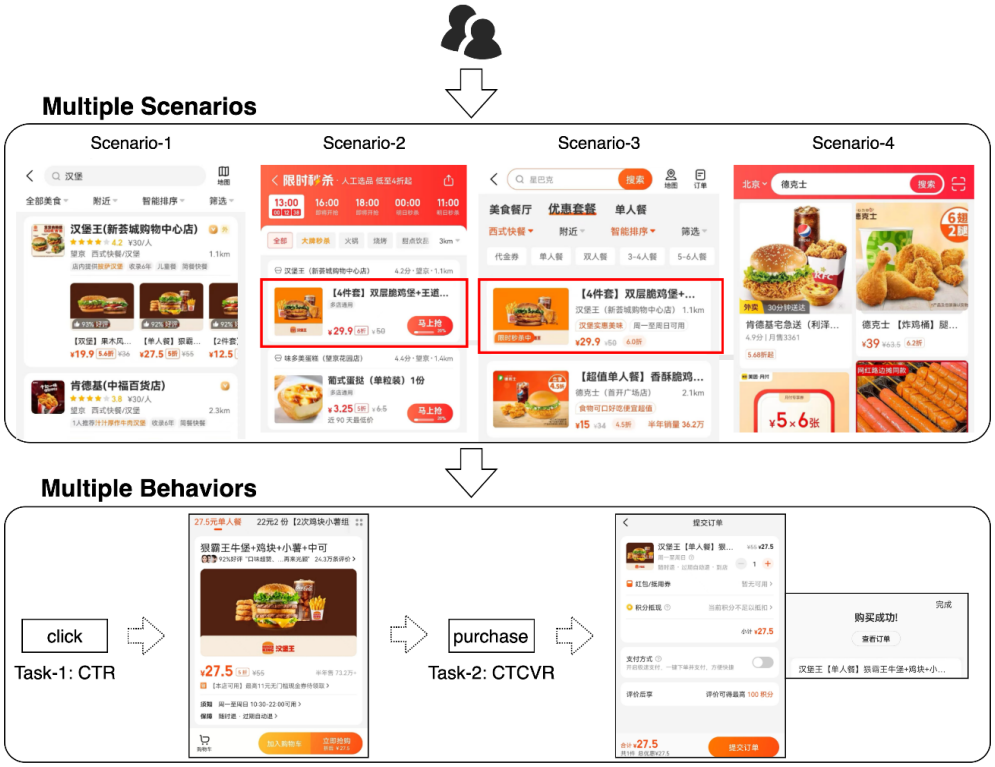


图 1　用户在美团到餐平台中的交互流程

但随着推荐场景数量的增加，传统地针对单个场景独立开发推荐模型，往往会导致如下问题：

1. 仅根据单场景自身的数据进行建模，无法利用到用户在跨场景中丰富的行为信息，忽视了场景共性信息，特别是考虑到多种场景中可能会存在重复展示的商品（在上图 1 中，红色矩形框圈中的其实是相同的商品）。
2. 一些长尾的业务场景由于流量较小且用户行为较为稀疏，数据量不足以让模型有效地进行建模。
3. 由于每个场景的特征挖掘、模型训练和上线部署是独立开发且相互隔离的，这会大大增加计算成本和维护负担。

总的来讲，推荐算法对各场景单独建模存在诸多的局限性。然而，简单地将多个场景数据集进行合并训练一个排序模型来提供服务，并不能有效地捕获到每个场景的特有信息。

此外，除了多场景推荐问题，每个场景中的用户满意度和参与度等通常都存在不同的衡量指标需要共同优化，例如点击率（CTR）和点击转化率（CTCVR）。因此需要开发一个有效和统一的框架，来解决这种在多个场景中优化各种指标复杂性的问题（即多场景多任务优化问题）。

在最近的一些研究中，相关方法往往是将多场景推荐做为一个多任务学习（Multi-Task Learning，MTL）问题进行建模，并且此类方法大多使用多门控混合专家（Multi-gate Mixture-of-Experts，MMoE）网络框架作为模型改进的基础来学习场景之间的共性和特性。然而，这种基于 MTL 的方法往往将多个场景的数据信息投影到同一个特征空间进行优化，这很难充分捕捉到具有多个任务的众多场景之间的复杂关系，因此也无法进一步提升多场景多任务学习模型的性能。

从直觉来看，多场景和多任务的信息建模应属于不同层次的优化，应该进行分层处理。因此，在本文中，我们提出了一种**层次化信息抽取网络**（Hierarchical information extraction Network，HiNet）。具体来说，我们设计了一个端到端的两层信息抽取框架，来共同建模场景间和任务间的信息共享和协作。

首先，在场景抽取层（Scenario Extraction Layer），HiNet 能够通过单独的专家模块提取场景共享信息和场景特有信息。为了进一步加强对当前场景的表示学习，我们设计了场景感知注意力网络（Scenario-aware Attentive Network，SAN），显式学习其他场景对当前场景的信息表征贡献程度。

然后，在任务抽取层（Task Extraction Layer），利用自定义的由任务共享和任务特有专家网络组成的门控网络，有效地缓解了多任务学习中共享信息和任务特有信息之间的参数干扰。

通过在模型结构上分离场景层和任务层的信息提取，可以将不同场景下的多个任务明确划分到不同的特征空间中进行优化，从而有利于提升模型性能。

整个论文的主要创新点如下：

1. 我们提出了一种新颖的多场景多任务学习模型 HiNet，用于优化多场景下的多种任务指标，其中创新性地应用了分层信息抽取架构。
2. 在场景信息抽取层中，我们提出了场景感知注意力网络 SAN 模块，进一步增强

了场景信息建模的能力。

3. 离线评估和在线 A/B 测试的实验证明了 HiNet 优于当前主要方法。目前，HiNet 已在美团到餐中的两个场景中进行了全面部署。

2. 层次化信息抽取网络

2.1 问题定义

如上所述，我们主要关注的是多场景多任务推荐的优化问题。我们将该问题定义为：$\hat{y}_i^j = f_i^j(x, s_i)$，其中 s_i 表示第 i 个场景指示，$\hat{y}_i^j$ 是第 i 个场景下任务 j 的预估值，xx 表示输入的稠密特征。

原始的特征输入中主要包括了用户画像特征、用户行为特征、当前场景特有特征和商品特征，其中的数值特征首先被转化为分类特征，然后将所有分类特征映射到低维向量空间来获得 x。考虑到美团到餐平台中具体的优化目标，我们分别为每个场景设置了 CTR 和 CTCVR 两个任务。

2.2 方法介绍

本小节将展开介绍层次化信息抽取网络模型 HiNet。如下图 2-(A) 所示，HiNet 模型中主要包括场景抽取层和任务抽取层两个核心模块。其中在场景抽取层主要包括了场景共享专家（Scenario-shared expert）模块、当前场景特有专家（Scenario-specific expert）模块以及场景感知注意力网络，通过这三部分的信息抽取，最终形成了场景层次的信息表征；而在任务抽取层中，我们使用自定义门控网络模块 CGC（Customized Gate Control）来对当前场景的多任务学习进行建模。下文将详细介绍上述提到的 HiNet 模型的关键部分。

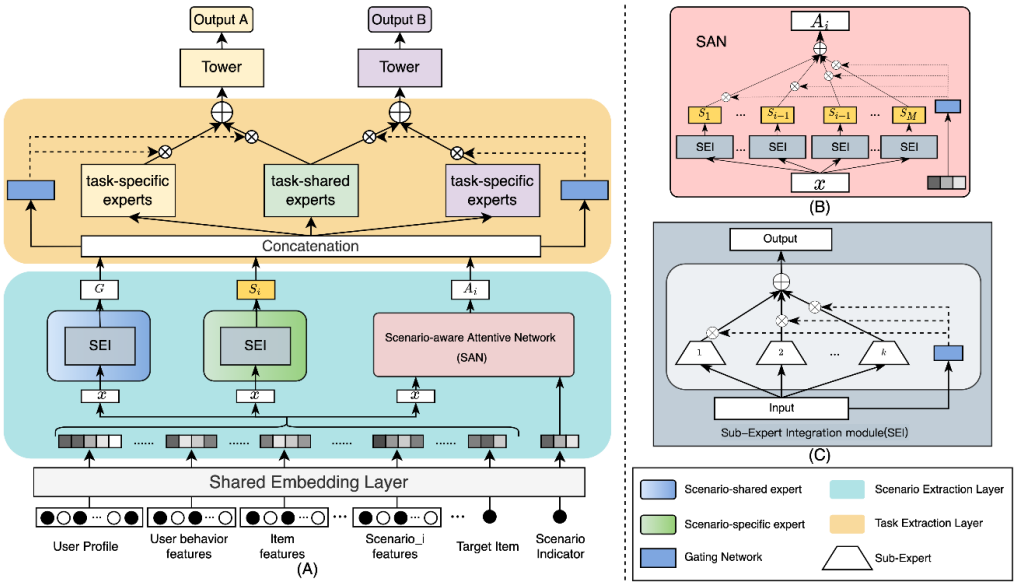


图 2　Hierarchical information extraction Network 模型结构

2.2.1 场景抽取层（Scenario Extraction Layer）

场景抽取层的作用是提取场景特有的信息表征和场景间共享的有价值信息表征，这是提高任务层次信息表征能力的基础。在场景抽取层，主要包括场景共享专家网络、场景特有专家网络和场景感知注意力网络三部分，下文将依次进行介绍。

1. 场景共享 / 特有专家网络

考虑到用户跨场景的穿插式行为以及多个场景间商品重叠的现象，到餐业务中多个场景的数据之间存在着有价值的共享信息。因此在策略上，我们设计了场景共享专家网络。这里受到混合专家网络架构 MoE（Mixture of Expert）的影响，场景共享专家网络是通过使用子专家集成模块 SEI（Sub-Expert Integration module，如图 2-©）生成的。

具体来说，场景共享专家网络的最终输出为 G，其公式为：

$$G = \sum_{k=1}^{K_s} g_{sh}^{k}(x) Q_{sh}^{k}(x)$$

其中，Q_{sh}^k表示第k个子专家网络，该网络是由多层感知器（Multilayer perceptron，MLP）和激活函数组成，K_s表示子专家网络$Q_{sh}(\cdot)$的数量，$g_{sh}(x)$表示门控网络的输出，它是通过带有Softmax激活函数的简单线性变换得到：

$$g_{sh}(x) = softmax(W_{sh}x)$$

除了使用场景共享专家网络提取不同场景间的共享信息，我们还为每个场景分别设计了场景特有专家网络来学习场景特有的信息，该网络也是由SEI模块组成。具体地，第i个场景的场景特有专家网络的输出S_i表示如下：

$$S_i = \sum_{k=1}^{K_i} g_{sp}^k(x) Q_{sp}^k(x)$$

其中Q_{sp}^k表示第k个子专家网络，K_i是$Q_{sp}(\cdot)$的数量，$g_{sp}(x)$表示场景特有专家网络所对应的门控网络的输出。

场景感知注意力网络

如上文所述，不同场景之间存在一定程度的相关性，因此来自其他场景的信息也可以对当前场景的信息表征做出贡献，从而增强当前场景的信息表达能力。考虑到不同场景间对彼此的表征能力贡献不同，我们设计了场景感知注意力网络（Scenario-aware Attentive Network，SAN）来衡量其他场景信息对当前场景信息表征贡献的重要性。具体来说，SAN 包含两部分输入：

a. 场景指示s_i生成的嵌入向量$Emb(s_i)$，该向量通过带有Softmax函数的门控网络后可以计算出其他场景对当前场景信息表征的重要性权重。

b. $S = [S_1, \cdots, S_{i-1}, S_{i+1}, \cdots, S_M]$，$S$对应了一系列其他场景生成的信息表示。

通过SAN模块计算的第i个场景的输出A_i是场景表示S的加权和：

$$A_i = \sum_{m \neq i}^{M} g_a^i(s_i) S_m$$

$$g_a^i(s_i) = softmax(W_a^i Emb(s_i))$$

其中$Emb(\cdot)$表示场景指示投影为嵌入向量，$g_a^i(\cdot)$表示基于权重$W_a^i \in R^{(M-1)\times d}$的门控网络，$d$表示$Emb(\cdot)$的维数，$M$是场景的个数。

通过SAN模块可以根据复杂的场景相关性不同程度地传递跨场景信息，有效地增强了场景的表示学习，从而提高了HiNet模型的性能。

综上所述，场景抽取层的整体输出C_i可以表示为：

$$C_i = Concat[G, S_i, A_i]$$

2.2.2 任务抽取层（Task Extraction Layer）

为了解决多任务学习中的负迁移问题，在任务抽取层，我们受到 PLE（Progressive Layered Extraction）模型的启发，采用了自定义门控网络 CGC 模块。

自定义门控网络

自定义门控网络主要由两部分组成：任务共享专家网络和任务特有专家网络。前者主要负责学习当前场景中所有任务中的共享信息，后者用于提取当前场景中各个任务的特有信息。

类似地，该结构通过门控网络计算所有专家网络的加权和作为输出。此外，为了避免不同场景之间的任务互相干扰，场景抽取层中第i个场景的输出C_i将被输入到特定场景的CGC模块中。最终，第i场景中任务j的塔单元输入T_i^j如下：

$$T_i^j = \delta_i^j(C_i)\left[E_{th}^i(C_i)||E_{tp}^{ij}(C_i)\right]$$

其中，$E_{th}^i(C_i)$和$E_{tp}^{ij}(C_i)$分别表示在第i个场景中任务共享专家网络和任务j的特有专家网络的集合。$\delta_i^j(C_i)$是一个门控网络，通过线性变换层和Softmax层来计算第i场景的任务j的权重向量：

$$\delta_i^j(C_i) = softmax(W_i^j C_i)$$

其中，$W_i^j \in R^{(m_i+n_i^j)\times d'}$是参数矩阵，$m_i$和$n_i^j$分别是$E_{th}^i(C_i)$和$E_{tp}^{ij}(C_i)$的维度，$d'$是$C_i$的维数。

最后，在第i个场景下任务j的预测值可表示为：

$$\hat{y}_i^j = \tau_i^j(T_i^j)$$

其中，$\tau_i^j(\cdot)$表示第i个场景下任务j的塔单元，由具有激活函数的MLP组成。

2.3 训练目标

我们提出的 HiNet 的最终损失函数是：

$$Loss = \sum_{i=1}^{M}\sum_{j=1}^{N_i} \lambda_i^j \cdot Loss_i^j(y_i^j,\ \hat{y}_i^j)$$

其中，N_i表示第个i场景下的任务数，λ_i^j是控制不同损失比例的超参数。

3. 实验

3.1 实验设置

- **数据收集**：我们收集了美团到餐平台中的六个场景（场景编号为 a 到 f）的用户日志数据作为我们的多场景多任务训练和评估数据集，其中场景 a 和 b 是大场景数据集。相比之下，c 到 f 被作为小场景数据集。

	All	Scenario-*a*	Scenario-*b*	Scenario-*c*	Scenario-*d*	Scenario-*e*	Scenario-*f*
#Exposure	23.6M	11.5M	6.3M	1.5M	1.7M	89K	1.7M
#Click	3.5M	1.4M	1.4M	204K	234K	3.6K	188K
#Order	715K	302K	375K	9.3K	9.1K	0.28K	19K
#CTR	15.38%	12.56%	22.50%	14%	13.84%	4.12%	11.17%
#CTCVR	3.15%	2.64%	5.91%	0.64%	0.54%	0.31%	1.13%

表 1　各场景样本数据集统计

1. **评估指标**：我们分别考虑每个场景的 CTR 和 CTCVR 任务的性能评估，并采用 AUC（Area Under ROC Curve）作为多场景多任务数据集的评估指标。
2. **模型对比**：为了公平地对比我们提出的 HiNet 模型与业界的 SOTA（State-Of-The-Art）模型，我们使用相同条件的实验环境和模型参数量，并对每个模型进行了充分调优和多次实验。具体对比的模型如下：

多任务学习模型：

- Shared Bottom：该模型是一个具有硬参数共享的神经网络模型。
- MMoE：该方法使用灵活的门控网络调整专家网络表示信息，并最终使用塔单元融合每个任务的所有专家网络表示信息。
- PLE：该模型基于 MMoE，将专家网络显式划分为任务共享专家和任务特有专家，有效缓解了“跷跷板”现象带来的负迁移问题。

多场景学习模型：

- HMoE：该方法由 MMoE 改进而来，对多个场景的预测值进行建模，并针对当前场景优化任务预测结果。
- STAR：该方法通过星型拓扑结构构造一个共享的和场景特有的网络，用于学习当前场景的信息表征。

需要指出的是，上述用于对比的模型最初只是为了单纯解决多任务学习或多场景学习的问题而提出的。为了实现公平的实验比较，我们在实验中对相关对比模型做了自适应扩展，以满足多场景多任务建模的需求。

3.2 性能比较

Model Type	Model	Scenario-*a*		Scenario-*b*		Scenario-*c*		Scenario-*d*		Scenario-*e*		Scenario-*f*	
		auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr
Multi-task learning	Shared Bottom	0.7631	0.8426	0.7863	0.7883	0.6962	0.7307	0.6192	0.6131	0.6108	0.6079	0.7847	0.8088
	MMoE	0.7876	0.8581	0.7955	0.7978	0.7098	0.7532	0.6702	0.6881	0.6203	0.6106	0.8096	0.8337
	PLE	0.7741	0.8600	0.7936	0.7852	0.7108	0.7516	0.6794	0.7053	0.6169	0.6108	0.8075	0.8295
Multi-scenario learning	HMoE	0.7873	0.8556	0.7947	0.8014	0.7015	0.7531	0.6754	0.7174	0.6088	0.6183	0.8073	0.8311
	STAR	0.7716	0.8572	0.7906	0.8138	0.7040	0.7499	0.6786	0.7144	0.6163	0.6107	0.8051	0.8305
ours	HiNet	**0.7904**	**0.8675**	**0.7996**	**0.8194**	**0.7146**	**0.7620**	**0.6858**	**0.7282**	**0.6253**	**0.6203**	**0.8115**	**0.8359**

表 2　在所有场景下相关对比模型的性能比较

表 2 展示的是我们在美团到餐平台中六个场景下的各个模型的性能对比。从结果可以看出，我们提出的 HiNet 模型在所有场景的 CTR 和 CTCVR 任务指标上都优于其他对比模型，这证明了 HiNet 在多场景多任务建模的优势。

3.3 消融研究

为了研究 HiNet 模型中每个关键组件的效果，我们设计了两个 HiNet 模型的变体用于消融分析。具体如下：

- HiNet（w/o hierarchy）：表示去掉了信息抽取的层级架构，直接采用 CGC 网络进行多场景多任务学习建模。
- HiNet（w/o SAN）：表示的是在场景抽取层中删除 SAN 模块后的 HiNet 模型。

Model	Scenario-*a*		Scenario-*b*		Scenario-*c*		Scenario-*d*		Scenario-*e*		Scenario-*f*	
	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr	auc_ctr	auc_ctcvr
HiNet(w/o hierarchy)	0.7737	0.8574	0.7836	0.8028	0.7108	0.7503	0.6753	0.6981	0.6151	0.6122	0.7861	0.8170
HiNet(w/o SAN)	0.7880	0.8579	0.7947	0.8148	0.7114	0.7545	0.6817	0.7252	0.6197	0.6198	0.8045	0.8287
HiNet	**0.7904**	**0.8675**	**0.7996**	**0.8194**	**0.7146**	**0.7620**	**0.6858**	**0.7282**	**0.6253**	**0.6203**	**0.8115**	**0.8359**

表 3　HiNet 模型的消融实验结果对比

从表 3 的实验结果，我们可以观察到，变体模型 HiNet（w/o hierarchy）在所有指标上都有严重的性能下降，这表明分层信息抽取架构可以有效地捕获跨场景的共性和差异，从而提高模型的性能。类似地，在场景抽取层去掉 SAN 模块后，变体模型 HiNet（w/o SAN）在多个场景下性能也有明显的下降，这表明 SAN 模块学习到的权重信息可以有效地增强场景抽取层的信息表征能力。

3.4 在线 A/B 测试

为了进一步验证我们提出的 HiNet 模型的在线性能，我们在美团到餐平台中的场景 a 和 b 中部署了 HiNet 模型，并与基线模型进行了为期一个月的在线 A/B 测试。

model	Scenario-a			Scenario-b		
	CTR	CTCVR	Order Gain	CTR	CTCVR	Order Gain
baseline	-	-	-	-	-	-
HiNet(ours)	↑ 0.24%	↑ 3.38%	↑ 2.85%	↑ 0.11%	↑ 1.47%	↑ 1.75%

表 4　场景 a 和 b 的在线 A/B 测试收益

从表 4 可以看出，HiNet 模型在多个场景的 CTR 和 CTCVR 指标中均超过了基线模型，并且在订单增益上有显著的提升，这进一步说明了我们提出的 HiNet 模型的有效性。目前，HiNet 模型已经全面部署在上述两个业务中，并为业务的增长做出了一定的贡献。

4. 总结与展望

多场景多任务建模是目前推荐系统中最关键和最具挑战性的问题之一。以往的模型主要通过将所有信息投影到同一个特征空间来优化不同场景下的多个任务，这导致模型性能存在不足。

在本文中，我们提出了层次化信息抽取网络 HiNet 模型，它利用分层优化架构对多场景多任务问题进行建模。在此基础上，我们在场景抽取层设计了场景感知注意力网络模块 SAN 来增强场景的表示学习能力。离线和在线 A/B 测试实验都验证了 HiNet 模型的优越性。

值得一提的是，目前业界已经出现了大量的图神经网络在推荐模型上的应用。受此启发，在未来的工作中，美团到餐算法团队将图神经网络的信息传递能力结合到多场景多任务学习建模的方案中，继续实践我们的方法，并进一步设计更加完善的模型，来解决在美团到餐平台中存在的复杂的多场景多任务建模问题。

作者简介

周杰、先帅、文豪、薄琳、张琨等，均来自美团到店 / 平台技术部。

参考文献

[1] P. Li, R. Li, Q. Da, A.-X. Zeng, and L. Zhang, “Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space,” in Proceedings of the 29th ACM International Conference on * Information & Knowledge Management (CIKM),

2020, pp. 2605 – 2612.
[2] X.-R. Sheng, L. Zhao, G. Zhou, X. Ding, B. Dai, Q. Luo, S. Yang, J. Lv, C. Zhang, H. Deng et al., "One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction," in Proceedings of the 30th * ACM International Conference on Information & Knowledge Management (CIKM), 2021, pp. 4104 – 4113.
[3] 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 Proceedings of the 24th ACM SIGKDD international conference on * knowledge discovery & data mining (SIGKDD), 2018, pp. 1930 – 1939.
[4] H. Tang, J. Liu, M. Zhao, and X. Gong, "Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations," in Proceedings of the 14th ACM Conference on Recommender Systems (RecSys), 2020, pp. 269 – 278.
[5] L. Torrey and J. Shavlik, "Transfer learning," in Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, 2010, pp. 242 – 264.
[6] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345 – 1359, 2010.
[7] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu, "Cross-domain recommendation: challenges, progress, and prospects," in 30th International Joint Conference on Artificial Intelligence (IJCAI). International Joint * Conferences on Artificial Intelligence, 2021, pp. 4721 – 4728.
[8] Y. Zhang and Q. Yang, "A survey on multi-task learning," IEEE Transactions on Knowledge and Data Engineering, 2021.
[9] S. Ruder, "An overview of multi-task learning in deep neural networks," arXiv preprint arXiv:1706.05098, 2017.
[10] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in Thirty-second Conference on Neural Information Processing Systems (NeurIPS), 2018.
[11] C. Rosenbaum, T. Klinger, and M. Riemer, "Routing networks: Adaptive selection of non-linear functions for multi-task learning," in International Conference on Learning Representations (ICLR), 2018.
[12] J. Wang, S. C. Hoi, P. Zhao, and Z.-Y. Liu, "Online multi-task collaborative filtering for on-the-fly recommender systems," in Proceedings of the 7th ACM conference on Recommender systems (RecSys), 2013, pp. 237 – 244.
[13] R. Caruana, "Multitask learning," Machine learning, vol. 28, no. 1, pp. 41 – 75, 1997.
[14] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," Journal of Big data, vol. 3, no. 1, pp. 1 – 40, 2016.
[15] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," arXiv preprint arXiv:1701.06538, 2017.
[16] D. Eigen, M. Ranzato, and I. Sutskever, "Learning factored representations in a deep mixture of experts," Computer Science, 2013.
[17] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," Neural computation, vol. 6, no. 2, pp. 181 – 214, 1994.
[18] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," Neural computation, vol. 3, no. 1, pp. 79 – 87, 1991.

[19] S. E. Yuksel, J. N. Wilson, and P. D. Gader, “Twenty years of mixture of experts,” IEEE transactions on neural networks and learning systems, vol. 23, no. 8, pp. 1177–1193, 2012.

[20] Y. Zhang, C. Li, I. W. Tsang, H. Xu, L. Duan, H. Yin, W. Li, and J. Shao, “Diverse preference augmentation with multiple domains for cold-start recommendations,” in IEEE International Conference on Data Engineering (ICDE), 2022.

KDD 2023 | 美团技术团队精选论文解读

作者：美团科研合作

ACM SIGKDD（国际数据挖掘与知识发现大会，简称 KDD）于 1995 年首次举办，自此成为数据挖掘领域的重要国际会议之一，也是研究数据挖掘和知识发现这一领域的学者和企业人士分享研究成果、讨论前沿话题、推动技术进步和发展的重要平台。ACM SIGKDD 的宗旨是促进数据挖掘的应用和理论研究，支持和推动先进技术和方法的开发和创新，并推广数据挖掘和知识发现领域的教育、培训和普及。该组织致力于提高数据挖掘和知识发现的社会价值，推动其在各个领域的应用和发展。

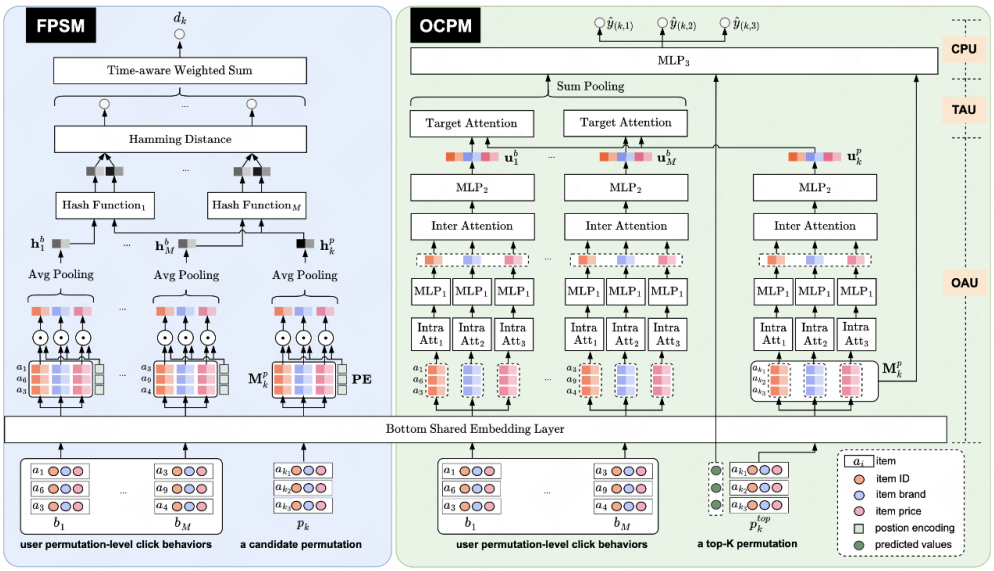


01《PIER: Permutation-Level Interest-Based End-to-End Re-ranking Framework in E-commerce》

论文作者：晓文、杨凡（共同一作）、王泽、晓旭、国钢、永康、兴星、王栋等，均来自美团技术团队。

论文类型：长文

论文下载：PDF



论文简介：Feed 流推荐作为目前最主流的推荐载体，其推荐质量直接影响用户体验 / 商家效益 / 平台收入等多个核心指标。Feed 推荐特点是为每个用户的请求生成并展示多个 items，用户的点击行为会同时受到上文和下文影响。由于传统 point-CTR 预测是在展现 items 产生之前进行的，导致无法利用上下文信息，推荐质量受损。目前工业届更多考虑重排方案，然而由于排列的多样性，穷举会导致排列个数过多，实际很难落地。目前业界通常存在两种近似方案：

1. 只考虑位置偏置和上文信息，但忽略下文影响效果；
2. 同时考虑上下文，Listwise 预估后重新排序，该方案存在 Evaluation Before Reranking 的问题。

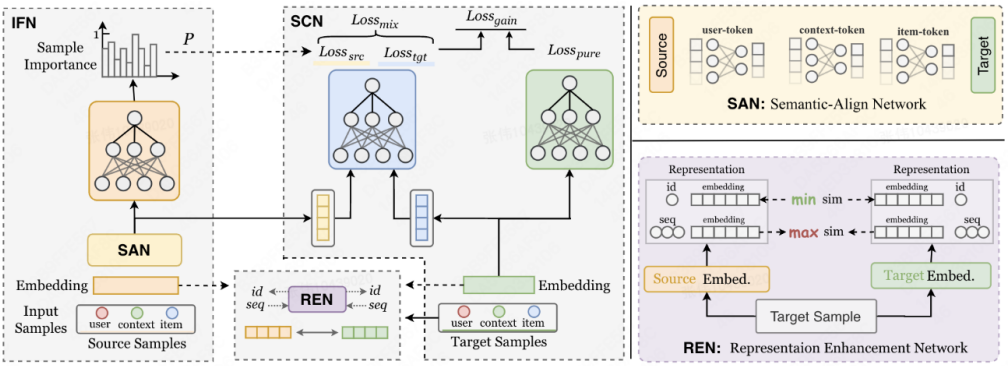
以上两个方案在效果上均有损，因此如何在重排阶段同时考虑上下文，并且降低复杂度兼顾可落地性，是一个非常重要的问题。针对以上问题，论文提出了一个名为 PIER 的端到端重新排序框架，该框架遵循两阶段范式并包含两个主要模块，分别称为 FPSM 和 OCPM。基于用户页面兴趣偏好，在 FPSM 中应用 SimHash 从全排列中快速的生成候选列表集合，降低了落地复杂度；然后在 OCPM 中设计了一种新颖的全向注意力建模机制，以更好地捕获列表中的上下文信息；最后，通过引入对比学习损失以端到端的方式联合训练这两个模块，使用 OCPM 的预测值来指导 FPSM 生成更好的列表。离线实验结果表明，PIER 在公开和工业数据集上均优于基线模型，目前已经部署到美团外卖广告场景，取得了较为显著的成果。

02《A Collaborative Transfer Learning Framework for Cross-domain Recommendation》

论文作者：张伟、鹏业、张波、兴星、王栋等，均来自美团技术团队。

论文类型：长文

论文下载：PDF



论文简介：在推荐系统中，为了满足用户多样化的兴趣和需求，催生出多种业务领域（Domain）。每个领域的点击率（CTR）可能会有较大差别，这导致了对不同业务领域的 CTR 预测建模需求。为了提升目标域效果，越来越多的学者尝试综合利用多域数据改善模型表现，也被称为知识迁移。这其中主要有 2 种迁移学习的范式：1) 预训练 + 微调：源域的最优解对于目标域可能是局部最优，导致负迁移（Negative Transfer）；2) 混合样本 + 多任务训练：不同域之间数量和特征存在差异，梯度容易被数据量大的域主导（Seesaw Effect）。

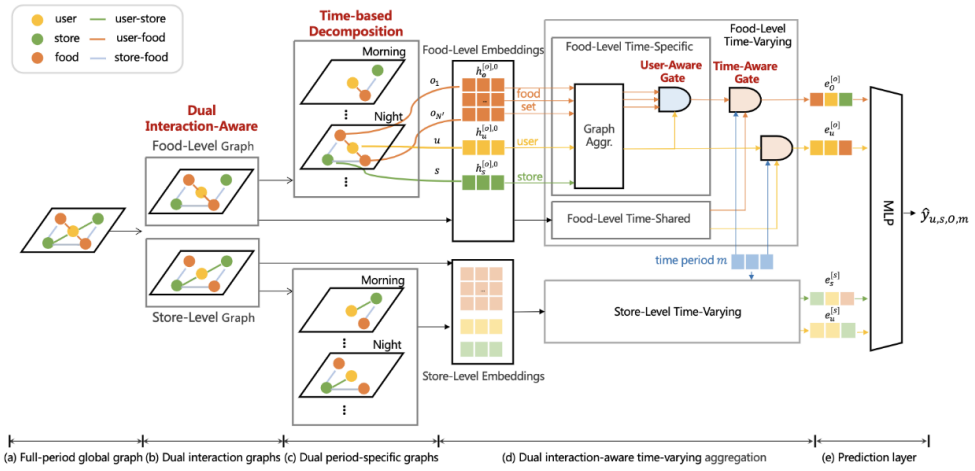
针对以上问题，该论文提出了一种协同跨域迁移学习框架（简称 CCTL）来解决这些问题，通过协同训练的方式，评估源域样本对目标域的信息增益，以此作为反馈计算源域样本权重，从而使目标域获得更多有益的迁移知识。框架整体上分为两个模块：IFN 模块用于输出源域样本表征及权重，SCN 模块用于主模型训练及源域增益计算及反馈。同时，通过 REN 模块使用对比学习强化不同域的差异性信息，使用 SAN 模块对齐源域目标域在多个 Token 的表征。CCTL 被部署在美团外卖展示广告场景，并取得较为明显的收益。

03《Modeling Dual Period-Varying Preferences for Takeaway Recommendation》

论文作者：玉婷（美团）、吴贻清（中国科学院计算技术研究所）、朱勇椿（中国科学院计算技术研究所）、庄福振（北京航空航天大学）、瑞东（美团）、北海（美团）、展博（美团）、安竹林（中国科学院计算技术研究所）、徐勇军（中国科学院计算技术研究所）

论文类型：长文

论文地址：PDF



论文简介：外卖推荐系统旨在为用户提供含有符合用户兴趣偏好的菜品的商家，已经服务于数十亿用户。与传统推荐不同，外卖推荐面临两大挑战：

1. 双重交互感知偏好。传统推荐往往关注单一层次的用户对商品的偏好，而外卖推荐则需要综合考虑用户对商家和菜品的双重偏好。
2. 一天内的用户偏好变化。传统的推荐通常从会话级别或天级别的角度对用户偏好的连续变化进行建模。然而，在实际的外卖系统中，一天中的早、中、晚、深夜时段，用户的偏好差异显著。

为了应对这些挑战，该论文提出了一种针对于外卖推荐场景的双重时段变化偏好建模模型 (DPVP)。 具体而言，论文设计了双重交互感知（Dual Interaction-Aware）模块，旨在根据用户与商店和食品的交互来捕捉用户的双重偏好。此外，为了模拟用户一天中

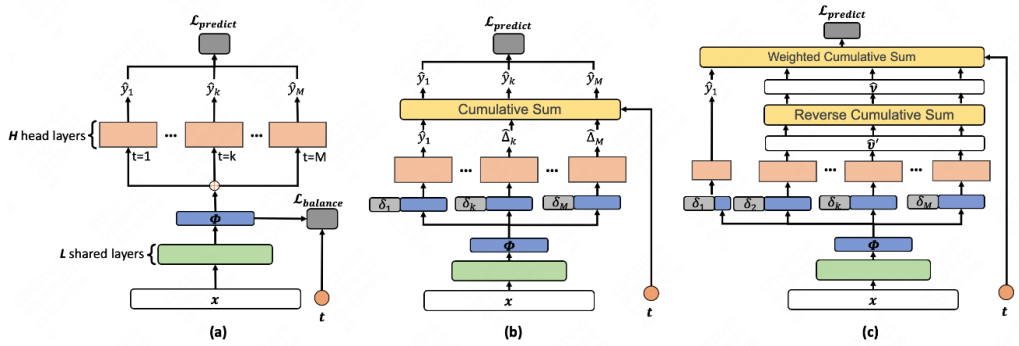
不同时段变化的偏好，论文提出了基于时间的分解（Time-based Decomposition）模块以及时间感知门控机制（Time-Aware Gate）。离线和在线实验均表明，论文模型在现实世界的数据集上优于最先进的方法，并且能够捕捉用户的双重时段变化的偏好。此外，该模型已经在美团外卖平台上线，取得了较为明显的成绩。

04《A Multi-stage Framework for Online Bonus Allocation Based on Constrained User Intent Detection》

论文作者：王超、晓巍、徐帅、王哲、志强、冯岩、游安、陈彧等，均来自美团技术团队。

论文类型：长文

论文地址：PDF



论文简介：奖励分配是营销领域的一个重要问题，通常可以建模为一个背包优化问题，主流的解决方案通常是两阶段的方法。 第一阶段是用户意图检测，预测用户在每个奖励下的转化概率。在第二阶段求解运筹优化问题，得到最优分配策略。 然而，该解决方案通常面临以下挑战：

1. 在用户意图检测阶段，由于交互和噪声的稀疏性，传统的 uplift 估计方法缺乏可解释性，常出现违背边际效用为正的领域知识；
2. 两个阶段之间存在最优性差距，限制了第二阶段所获得最优值的上界；
3. 由于线上环境的变化，实际奖励成本消耗经常违反给定的预算限额。

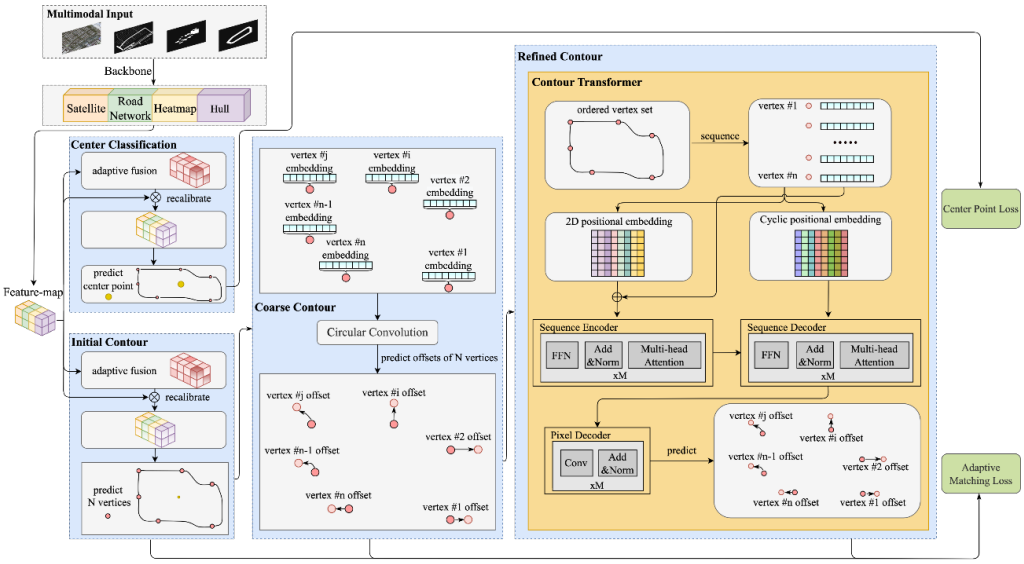
为了解决上述挑战，论文提出了一个由三个模块组成的框架，即用户意图检测模块、在线分配模块和反馈控制模块。对于第一个挑战，在用户意图检测模块中，我们基于深度表示学习对效用增量进行隐式建模，提出单调约束结构增强模型解释性。为了减小两阶段最优性差距，进一步提出了凸约束模型来提高最优值的上限，以解决第二个挑战。针对第三个挑战，为了应对在线成本消耗地波动，论文利用框架中的反馈控制策略，使实际成本更准确地接近给定的预算限制。基于真实的美团支付数据评估以及线上 A/B 实验均证明了该方案的优势和有效性。

05《C-AOI: Contour-based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform》

论文作者：奕达、丽影、大平、水平、方潇、井华、仁清、致钊等，均来自美团技术团队。

论文类型：长文

论文下载：PDF



论文简介： AOI 是配送业务数字化运营的基础设施，也是关键决策算法的基础单元。现有的 AOI 自动生成相关研究主要依赖于三种方式：1）预定义的形状；2）基于密度的聚

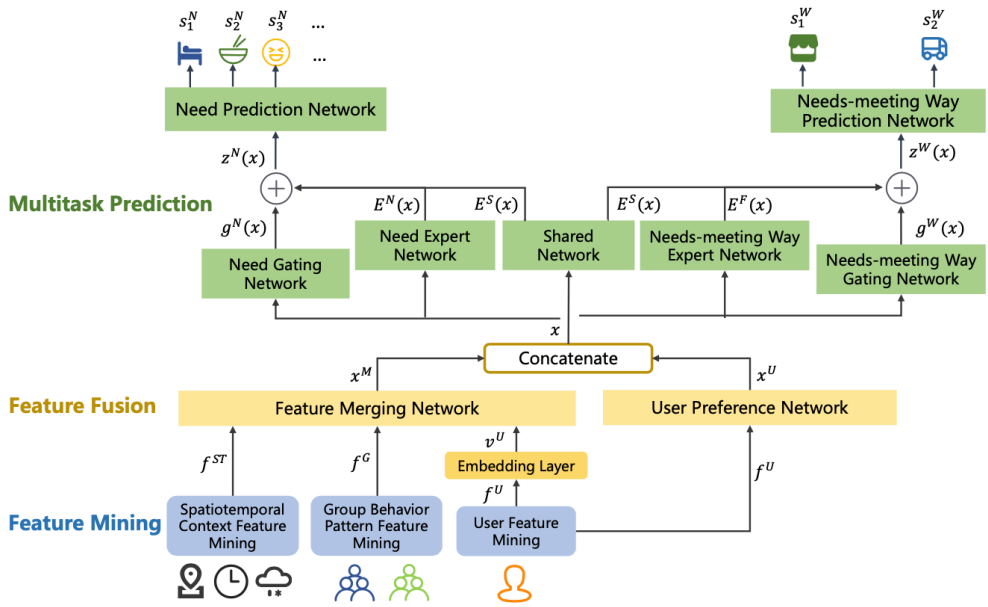
类；3）需要多个模型进行后处理的语义分割方法。上述研究方法降低了 AOI 轮廓的生成质量上限从而限制了自动生成 AOI 轮廓的落地。为了解决上述问题，该论文将其构建为基于实例分割的轮廓回归问题并提出了一种端到端的 AOI 轮廓自动生成框架 C-AOI。在 C-AOI 中，提出了 Contour Transformer、循环位置编码以及自适应匹配损失优化顶点回归路径。该模型在美团配送业务数据集上进行了实验，大幅提升 AOI 边界生成质量的同时具备较快的推理速度。此外，该模型已投产并应用于美团履约场景。

06《NEON: Living Needs Prediction System in Meituan》

论文作者：兰孝翀（清华大学）、高宸（清华大学）、诗琪（美团）、秀琦（美团）、影格（美团）、张晗（美团）、华周（美团）、恒亮（美团）、李勇（清华大学）

论文类型：长文

论文地址：PDF



论文简介：用户意图包含人们日常生活中的各类需求，如吃、住、娱乐等。美团作为连

接用户和服务提供者的生活服务平台，用户意图预测问题往往被定义为预测用户当前的业务需求。准确理解用户意图对于生活服务平台而言至关重要，理解结果能够广泛应用于平台推荐系统、精准化营销等多种业务场景。然而，当前平台在用户意图预测问题上的探索并不是很多，且面临如下两个严峻挑战：

1. 用户需求，经常受人物、地点、时间、自然环境等多种复杂因素的影响；
2. 用户意图存在多样化特点，并非用户的所有需求都会在美团 App 上体现。

为了应对上述两个挑战，本论文设计了一个名为 NEON 的用户意图预测系统，其主要包括三个阶段：特征挖掘、特征融合和多任务预测。

1. 在特征挖掘阶段，基于时、空、人、环境等维度进行用户消费场景定义，并重点提取了不同场景下的群体行为特征用于建模，通过人群共性行为辅助进行系统冷启优化，有效缓解了数据稀疏问题。
2. 在特征融合阶段，构造了融合用户个体及所属群体行为特征的网络结构，该网络可以有效学习不同场景下的用户意图变化。
3. 在多任务预测阶段，除了预测用户业务意图之外，引入辅助任务，用来预测用户履约方式偏好（送货到家、到店消费），以此来增强模型对时、空上下文的学习。

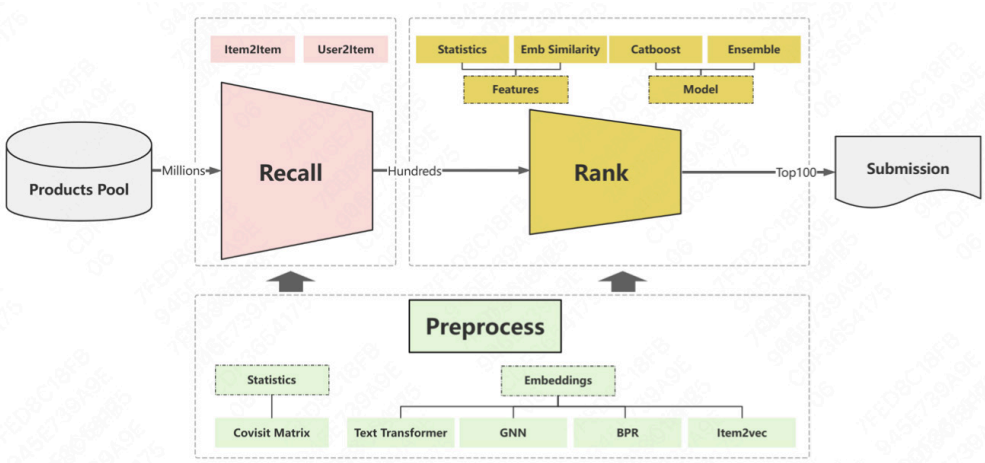
论文中的 NEON 系统，已部署至美团的推荐、营销、运营算法引擎中，大规模在线 A/B 测试评估实验表明，NEON 在上述多个应用场景下均可带来业务效果的显著提升，实验结果证明了 NEON 系统在细粒度用户需求刻画、履约方式偏好识别、潜在需求预测等方面的有效性。

07《A Hybrid Approach of Statistics and Embeddings for Multilingual and Multi-Locale Recommendation》

论文作者：伟佳（美团）、詹金（DataRobot）、黄钟山（Freelance）、王露（微软）、王强（美团）

论文类型：Workshop

论文下载：PDF



论文简介：为了鼓励多语言推荐系统的发展，亚马逊发布了一个多语言和多地区购物会话数据集，并基于该数据集举办了 KDD Cup 2023 多语言多地区会话推荐挑战赛。本论文介绍了美团在此次比赛中的解决方案。按照推荐系统中广泛使用的设定，美团的解决方案包括两个阶段：召回和排序。在第一阶段，利用各种召回策略检索一组候选 Item，包括基于共同访问矩阵的 I2I 召回、基于 GraphEmbedding 的 I2I 召回、基于 TextTransformer 的 I2I 搜索和基于 BPR 的 U2I 召回。在第二阶段，训练了一个模型来预测每个用户与候选产品互动的概率。该模型是两个 GBDT 模型的集成，包括丰富的统计特征和 Embedding 相似度特征。该团队在在 Track2 中获得了第 3 名，在 Track1 中获得了第 4 名。

美团科研合作

美团科研合作致力于搭建美团技术团队与高校、科研机构、智库的合作桥梁和平台，依托美团丰富的业务场景、数据资源和真实的产业问题，开放创新，汇聚向上的力量，围绕机器人、人工智能、大数据、物联网、无人驾驶、运筹优化等领域，共同探索前沿科技和产业焦点宏观问题，促进产学研合作交流和成果转化，推动优秀人才培养。面向未来，我们期待能与更多高校和科研院所的老师和同学们进行合作。欢迎老师和同学们发送邮件至：meituan.oi@meituan.com。

CVPR 2023 | 美团技术团队精选论文解读

作者：美团技术团队

写在前面

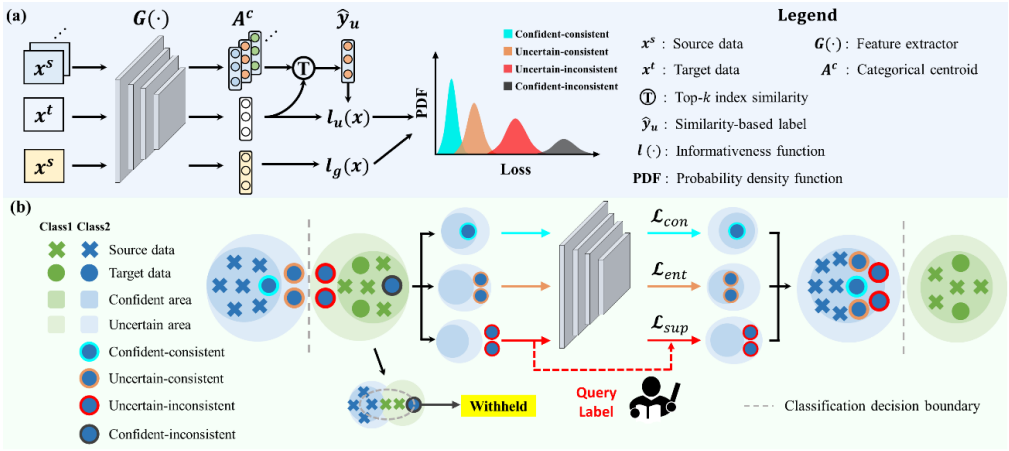
CVPR 全称为 IEEE Conference on Computer Vision and Pattern Recognition，国际计算机视觉与模式识别会议。该会议始于 1983 年，与 ICCV 和 ECCV 并称计算机视觉方向的三大顶级会议。根据谷歌学术公布的 2022 年最新学术期刊和会议影响力排名，CVPR 在所有学术刊物中位居第 4，仅次于 Nature、NEJM 和 Science。2023 年，CVPR 共收到全球 9155 篇论文投稿，最终 2360 篇被接收，接收率约为 25.78%。



| 01 Divide and Adapt: Active Domain Adaptation via Customized Learning

论文作者：黄铎峻（中山大学，美团实习生），李继昌（香港大学），陈伟凯（腾讯 – 美国），黄君实（美团），柴振华（美团），李冠彬（中山大学）

论文下载：PDF



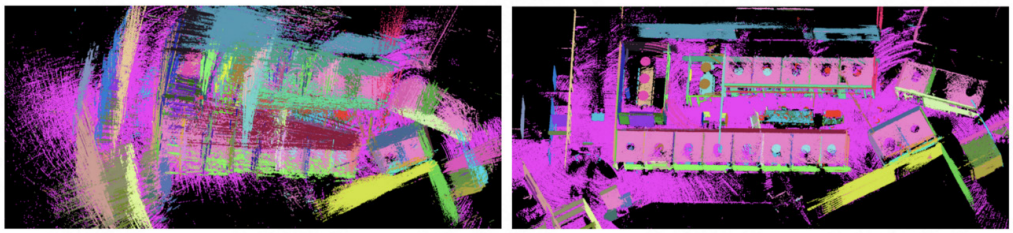
论文简介：该论文入选焦点论文（Highlight Paper）。近年来，主动领域自适应被提出用于在领域自适应问题中，设计主动学习算法在未标注的目标域数据中选择最有信息量的一小批样本进行标注，来最大化的提升深度学习模型在目标域数据上的性能，达到高效的标注效率。在实际的开放场景中，目标域样本相对于源域数据的可区分度各异，即对于源域上初始化的而言具有不同层次的可迁移性。目前，鲜有工作对可迁移性各异的目标域样本分类讨论，设计高价值样本的采样策略和定制化的领域自适应训练策略。

我们提出了一种分而治之的策略，综合考虑目标域样本与源域数据的差异性，和模型预测的不确定性，来设计主动学习的采样策略，并提出定制化的目标函数来约束具有不同可迁移性的样本子集，进一步实现采样的鲁棒性。实验表明，我们实现了在多种领域自适应场景下的最优性能，包括无监督领域自适应（UDA）、半监督领域自适应（SSDA）和无源域领域自适应（SFDA）等等。

| 02 Efficient Second-Order Plane Adjustment

论文作者：周力普（美团）

论文下载：PDF



(a) Point cloud from perturbed poses (b) Point cloud from our algorithm

论文简介：该论文入选焦点论文（Hightlight Paper），推导出了平面优化问题 Hessian 矩阵的闭式解，由此提出了高效的平面优化问题的二阶优化算法。该算法可以广泛应用于深度传感器的三维高精度重建。

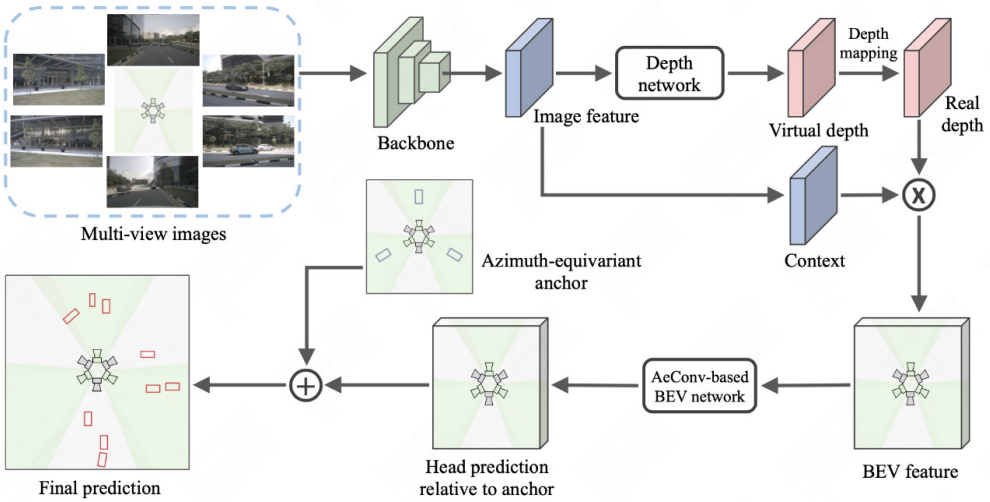
平面通常用于深度传感器的 3D 重建，例如 RGB-D 相机和 LiDAR。本文研究估计最佳平面和传感器位姿的问题。这由此产生的最小二乘问题在文献中被称为平面调整（PA）。迭代方法常被用来解决这些最小二乘问题。通常，牛顿法很少用于大规模最小二乘问题，因为 Hessian 矩阵的计算时间复杂度很高。相反，迭代算法通常采用使用 Hessian 矩阵的近似值，例如 Levenberg Marquardt（LM）方法。本文采用牛顿法有效地解决了 PA 问题。

具体来说，给定姿势，最佳平面有一个闭式解。因此，我们可以从中消除平面参数，它显著减少了变量的数量。此外，由于最佳平面是姿势的函数，这种方法实际上确保了每次迭代都能产生最佳平面，这有利于收敛。其难点在于如何高效计算 Hessian 矩阵以及由此产生的梯度向量。本文提供一个高效的解决方案。实证结果表明，我们的算法优于目前的 SOTA 算法。

| 03 AeDet: Azimuth-invariant Multi-view 3D Object Detection

论文作者：冯承健（美团），揭泽群（美团），钟毓杰（美团），初祥祥（美团），马林（美团）

论文下载：PDF



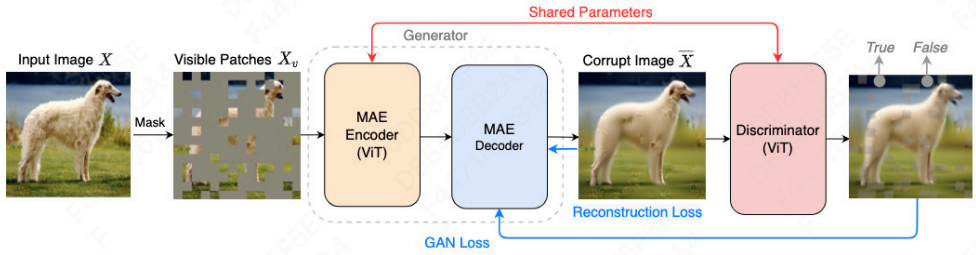
论文简介：近年来，基于鸟瞰图的多视图 3D 目标检测技术在 Brid-Eye-View（BEV）空间通过卷积网络检测物体，取得了巨大的进展。然而，传统的卷积忽略了 BEV 特征的径向对称性，增加了检测器优化的难度。为了保持 BEV 特征的固有性质并便于模型优化，我们提出了方位角等变卷积（AeConv）和方位角等变锚。方位角等变卷积的采样网格始终是径向的，因此可以学习到方位不变的 BEV 特征。而方位角等变锚使得检测头能够学习预测与方位无关的目标。

此外，我们还引入了一种相机解耦的虚拟深度，以统一具有不同相机内参的图像的深度预测。由此产生的目标检测器被称为方位等变检测器（AeDet）。我们在 nuScenes 数据集上进行了多视图 3D 目标检测实验：方位等变检测器取得 62.0% NDS，显著超越了现有的多视图 3D 目标检测方法。

| 04 Masked Auto-Encoders Meet Generative Adversarial Networks and Beyond

论文作者：费政聪（美团），范铭源（美团），朱理（美团），黄君实（美团），魏晓明（美团），魏晓林（美团）

论文下载：PDF



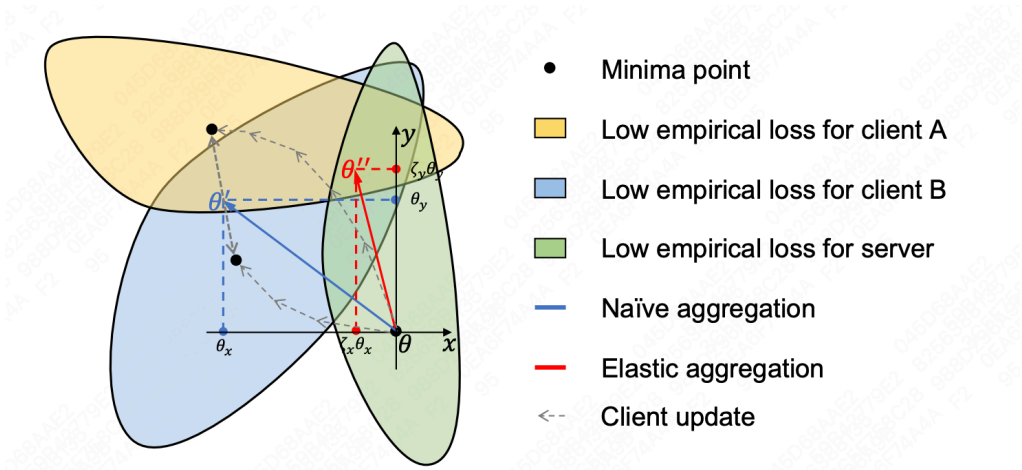
论文简介：掩码自动编码器（Mask autoencoder，MAE）预训练方法通过随机掩码图像块（patch），然后训练视觉模型基于未掩码的图像块来重建原始图像的像素。尽管它们在下游视觉任务中表现出很好的性能，但它们通常需要大量的计算才能生效。在本文中，我们介绍了一种基于生成对抗网络（Generative Adversarial Networks，GAN）的预训练框架。

具体来说，将生成器预测的图像和原始掩盖的图像进行拼接，并使用判别器来预测每一个拼接的图像块是否被替换。实验表明，由于对完整图像进行输入建模和反馈，我们提出的 MAE-GAN 框架比原始的 MAE 像素重建效果更好。同时，通过共享主网络参数，在相同的模型大小、数据和计算量下，我们的方法学习的视觉表征能力显著优于 MAE 方法。特别地，我们在 ImageNet-1k 上预训练了 200 epoch 的 ViT-B 模型在下游图像分类结果上优于训练了 1600 epoch 的 ViT-B MAE 基准。

| 05 Elastic Aggregation for Federated Optimization

论文作者：陈登盛（美团），胡杰（美团），Vince Junkai Tan，魏晓明（美团），吴恩华（中科院软件所）

论文下载：PDF



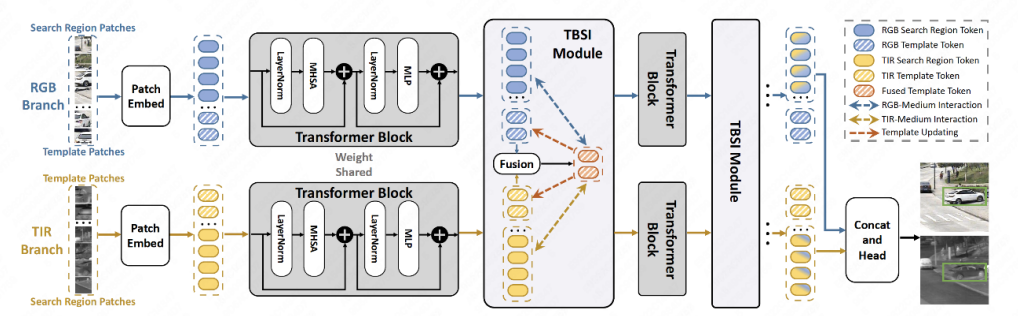
论文简介：在人工智能安全性上，联邦学习旨在保证数据隐私不遭到泄漏的前提下完成模型的协同训练。由于在不同终端设备的数据分布存在较大差异，导致全局共享模型在使用本地数据进行优化以后会偏向本地的数据分布状态。我们称这种现象为「客户端漂移（Client Drift）」。客户端漂移现象的存在，会导致全局共享模型在优化前期收敛缓慢，在优化后期无法达到更优解。

我们提出了弹性聚合（ Elastic Aggregation），一种新的参数更新方法用于缓解上述现象。弹性聚合首先利用每个终端设备上的无标签数据计算出对应模型参数对结果产生影响的敏感程度（Parameter Sensitivity），然后这个参数敏感程度来对全局共享模型进行加权聚合更新。弹性聚合是首个在联邦学习中充分利用无标签数据来提升模型性能的方法，并且能够非常容易地嵌入到现有的其它联邦学习优化算法中。实验表明，在联邦学习场景下弹性聚合方法可以显著提升视觉和文本理解任务的性能。

| 06 Bridging Search Region Interaction with Template for RGB-T Tracking

论文作者：惠天瑞（中国科学院信息工程研究所，美团实习生），荀子政（北京航空航天大学），彭风光（北京航空航天大学），黄君实（美团），魏晓明（美团），魏晓林（美团），戴娇（中科院信工所），韩冀中（中科院信工所），刘偲（北京航空航天大学）

论文下载：PDF



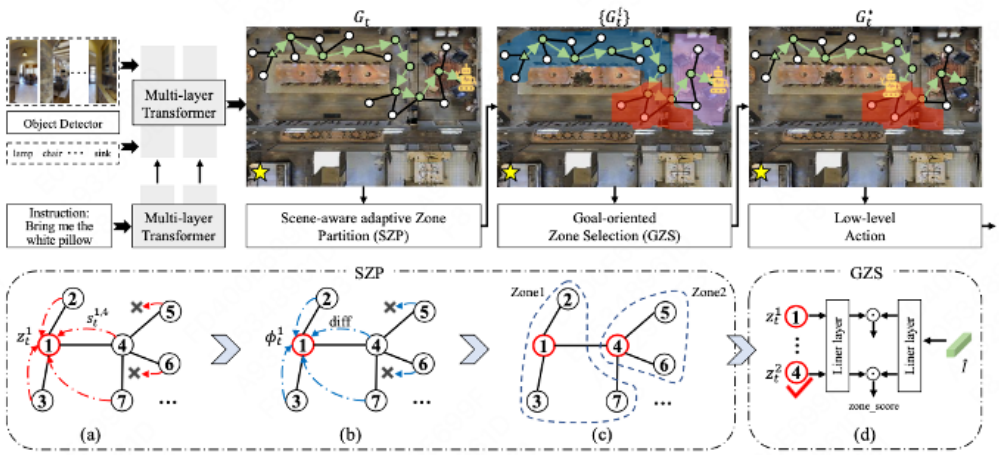
论文简介：RGB-T 跟踪旨在利用可见光（RGB）和热红外（TIR）模态的互补增强能力来改进不同场景下的单目标跟踪效果，其中有效的跨模态交互是方法设计的关键一环。先前的工作直接拼接 RGB 和 TIR 的搜索区域特征，或对孤立的 RGB 和 TIR 候选框对进行模态融合，导致冗余背景噪声的引入或局部区域内的上下文建模不足。

为了缓解上述局限性，我们提出了模板桥接搜索区域交互（TBSI）模块，该模块利用模板作为媒介，通过收集和分发目标相关的对象和环境上下文来桥接 RGB 和 TIR 搜索区域之间的跨模态交互。原始模板也会使用来自模板中介的丰富多模态上下文进行更新。我们将 TBSI 模块插入到 ViT 主干网络中，实现统一的特征提取、搜索区域 – 模板匹配和跨模态交互，在三个主流 RGB-T 跟踪数据集上取得了超越现有方法的优异性能。

| 07 Adaptive Zone-aware Hierarchical Planner for Vision-Language Navigation

论文作者：高晨（北京航空航天大学，美团实习生），彭兴宇（北京航空航天大学），严汨（北京大学），王鹤（北京大学），杨立荣（美团），任海兵（美团），李鸿升（香港中文大学），刘偲（北京航空航天大学）

论文下载：PDF



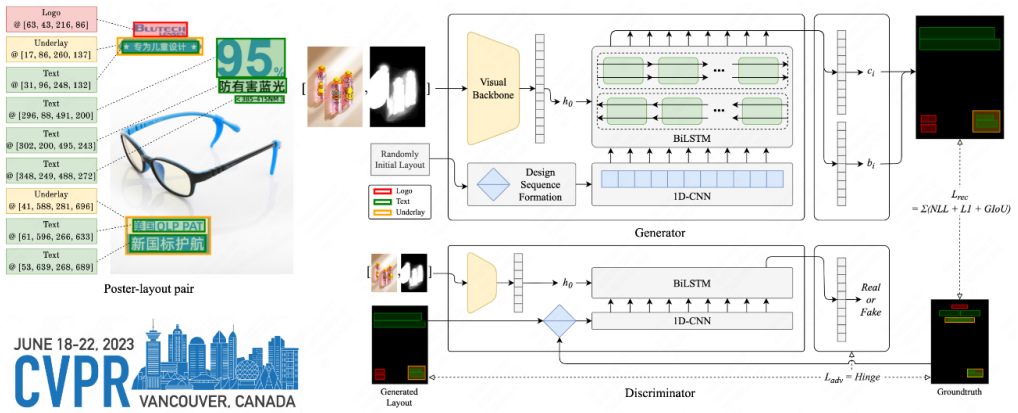
论文简介：本工作聚焦 Vision-Language Navigation（VLN）任务。在 agent 导航过程中，需要自适应地设置并实现一系列子目标。然而，先前的方法采用单步规划方案，即在每一步直接执行导航动作。在本工作中，我们提出了一种 Adaptive Zone-aware Hierarchical Planner（AZHP），明确地将导航过程分为两个异构的阶段，即通过分区 / 选区（High-Level Action）和子目标执行（Low-Level Action）以进行分层规划。

具体而言，AZHP 通过状态切换模块（SSM）异步地执行两级操作。对于 High-Level Action，我们提出了一种 Scene-Aware Adaptive Zone Partition（SZP）方法，以自适应地将整个导航区域划分为不同的子区域。通过目标区域选择（GZS）方法，为当前子目标选择合适的区域。对于 Low-Level Action，Agent 在所选区域执行多步骤导航决策。此外，我们提出 HRL 策略和辅助监督，以训练 AZHP 框架。实验证明了我们提出方法的优越性，在多个 VLN 数据集（REVERIE、SOON、R2R）上均达到了最优的性能。

| 08 PosterLayout: A New Benchmark and Approach for Content-Aware Visual-Textual Presentation Layout

论文作者：徐筱媛（北京大学，美团实习生），何相腾（北京大学），彭宇新（北京大学），孔浩（美团），张庆（美团）

论文下载：PDF



论文简介：图文展示的布局生成旨在给定的图像画布上自动安排元素（例如：广告文本、图标、衬底）的空间位置，用于广告设计中能够取代呆板的预定义模板。现有的布局生成工作忽略了图像画布与布局的交叉关系，导致两者难兼容。

为此，该论文首先从源域多样性、主题多样性以及布局复杂度三个方面切入，建立图文展示布局数据集和评测基准 PosterLayout，并提出设计序列生成网络（Design Sequence GAN），通过引入人类经验的设计序列形成（Design Sequence Formation）算法，将布局自动重组为隐含时间信息的设计序列，并以画布图像的视觉特征作为初始状态，模拟人的设计行为，自动生成与画布内容兼容的布局。实验结果验证了新基准和新方法的有效性，达到超越现有方法的性能。该算法在论文接收前已应用上线，目前已在美团 App 的外投首页广告制图等场景落地。

美团科研合作

美团科研合作致力于搭建美团技术团队与高校、科研机构、智库的合作桥梁和平台，依

托美团丰富的业务场景、数据资源和真实的产业问题，开放创新，汇聚向上的力量，围绕机器人、人工智能、大数据、物联网、无人驾驶、运筹优化等领域，共同探索前沿科技和产业焦点宏观问题，促进产学研合作交流和成果转化，推动优秀人才培养。面向未来，我们期待能与更多高校和科研院所的老师和同学们进行合作。欢迎老师和同学们发送邮件至：meituan.oi@meituan.com。

后台/系统

Code：美团代码托管平台的演进与实践

作者：潘陶 费翔 丹丹 毛强

1. 引言

Code 是美团自研的代码托管平台，其中包括了代码版本管理、分支管理及代码评审等功能，协同众多研发流程工具平台，支撑内部所有工程师的日常研发工作。经过近 3 年的建设，目前 Code 托管了数以万计的仓库，日常处理千万级的 Git 相关请求，稳定支撑着美团研发流程规范的持续落地。本文主要介绍美团在建设代码托管平台过程中面临的一些挑战和实践经验。

2. 美团代码托管平台建设之路

2.1 代码托管平台的发展史

回顾美团代码托管平台的发展史，整个历程可以划分为三个阶段：单机部署、多机部署以及自研分布式代码托管平台。

第一阶段：单机部署

美团最初的代码托管平台，和绝大多数 Web 系统一样，单机部署即可运行，所有用户的请求均通过 Web 应用进行响应。由于 Git 使用基于文件组织形式的存储模式，无论是通过页面访问还是执行 Git 命令操作，最终都会表现为磁盘的文件读写，高 IO 磁盘尤为重要。整体架构如下图 1 所示：

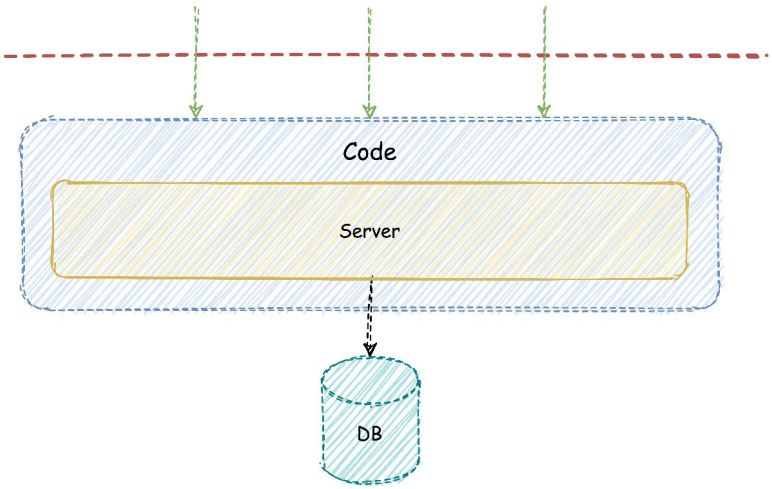


图 1　单机部署

第二阶段：多机部署

在访问规模不大的情况下，第一阶段这种单机架构可以满足日常的开发需求。但随着研发团队业务需求的不断增长，测试自动化流程的逐步完善，扩展性瓶颈也愈发明显，主要表现为以下 2 个方面：

- **存储：**由于公司资源限制和地域分配不均等因素，代码托管平台部署机器已配置最大容量的可用 SSD 磁盘，使用率仍高达 80%，可用空间严重不足。
- **负载：**随着研发人员的不断增多，在访问高峰期，CPU 和 IO 负载高达 95% 以上，页面出现严重的卡顿，仅能通过限流保障系统的持续服务。

因而，单机部署无法再承载高峰期的访问量，系统扩容刻不容缓。于是，我们开始设计了一套能够通过多机负载同一仓库 IO 的读写分离架构方案，以解决较为严重的 IO 负载问题。在读写分离架构中，最重要的是要保证用户视角的数据一致性（用户随时可以读取提交的最新代码），这里采取了以下措施：

1. 写操作仅发生在主节点。
2. 采用懒汉同步模式，在读取数据时触发从节点同步数据，若失败，则路由到主节点。
3. 采用独主兜底模式，遇遇到突发情况时可以迅速禁用从节点，保障数据安全。

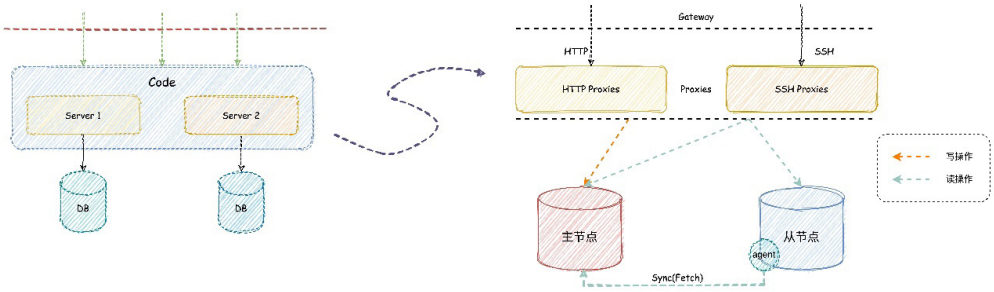


图 2　多机部署

如图 2 所示，我们将仓库访问形式按照应用层协议区分为 HTTP 和 SSH，分别由对应的解析代理模块进行读写分发操作后再下发到主从节点（此处采用了 Round-Bobin 的算法分发读请求），使得读吞吐量整体扩大了 2 倍。对于从节点，我们部署了 Agent，在用户发起读请求时会触发同步仓库数据的 Fetch 操作，以保证数据的一致性。

第三阶段：自研分布式代码托管平台

在第二阶段，虽然通过多机负载 IO 的读写分离架构短暂性地解决了扩展性瓶颈问题，但仓库数据仍在持续不断地指数增长。同时，除扩展性问题之外，可用性瓶颈也凸显出来，主要表现在以下 2 个方面：

- **运维：**无论是日常迭代更新版本还是热修复紧急 Bug，都需要停服才能部署系统，停服期间用户无法使用代码托管平台。
- **备份：**系统采用冷备份的方式多副本存储 Git 数据，无法保证核心数据的实时恢复，异常情况下存在数据丢失风险。因此，搭建具备高可用性和水平扩展性的分布式架构迫在眉睫。我们调研了业界主流代码托管平台的分布式方案，并结合公司内部的业务特性，最终选择了基于应用层分片的分布式架构，该架构满足了以下 2 个特性：
- **高可用：**采用三副本多活模式，规避代码丢失风险，且系统版本更新无需停服，单机断电、宕机均可正常提供服务。
- **水平扩展：**可通过扩容分片集群的方式进行存储和负载扩展，实现广义下的“无限”容量。

综上所述，Code 基于 GitLab 生态开源组件二次开发，并采用了应用层分片多活模式

的分布式架构方案，简介如下：

1. 底层存储服务基于 GitLab 生态开源组件二次开发，有良好的生态和丰富的功能支持。
2. 各服务间均通过 gRPC 进行交互通信，主要考虑点是 Git 大多数为二进制数据通信，gRPC 基于 HTTP 2.0，有良好的传输性能和流式支持。
3. 通过路由模块实现逻辑层与存储层有效隔离，逻辑层对物理分片无感知，存储层如同一个整体提供服务。
4. 采用了多活复制模式的数据保障架构，提高读写吞吐量，满足日均千万级的请求量需求。
5. 针对于应用层分片的劣势，在架构设计时也做了相应的针对性优化，具体如下：
   - 热点库：提供了自动化的分片迁移能力，在发现仓库出现热点时，可进行分片迁移达到分片均衡。
   - 跨分片数据交互：通过业务层的 Git 事务包装，我们使用共享 Object 的模式并确保相互关联的仓库均落在同一分片上，既避免了跨分片通信的问题，也减少了磁盘空间占用和访问时延。

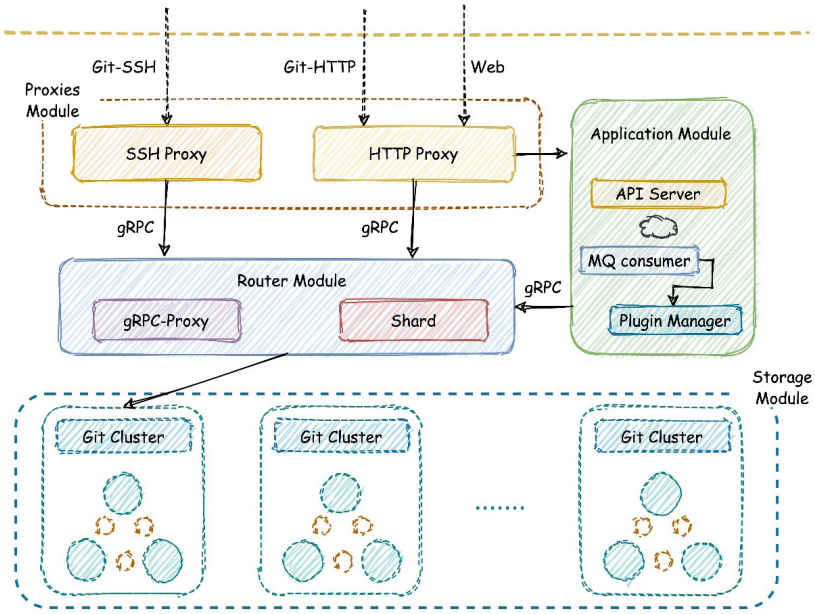


图 3　Code 系统架构图

3. 美团代码托管平台架构演进的落地和挑战

代码托管平台在架构演进过程中，最终完成了以下两个目标：

- **高可用：**缩短停机时间，提高可用性，系统稳定可靠。
- **高扩展：**针对计算和存储资源，可以实现水平扩展。

接下来，针对于每个目标，本文分别从技术挑战、方案选型、设计及解决方案等方面详细介绍我们的实践经验。

3.1 扩展性目标

3.1.1 技术挑战

在进行水平扩展改造时，主要面临了以下两类挑战：

- **规模性：**在研发流程自动化等背景下，美团代码托管平台需要具备千万级吞吐、低延迟及高可用的系统性能，以提高研发效率。
- **兼容性：**技术改造涉及的场景比较多，主要有两方面的考量：（1）用户低感知，新老系统保证现有通信方式及平台使用方式不变；（2）兼顾过渡时期底层存储介质多样性、运维体系兼容等问题，并保障上下游系统的正常运行。

3.1.2 方案选型

经过对主流代码托管平台（GitHub、GitLab、Bitbucket 等）的调研分析，我们发现各大平台主要采用了以下两种架构解决扩展性问题。

架构设计		
方案	优势	劣势
网络共享存储	• 存储"无限"扩展，无须担心存储问题 • 仅需改造存储层逻辑，接入成本不高	• 公司现有的网络共享存储系统（例如EBS、DFS）暂时无法满足很高的系统稳定性需求 • 共享缓存难以实现，无法通过缓存加速提供更好地访问性能
应用层分片	• 存储可以通过集群扩容达到水平扩展能力 • 大部分逻辑处于应用层，相比共享存储改造较多，但底层侵入性小	• 依旧会出现分片单点和热点问题，需要提供完备的分片迁移方案 • 在跨分片的数据交互场景下，请求会有一定的延迟

图 4　架构设计方案对比

通过上述对比可以发现，如果直接接入共享存储，暂时无法满足代码托管平台的稳定性和性能要求（若 Git 机制进行并行优化，且使用更高读写性能的分布式存储系统，或许是一个不错的选择）。在共享存储优化改造成本较高的前提下，我们最终采用了应用层分片的分布式架构，它既满足扩展性的要求，也更加成熟和稳定，并表现出不错的性能。

3.1.3 方案设计

我们通过代理模块实现了请求分发，通过路由模块实现了仓库分片，通过应用模块的无状态改造实现了弹性伸缩，从而达成了水平扩展的架构目标。下面将对这些模块进行详细的介绍。

代理模块

- **SSH Proxy：**提供 Git-SSH 操作代理，提供 Git-SSH 请求代理，通过路由模块获取路由信息，到目标机器执行 SSH 操作。SSH Proxy 组件基于 go-crypto 库开发，实现了公钥识别用户，流量控制，长连接超时处理，SSH 转 gRPC 等功能。后续计划引入 signature 校验，以应对不同的使用场景。
- **HTTP Proxy：**提供 Git-HTTP/Web 请求代理，通过路由模块存储的仓库分片映射关系，决策仓库路由节点。HTTP Proxy 基于 Go-Gin 开发，实现了请求甄别，流量控制，多层代理等功能。最初 HTTP Proxy 还被作为灰度迁移的核心组件，通过流量统一收口，支持请求分发到新老 Code 系统，以确保请求和数据的平滑迁移。

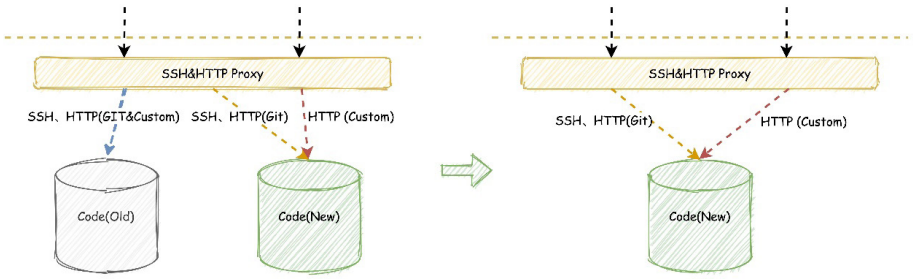


图 5　代理模块

路由模块

- **Shard：**记录仓库与其所在分片之间的映射关系，是应用层分片架构的“中枢系统”。Shard 服务除维护映射关系外，还是容灾模块必不可少的“决策者”，通过获取各个节点当前访问仓库的最新版本，从而决定读写路由。由于 Golang 出色的高并发表现，目前路由相关接口的平均响应时间在 15ms 以内。该模块的主要特性如下：
  - 建立仓库和分片的映射关系，为了避免由于仓库路径更新造成文件夹拷贝 / 移动等行为带来一定的复杂性，这里采用了仓库 ID 作为唯一标识。
  - 利用 Go Routine 获取节点的数据同步状态，并通过超时机制保障用户非无限时等待。
  - 使用 Key-Value Cache 存储仓库和分片的映射，以降低映射关系的请求时延。

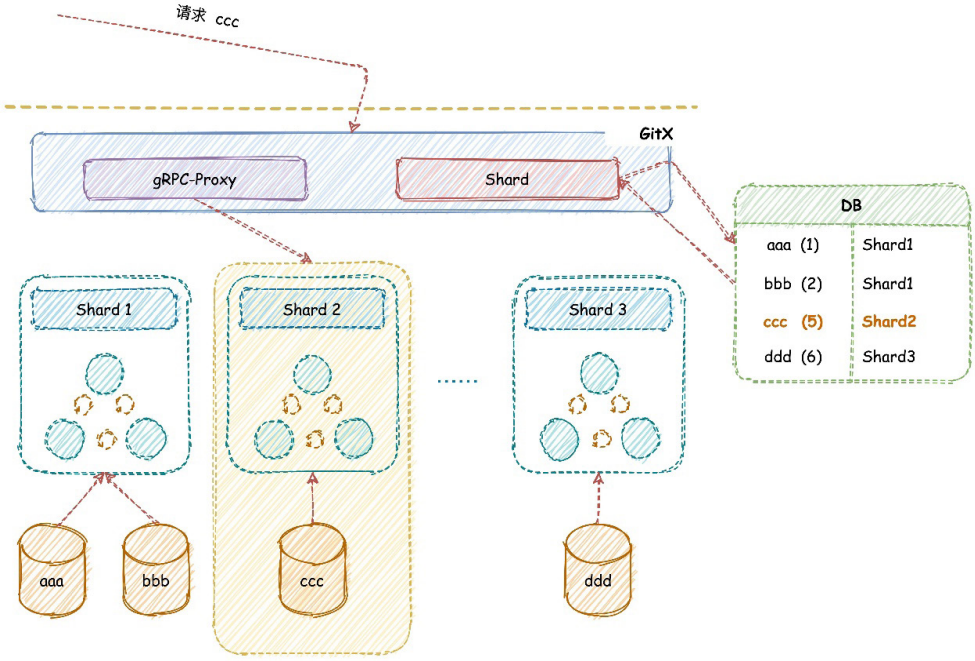


图 6　路由模块

应用模块

应用模块主要包括以下两点功能：

- 提供 Git 相关的业务逻辑接口处理代码内容信息、代码评审等复杂性业务请求。
- 监听代码和分支变更消息，发送事件通知打通第三方业务系统和收集度量信息。

整体模块架构如下图 7 所示：

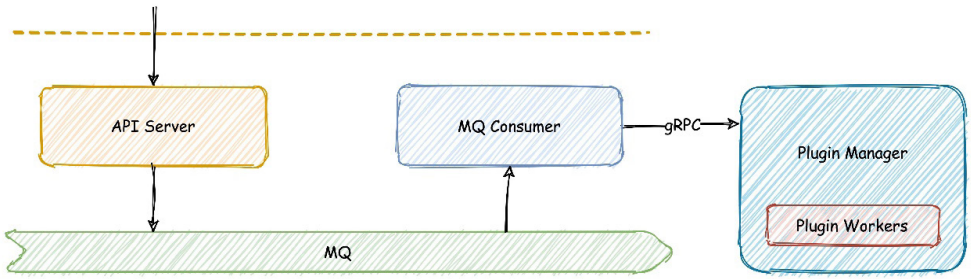


图 7　应用模块

3.1.4 解决思路

规模性解决思路

规模化的主要目标是：具备**支撑千万级请求的系统能力，并支持计算、存储等资源的水平扩展能力**，其中路由均衡是必不可少的一环。

a. 路由均衡

Code 系统对数据源可靠性要求较高，而对性能要求相对比较低，因而我们采用了**严格仲裁**的路由模式，具体的逻辑配置如下：

- **使用版本号判定哪个节点提供的代码内容最新：**版本号越大，代表数据越新，当版本最大时即为最新的数据。当 W+R > N 时总能读到最新的数据（N：总节点个数，W：判断写入成功需要的响应节点数，R：读取数据时至少要成功读取的个数），当 W 越小时，写入的可用性就越高，R 越小，读取的可用性就越高。我们选择了 N=3，R=W=2 的常规推荐配置，根据概率推算可达到 99.999% 的可用性水平。
- **采用读修复模式：**当读取数据时，若发现节点数据不一致，此时触发数据同步逻辑，以修复落后节点的数据。

该功能内置于路由模块的 Shard 服务，架构如下图 8 所示：

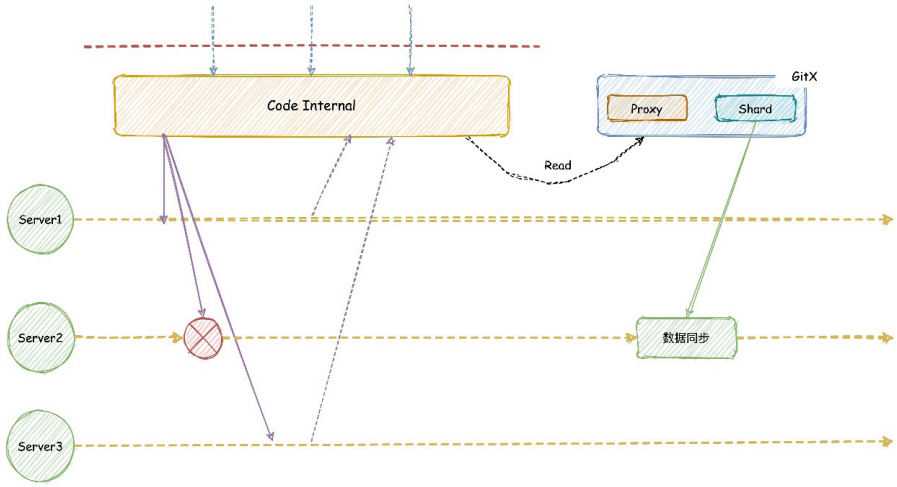


图 8　路由逻辑示意图

兼容性解决思路

兼容性目标总结为一句话就是：**业务使用无感知**。因此，我们主要从以下三个方面考虑兼容性。

a. 与各系统交互方式及现有基础设施兼容

Code 系统的众多下游系统（多套前端 UI、业务研发流程工具平台等）依赖系统提供的开放 API、Hook 机制等扩展功能，为了减少系统升级对业务方造成影响，需要保证系统交互方式兼容。同时还要确保系统运维监控体系正常运行，维持可监测状态，我们主要做了以下四件事情：

- **兼容核心功能：**使用频度高的功能平移到新系统，而使用中低频的功能，与业务沟通使用场景，再评估是否兼容。
- **重新设计部分功能：**提供更为合理的 WebHook 配置能力及崭新的代码评审功能。
- **边缘功能运营下线：**推进废弃和历史遗留功能的下线，并提供合理的替代方案。
- **打通运维体系：**保持现有监控埋点及运维接口接入方式，使系统处于可维护、可监测的状态。

b. 非分布式版本无缝切换到分布式版本

Code 系统仓库众多，需要有低成本的用户自主切换方式保障数据逐步迁移，我们主要做了以下三件事情：

- **可视化自动切换：**通过页面一键迁移按钮，低成本实现从非分布式版本切换到分布式版本（迁移进度可感知，执行过程中仓库可读不可写，确保数据完整）。
- **Proxy 屏蔽底层存储介质多样性：**通过 Proxy 保持单一的调用方式不变，可兼顾获取非分布式版本和分布式版本的存储数据。
- **特殊数据共享存储：**用户和 SSH Public Key 等数据与仓库数据没有强制关联关系，可实现数据共享。

c. 历史数据平滑迁移

Code 系统存在众多的历史代码数据和业务数据，如何有效、完整地将历史数据平滑迁移到新的分布式系统，变得尤为重要。为了达成业务使用无感知的目标，主要做了以下两件事情：

- **优先迁移“轻量”仓库：**先迁移使用功能单一的仓库，根据用户反馈逐步完善迁移能力。
- **业务维度批次迁移：**按照业务线划分迁移批次，同类使用模式的仓库同期迁移，以规避反馈问题多样性。

3.2 可用性目标

3.2.1 技术挑战

在进行可用性改造时，我们主要面临**数据安全性**层面的挑战。代码作为公司的重要资产之一，需达到两方面的要求：

1. 代码单点丢失可数据恢复。
2. 用户视角可以读到正确的代码数据。

3.2.2 方案选型

目前，业界主要有以下三种主流的数据复制模式。

数据保障		
方案	优势	劣势
单主复制	• 结构简单，易于实现，不会产生数据竞争冲突	• 数据写入依赖主节点，存在单点瓶颈 • 主从复制有一定的数据延迟，若主节点存在宕机故障时，会产生一定量的数据丢失
多主复制	• 多个主节点承担写流量，相比单主架构有更好的写入性能	• 依旧存在主从复制的延迟问题 • 两个主节点在同时写入数据产生竞争冲突时，需要通过一定的机制保障
无主复制	• 所有节点都可以承担读写，吞吐量大大提升	• 在写入数据时容忍数据不一致，在读取数据时需通过机制保障正确性 • 需要比较复杂的补偿和仲裁机制保障数据安全

图 9　数据保障方案对比

业界大多数分布式版本控制系统采用的是单主复制模式保障数据安全，随着美团内部研发流程的逐步完善，对于创建注释 Tag、分支管理等需求逐步增加，读写比从最初的 10:1 缩短到现在的 5:1，因此需要较高的写入性能。

我们权衡了高吞吐量和数据强一致性的双重目标，在单主复制架构的基础上，采用以仓库维度单主复制为核心，节点多活为特性的复制模式（下文简称为多活模式），从而保证了数据安全和系统可用性。

3.2.3 方案设计

我们主要通过存储模块中，对 Git 的读、写及初始化三类不同的请求分别采取相对应的数据处理机制，并结合多活复制模式，达成了高可用性的目标。

存储模块

Git Server：主要存储和管理 Git 仓库数据，提供 Git 相关的 gRPC 接口。该服务基于 GitLab 生态开源组件二次开发，主要在数据同步机制、容灾模块、部分底层命令上做了适配性优化，共涉及以下 4 个逻辑模块：

1. **Replication Manager：**数据复制核心模块，根据不同的请求（读、写或初始化）执行不同的复制逻辑，从而保障数据一致性。
2. **Code Core：**Git Server 的核心服务模块，主要提供了 Git 的 gRPC API 供上游模块使用。

3. **Git Core**：实现扩展性和高可用性的重要组件，这里通过源码的方式将 GitLab 生态开源组件引入到项目中，作为第三方 Git API 供项目使用。
4. **Git Command Factory**：Git 命令的中枢控制器，提供控制 Git 进程数量、传递参数上下文，隔离执行环境及格式化输出数据等功能。

各个逻辑模块间关联如下图 10 所示：

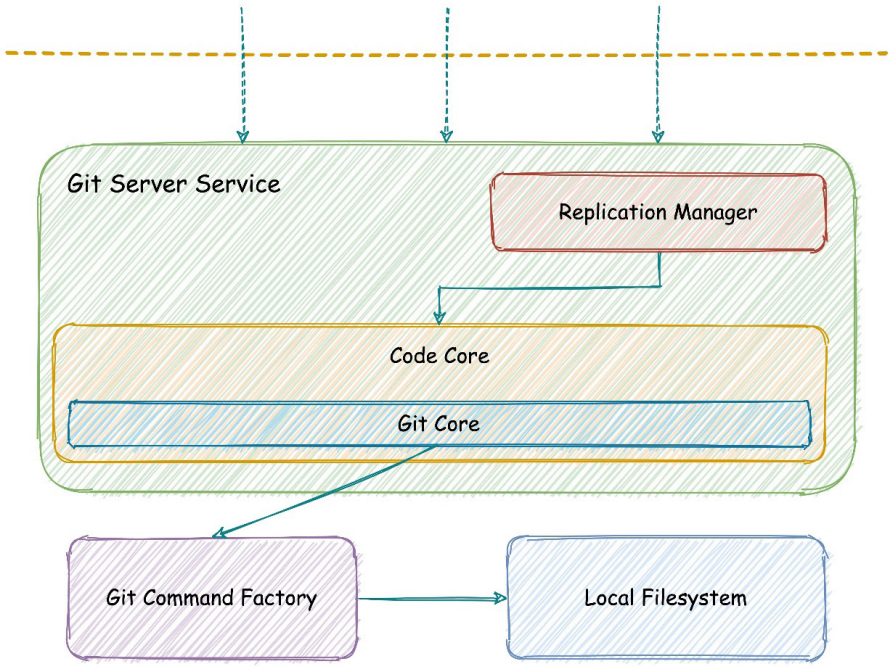


图 10　存储模块

Git Cluster：又称为分片，它由三个 Git Server 节点组成。三个节点间通过各自的 Replication Manager 模块获取到集群中其余节点的 IP 等信息，使用 gRPC 协议进行数据复制备份，可以保证用户视角的数据一致性，逻辑架构如下图 11 所示：

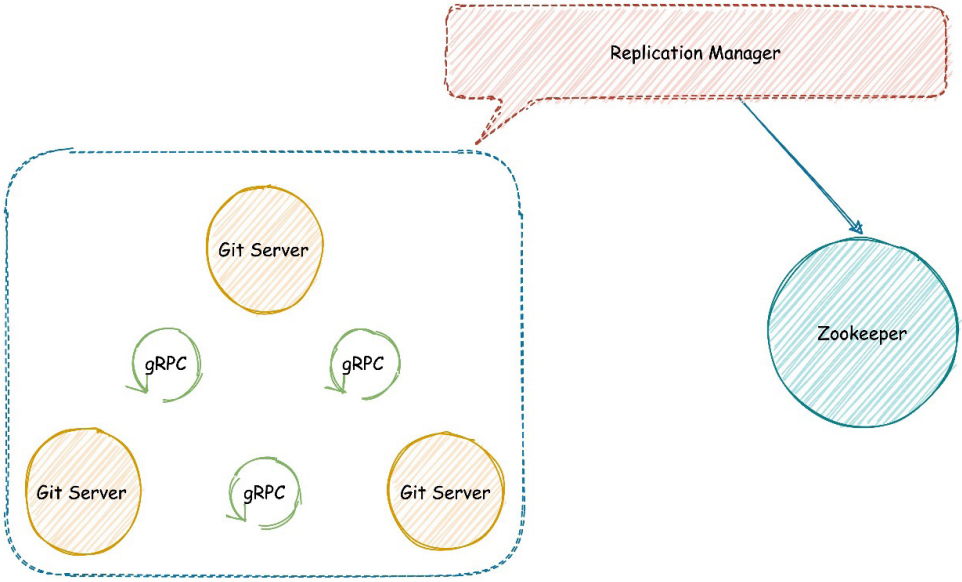


图 11　Git Cluster

3.2.4 解决思路

数据安全性解决思路

Code 系统要解决的问题中，数据安全问题尤为重要，是保证研发流程安全可靠的关键。在考虑数据安全性解决思路之前，先要明确数据一致性判别准则，Code 采用以下准则评判两个仓库数据一致。

数据一致评判准则：若仓库所在两个节点存储的 refs 数据完全一致，则称为这两个节点上的仓库数据一致。

目前系统数据安全机制主要有以下几个特点：

a. 多活复制

目前 Code 系统每个分片包含 3 个节点，即代码数据保证三副本，即使出现 1~2 台节点故障导致数据不可恢复的情况，也可通过其他节点进行数据恢复。我们采用了多活复制模式，即任何一个满足必要条件（当前访问仓库在该节点的数据均重演至最新版本）的机器节点均可以进行读写操作，与单主模式相比提高了写操作的吞吐量，节省了主备

切换的成本，使部署、节点替换及异常恢复更加简单。多活复制模式约束有以下两点：

1. “单写”机制：在同一时刻，同一个仓库的写操作须在同一节点进行。
2. 数据安全锁机制：若某仓库底层 Git 的操作出现异常错误，则在数据未恢复前，其后对该仓库的所有操作均会在该节点进行，会产生局部热点。

多活复制主要由数据存储和数据压缩两个部分组成。

01 数据存储

1. Git 主要由 objects 和 refs 两类数据组成。objects 数据为不可变数据，创建后为只读模式，以文件的形式存储于本地磁盘中；refs 数据为可变数据，可以进行更新。两类数据分别采用不同数据源进行存储。
2. 用户在访问仓库时，如果某个 objects 没有在任何一个分支的关联链中，那么判定为不可达，对于不可达的 objects，无需维护其一致性。不可达 object 的示例如下：

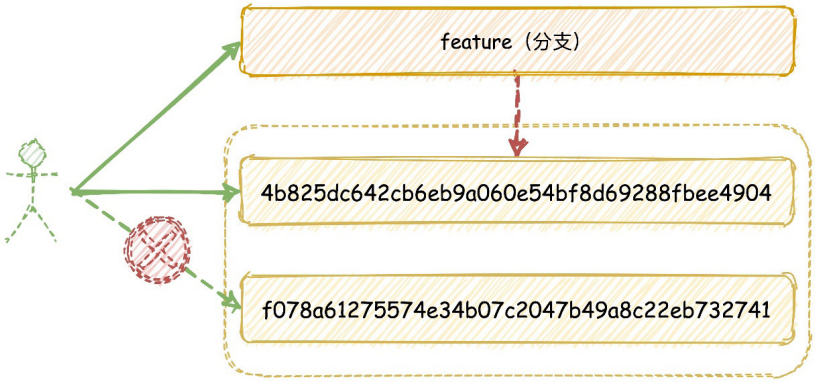


图 12　不可达 object 示例图

02 数据压缩

在 Code 系统中，需要记录 refs 的变更日志以进行数据回放，保证系统的数据一致性。由于每个仓库的 refs 数据变换是比较频繁的，会产生大量的日志，从而造成存储压力。因而我们采用了日志压缩技术，减少不必要的数据开销，压缩方式如下图 13 所示：

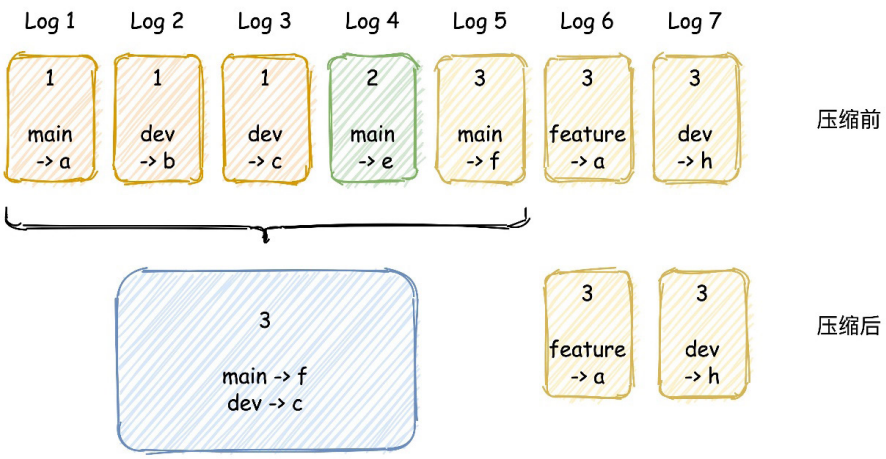


图 13　数据压缩

例如上图中的 main 分支，其初始状态为 `main -> a`，第 4 个 log 为 `main -> e`，第 5 个 log 为 `main -> f`，则这 3 个 log 可以压缩为一个 log，即 `main -> f` 并将其应用于初始状态，与压缩前回放触发的结果是一致的，main 都将指向值为 f 的 commit。

03 相关优化

在实践过程中，我们发现采用纯 Git 命令执行数据复制操作无法有效控制资源分配，因而从通信方式、并发形式及复制粒度等方面做了优化，从而提高了整体的数据复制效率。

b. 跨机房备份

Code 系统每组分片的 3 个节点至少来自于两个不同的机房（目前按照规范化部署，均改造为 3 机房），若其中一个机房发生故障，仍可提供服务。我们对该架构做了针对性的容灾演练，通过演练验证了节点掉线对系统的影响较小，结合灵活的节点替换能力，可在 30 分钟内上线新的节点，达到容灾平衡状态。

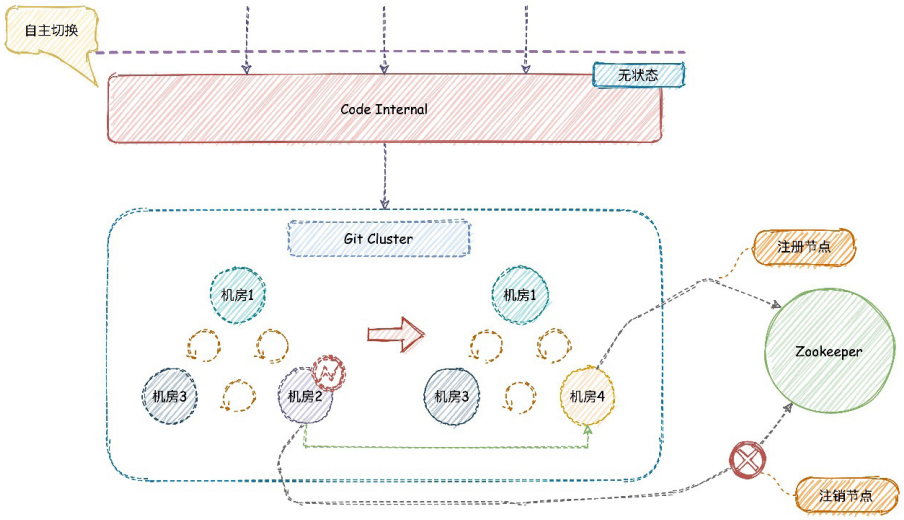


图 14　跨机房备份

c. 数据热备

Code 系统提供数据热备机制，通过数据复制的方式，任何写入的新数据会立即同步到其余副本节点，基本“0”延迟，保证了用户视角的强一致性。数据同步机制是热备的关键，我们主要通过以下步骤实现。

01 写操作阶段

1. 通过引入仓库粒度的写锁，保证同一个仓库同时只能在一个节点执行写入操作，然后通过 Git Internal Hook 机制触发 object 数据的同步，并持久化记录 refs 数据。
2. 副本节点通过读取持久化的 refs 数据，重演操作，从而保持了 refs 数据与写入节点一致。

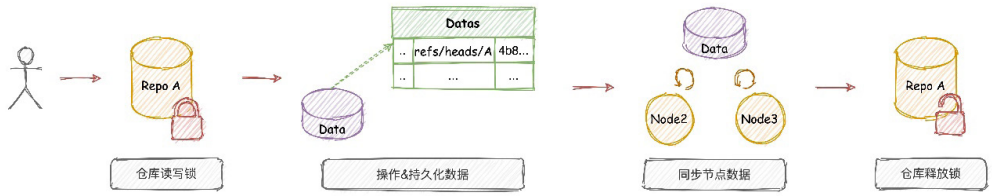


图 15　写操作步骤

02 读操作阶段

如果当前仓库持有写锁，则直接路由至持有写锁的节点读取数据。

如果未持有写锁，则用各个节点的版本和数据源存储的版本数据进行对比，将版本大于等于数据源存储的最新版本的所有节点作为候选路由节点，并采用负载均衡算法进行路由；如果没有符合条件的节点则需进行同步补偿，待补偿成功后再进行路由选择 。

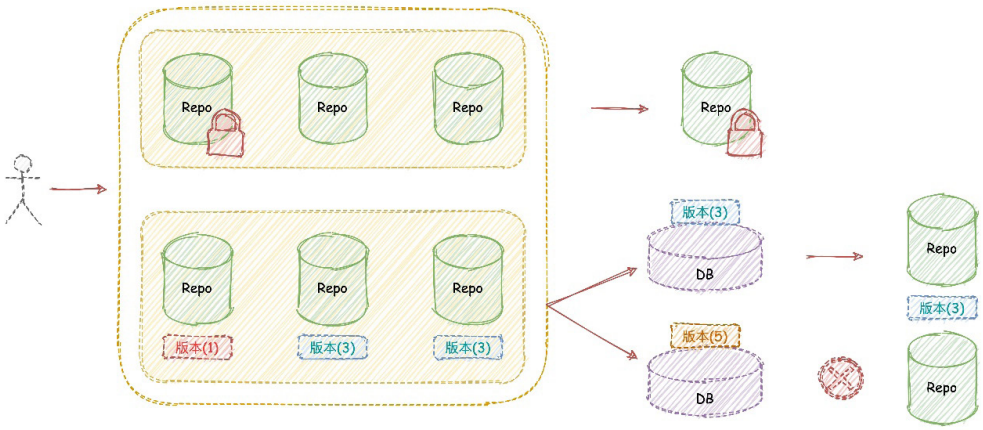


图 16　读操作步骤

03 相关优化

在最初实现中，我们采用了无状态同步，发现存在同步任务被多次执行的情况，后续通过任务前置检查等方式避免了不必要的数据同步任务，最终减少了 50% 的同步任务。

d. 数据巡检

数据巡检是保证系统平稳运行，数据安全可靠必不可少的一个环节，它可以及早地发现系统中潜在的隐患。巡检服务作为 Code 系统的核心服务，在降低数据风险，提高系统服务的稳定性方面起到了关键作用。对于巡检服务，我们主要从以下几个方面进行考虑：

- **透明性：**尽可能地避免对用户的正常请求产生影响，减少不必要的干扰，对于系统访问可以做到平稳可控。
- **可靠性：**作为数据安全的重要服务，它自身也要做到弹性伸缩，多点容灾，具有

高可用的特性。

- **可维护性：**对于数据巡检发现的问题，能够通过有效手段进行处理。同时要提高巡检服务的效率，随着系统架构的迭代出新、模块升级，巡检服务要随之更新，从而做到有效的保障。

综合以上几点，我们采用了无状态的服务架构，提供定点巡检、全量巡检、定时巡检等模式保障数据安全。其中巡检的数据主要分为以下两类：

- **refs 数据：**根据数据一致性评判准则，refs 数据是 Git 核心数据，因而它的检验是必不可少的。
- **版本数据：**Code 系统是基于版本进行读写路由的，因而当版本过大时，可能会产生大量的数据同步，为了避免突增同步请求对系统造成一定的 IO 抖动，监控版本差距是尤为必要的。

巡检服务的整体架构如下图 17 所示：

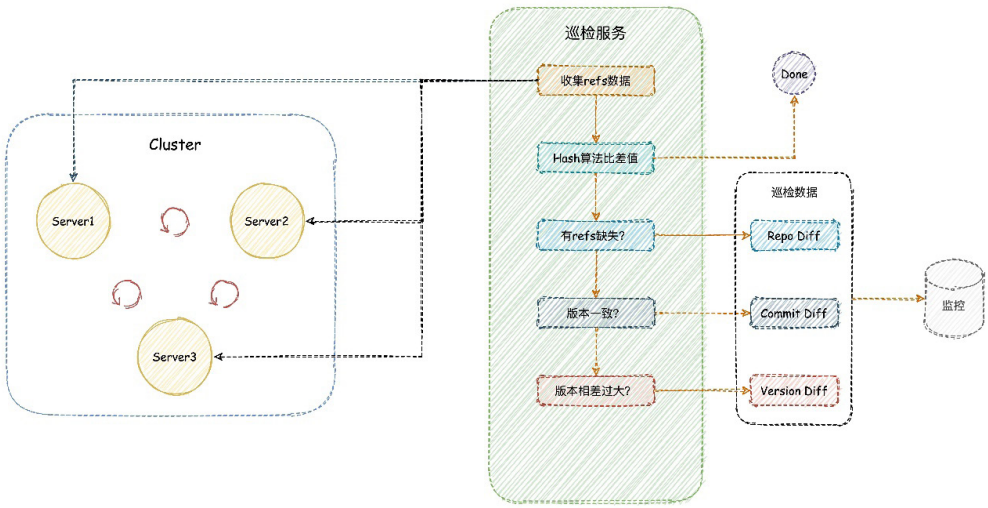


图 17　巡检模块

4. 总结

本文系统性地介绍了美团在 Code 系统演进过程中面临的扩展性和可用性两大瓶颈，并分别针对上述两类瓶颈和对应的挑战，详细阐述了解决方案和落地的实践经验。

基于上述的架构改造实践，目前美团代码托管平台实现了仓库容量水平扩展、负载自主均衡等特性，稳定支撑着研发流程规范的落地。我们未来会在支撑研发效率，保障研发安全方面继续进行探索和演进，争取积累更多宝贵的实践经验，后续再跟大家分享。

5. 未来展望

自动化运维：目前系统的运维机制自动化程度低，我们希望未来可以自动检出系统异常并进行恢复，其中包括数据修复，自动扩容及热点迁移等功能。

提供代码领域最佳实践：依托研发工具平台，持续推动美团研发流程规范的迭代更新，沉淀最佳实践并提供有力的工具支撑。

代码安全：与信息安全团队紧密合作，提供更为完备的安全控制策略，包括代码扫描、漏洞自动修复、危险行为预警等功能。

6. 本文作者及团队简介

潘陶、费翔、丹丹、毛强等，来自基础研发平台 – 研发质量与效率团队。
美团研发质量与效率团队，负责公司研发效能领域平台和工具的建设（包括研发需求管理工具、CI/CD流水线、分布式代码托管平台、多语言构建工具、发布平台、测试环境管理平台、全链路压测平台等），致力于不断推进优秀的研发理念和工程实践，建设一流的工程基础设施。

一次「找回」TraceId 的问题分析与过程思考

作者：李祯

1. 问题背景和思考

1.1 问题背景

在一次排查线上告警的过程中，突然发现一个链路信息有点不同寻常（这里仅展示测试复现的内容）：



在机器中可以清楚的发现“2022-08-02 19:26:34.952 DXMsgRemoteService ”这一行日志信息并没有携带 TraceId，导致调用链路信息戛然而止，无法追踪当时的调用情况。

1.2 问题复现和思考

在处理完线上告警后，我们开始分析“丢失”的 TraceId 到底去了哪里？首先在代码中定位 TraceId 没有追踪到的部分，发现问题出现在一个 @Async 注解下的方法，删除无关的业务信息代码，并增加 MTrace 埋点方法后的复现代码如下：

```
@SpringBootTest
@RunWith(SpringRunner.class)
@EnableAsync
public class DemoServiceTest extends TestCase {
      @Resource
              private DemoService demoService;
      @Test
              public void testTestAsy() {
              Tracer.serverRecv("test");
              String mainThreadName = Thread.currentThread().getName();
              long mainThreadId = Thread.currentThread().getId();
              System.out.println("------We got main thread: "+ mainThreadName
 + " - " +  mainThreadId + "  Trace Id: " + Tracer.id() + "----------");
              demoService.testAsy();
      }
}
```

```
@Component
public class DemoService {
      @Async
              public void testAsy(){
              String asyThreadName = Thread.currentThread().getName();
              long asyThreadId = Thread.currentThread().getId();
              System.out.println("======Async====");
              System.out.println("------We got asy thread: "+ asyThreadName +
" - " +  asyThreadId + "  Trace Id: " + Tracer.id() + "----------");
      }
}
```

运行这段代码后，我们看看控制台实际的输出结果：

```
------We got main thread: main - 1  Trace Id: -5292097998940230785----------
======Async====
------We got asy thread: SimpleAsyncTaskExecutor-1 - 630  Trace Id: null----------
```

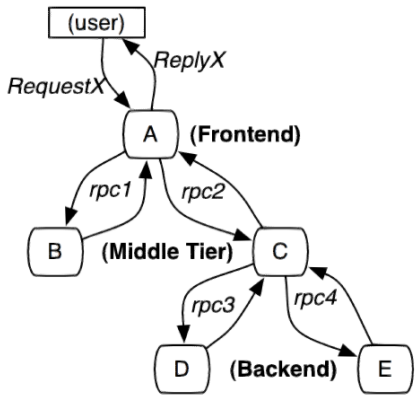
至此我们可以发现 TraceId 是在 @Async 异步传递的过程中发生丢失现象，明白了造成这一现象的原因后，我们开始思考：

- MTrace（美团内部自研的分布式链路追踪系统）这类分布式链路追踪系统是如何设计的?
- @Async 异步方法是如何实现的?
- InheritableThreadLocal、TransmittableThreadLocal 和 TransmissibleThreadLocal 有什么区别?
- 为什么 MTrace 的跨线程传递方案“失效”了?
- 如何解决 @Async 场景下“弄丢”TraceId 的问题?
- 目前有哪些分布式链路追踪系统? 它们又是如何解决跨线程传递问题的?

2. 深度分析

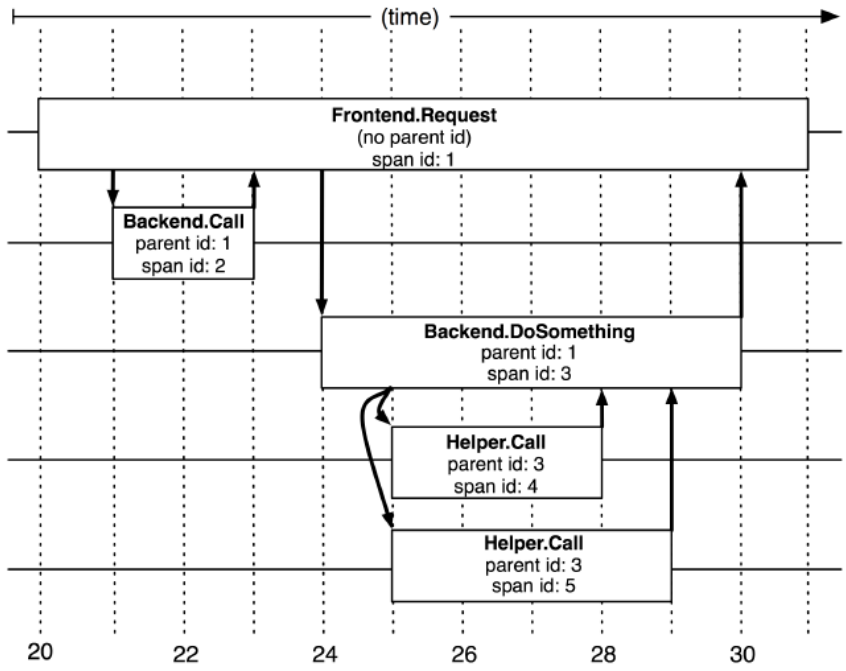
2.1 MTrace 与 Google Dapper

MTrace 是美团参考 Google Dapper 对服务间调用链信息收集和整理的分布式链路追踪系统，目的是帮助开发人员分析系统各项性能和快速排查告警问题。要想了解 MTrace 是如何设计分布式链路追踪系统的，首先看看 Google Dapper 是如何在大型分布式环境下实现分布式链路追踪。我们先来看看下图一个完整的分布式请求：

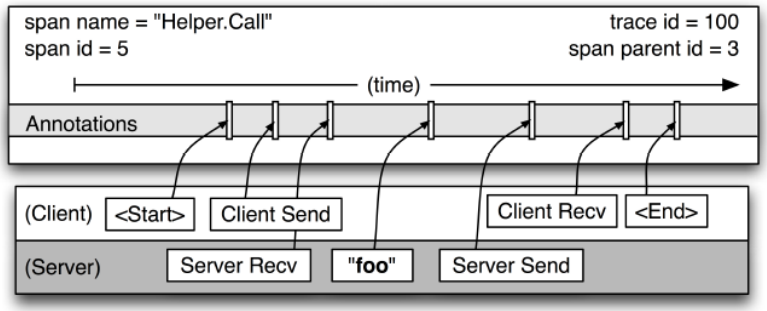


用户发送一个请求到前端 A，然后请求分发到两个不同的中间层服务 B 和 C，服务 B 在处理完请求后将结果返回，同时服务 C 需要继续调用后端服务 D 和 E 再将处理后的请求结果进行返回，最后由前端 A 汇总来响应用户的这次请求。

回顾这次完整的请求我们不难发现，要想直观可靠的追踪多项服务的分布式请求，我们最关注的是每组客户端和服务端之间的请求响应以及响应耗时，因此，Google Dapper 采取对每一个请求和响应设置标识符和时间戳的方式实现链路追踪，基于这一设计思想的基本追踪树模型如下图所示：



追踪树模型由 span 组成，其中每个 span 包含 span name、span id、parent id 和 trace id，进一步分析跟踪树模型中各个 span 之间的调用关系可以发现，其中没有 parent id 且 span id 为 1 代表根服务调用，span id 越小代表服务在调用链的过程中离根服务就越近，将模型中各个相对独立的 span 联系在一起就构成了一次完整的链路调用记录，我们再继续深入看看 span 内部的细节信息：



除了最基本的 span name、span id 和 parent id 之外，Annotations 扮演着重要的角色，Annotations 包括 <Strat>、Client Send、Server Recv、Server Send、Client Recv 和 <End> 这些注解，记录了 RPC 请求中 Client 发送请求到 Server 的处理响应时间戳信息，其中 foo 注解代表可以自定义的业务数据，这些也会一并记录到 span 中，提供给开发人员记录业务信息；在这当中有 64 位整数构成的 trace id 作为全局的唯一标识存储在 span 中。

至此我们已经了解到，Google Dapper 主要是在每个请求中配置 span 信息来实现对分布式系统的追踪，那么又是用什么方式在分布式请求中植入这些追踪信息呢?

为满足低损耗、应用透明和大范围部署的设计目标，Google Dapper 支持应用开发者依赖于少量通用组件库，实现几乎零投入的成本对分布式链路进行追踪，当一个服务线程在链路中调用其他服务之前，会在 ThreadLocal 中保存本次跟踪的上下文信息，主要包括一些轻量级且易复制的信息（类似 spand id 和 trace id），当服务线程收到响应之后，应用开发者可以通过回调函数进行服务信息日志打印。

MTrace 是美团参考 Google Dapper 的设计思路并结合自身业务进行了改进和完善后的自研产品，具体的实现流程这里就不再赘述了，我们重点看看 MTrace 做了哪些改进：

- 在美团的各个中间件中埋点，来采集发生调用的调用时长和调用结果等信息，埋点的上下文主要包括传递信息、调用信息、机器相关信息和自定义信息，各个调用链路之间有一个全局且唯一的变量 TraceId 来记录一次完整的调用情况和追踪数据。
- 在网络间的数据传递中，MTrace 主要传递使用 UUID 异或生成的 TraceId 和表示层级和前后关系的 SpanId，支持批量压缩上报、TraceId 做聚合和 SpanId 构建形态。
- 目前，产品已经覆盖到 RPC 服务、HTTP 服务、MySQL、Cache 缓存和 MQ，基本实现了全覆盖。
- MTrace 支持跨线程传递和代理来优化埋点方式，减轻开发人员的使用成本。

2.2 @Async 的异步过程追溯

从 Spring3 开始提供了 @Async 注解，该注解的使用需要注意以下几点：

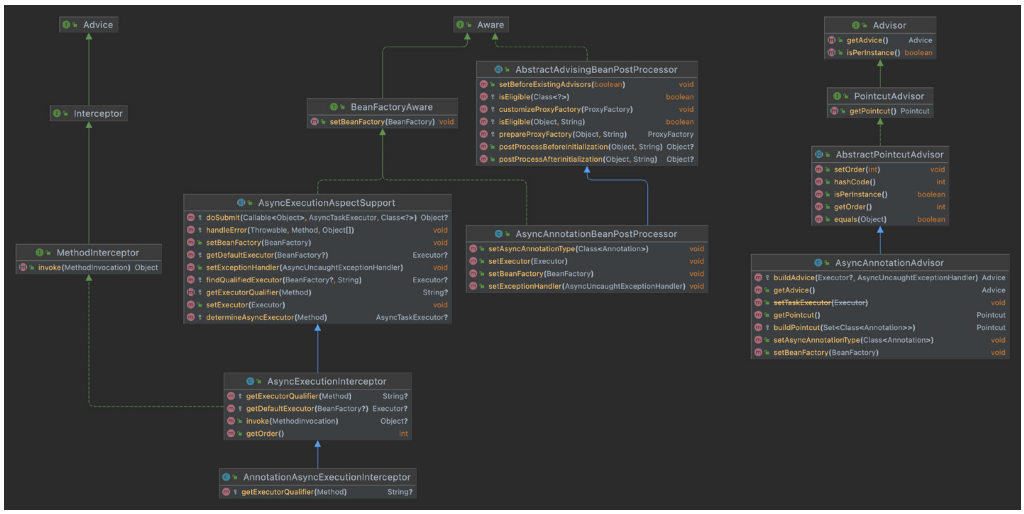
1. 需要在配置类上增加 @EnableAsync 注解；
2. @Async 注解可以标记一个异步执行的方法，也可以用来标记一个类表明该类的所有方法都是异步执行；
3. 可以在 @Async 中自定义执行器。

我们以 @EnableAsync 为入口开始分析异步过程，除了基本的配置方法外，我们重点关注下配置类 AsyncConfigurationSelector 的内部逻辑，由于默认条件下我们使用 JDK 接口代理，这里重点看看 ProxyAsyncConfiguration 类的代码逻辑：

```
@Configuration
@Role(BeanDefinition.ROLE_INFRASTRUCTURE)
public class ProxyAsyncConfiguration extends AbstractAsyncConfiguration {
      @Bean(name = TaskManagementConfigUtils.ASYNC_ANNOTATION_PROCESSOR_
BEAN_
NAME)
              @Role(BeanDefinition.ROLE_INFRASTRUCTURE)
              public AsyncAnnotationBeanPostProcessor asyncAdvisor() {
              Assert.notNull(this.enableAsync, "@EnableAsync annotation
metadata was not
injected");
              // 新建一个异步注解 bean 后置处理器
              AsyncAnnotationBeanPostProcessor bpp = new
AsyncAnnotationBeanPostProcessor();
              // 如果 @EnableAsync 注解中有自定义 annotation 配置则进行设置
              Class<? extends Annotation> customAsyncAnnotation = this.
```

```
enableAsync.
getClass("annotation");
                if (customAsyncAnnotation != AnnotationUtils.
getDefaultValue(EnableAsync.class,
"annotation")) {
                        bpp.setAsyncAnnotationType(customAsyncAnnotation);
                }
                if (this.executor != null) {
                        // 设置线程处理器
                        bpp.setExecutor(this.executor);
                }
                if (this.exceptionHandler != null) {
                        // 设置异常处理器
                        bpp.setExceptionHandler(this.exceptionHandler);
                }
                // 设置是否需要创建 CGLIB 子类代理，默认为 false
                bpp.setProxyTargetClass(this.enableAsync.
getBoolean("proxyTargetClass"));
                // 设置异步注解 bean 处理器应该遵循的执行顺序，默认最低的优先级
                bpp.setOrder(this.enableAsync.<Integer>getNumber("order"));
                return bpp;
        }
}
```

ProxyAsyncConfiguration 继承了父类 AbstractAsyncConfiguration 的方法，重点定义了一个 AsyncAnnotationBeanPostProcessor 的异步注解 bean 后置处理器。看到这里我们可以知道，@Async 主要是通过后置处理器生成一个代理对象来实现异步的执行逻辑，接下来我们重点关注 AsyncAnnotationBeanPostProcessor 是如何实现异步的：



从类图中我们可以直观地看到 AsyncAnnotationBeanPostProcessor 同时实现了 BeanFactoryAware 的接口，因此我们进入 setBeanFactory() 方法，可以看到对 AsyncAnnotationAdvisor 异步注解切面进行了构造，再接着进入 AsyncAnnotationAdvisor 的 buildAdvice() 方法中可以看 AsyncExecutionInterceptor 类，再看类图发现 AsyncExecutionInterceptor 实现了 MethodInterceptor 接口，而 MethodInterceptor 是 AOP 中切入点的处理器，对于 interceptor 类型的对象，处理器中最终被调用的是 invoke 方法，所以我们重点看看 invoke 的代码逻辑：

```
public Object invoke(final MethodInvocation invocation) throws Throwable {
      Class<?> targetClass = (invocation.getThis() != null ? AopUtils.
getTargetClass(invocation.getThis()) : null);
      Method specificMethod = ClassUtils.getMostSpecificMethod(invocation.
getMethod(), targetClass);
      final Method userDeclaredMethod = BridgeMethodResolver.
findBridgedMethod(specificMethod);
  // 首先获取到一个线程池
      AsyncTaskExecutor executor = determineAsyncExecutor(userDeclaredMethod);
      if (executor == null) {
              throw new IllegalStateException("No executor specified and no
default executor set on AsyncExecutionInterceptor either");
      }
  // 封装 Callable 对象到线程池执行
      Callable<Object> task = () -> {
              try {
                      Object result = invocation.proceed();
                      if (result instanceof Future) {
                              return ((Future<?>) result).get();
                      }
              }
              catch (ExecutionException ex) {
                      handleError(ex.getCause(), userDeclaredMethod,
invocation.getArguments());
              }
              catch (Throwable ex) {
                      handleError(ex, userDeclaredMethod, invocation.
getArguments());
              }
              return null;
      };
  // 任务提交到线程池
      return doSubmit(task, executor, invocation.getMethod().getReturnType());
}
```

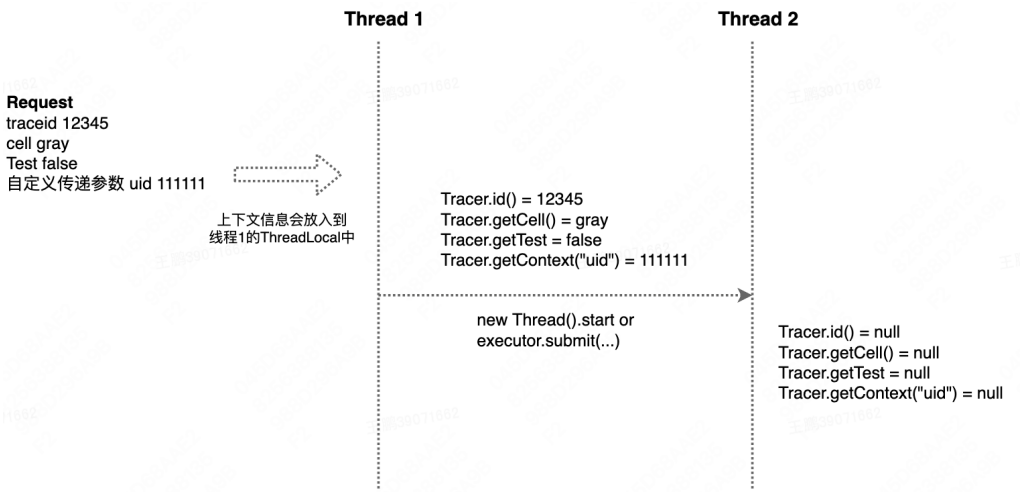
我们再接着看看 @Async 用了什么线程池，重点关注 determineAsyncExecutor 方法中 getExecutorQualifier 指定获取的默认线程池是哪一个：

```
@Override
@Nullable
protected Executor getDefaultExecutor(@Nullable BeanFactory beanFactory) {
      Executor defaultExecutor = super.getDefaultExecutor(beanFactory);
      return (defaultExecutor != null ? defaultExecutor : new
SimpleAsyncTaskExecutor()); // 其中默认线程池是 SimpleAsyncTaskExecutor
}
```

至此，我们了解到在未指定线程池的情况下调用被标记为 @Async 的方法时，Spring 会自动创建 SimpleAsyncTaskExecutor 线程池来执行该方法，从而完成异步执行过程。

2.3. “丢失”TraceId 的原因

回顾我们之前对 MTrace 的学习和了解，TraceId 等信息是在 ThreadLocal 中进行传递和保存，那么当异步方法切换线程的时候，就会出现下图中上下文信息传递丢失的问题：



下面我们探究一下 ThreadLocal 有哪些跨线程传递方案？ MTrace 又提供哪些跨线程传递方案？ SimpleAsyncTaskExecutor 又有什么不一样？逐步找到“丢失”TraceId 的原因。

2.3.1 InheritableThreadLocal、TransmittableThreadLocal 和 TransmissibleThreadLocal

在前面的分析中，我们发现跨线程场景下上下文信息是保存在 ThreadLocal 中发生丢

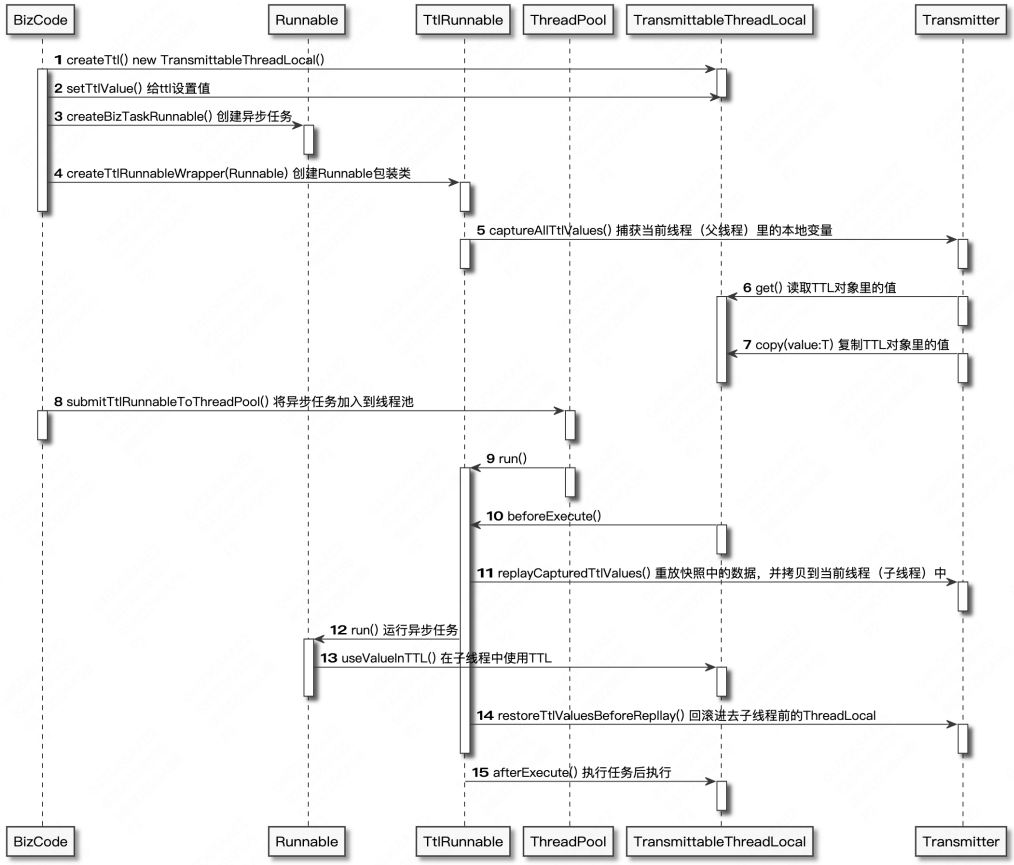
失，那么我们接下来看看 ThreadLocal 的特点及其延伸出来的类，是否可以解决这一问题：

- ThreadLocal 主要是为每个 ThreadLocal 对象创建一个 ThreadLocalMap 来保存对象和线程中的值的映射关系。当创建一个 ThreadLocal 对象时会调用 get() 或 set() 方法，在当前线程的中查找这个 ThreadLocal 对象对应的 Entry 对象，如果存在，就获取或设置 Entry 中的值；否则，在 ThreadLocalMap 中创建一个新的 Entry 对象。ThreadLocal 类的实例被多个线程共享，每个线程都拥有自己的 ThreadLocalMap 对象，存储着自己线程中的所有 ThreadLocal 对象的键值对。ThreadLocal 的实现比较简单，但需要注意的是，如果使用不当，可能会出现内存泄漏问题，因为 ThreadLocalMap 中的 Entry 对象并不会自动删除。
- InheritableThreadLocal 的实现方式和 ThreadLocal 类似，但不同之处在于，当一个线程创建子线程时会调用 init() 方法：

```
private void init(ThreadGroup g, Runnable target, String name,long stackSize,
AccessControlContext acc,Boolean inheritThreadLocals) {
      if (inheritThreadLocals && parent.inheritableThreadLocals != null)
  // 拷贝父线程的变量
      this.inheritableThreadLocals =ThreadLocal.createInheritedMap(parent.
inheritableThreadLocals);
      this.stackSize = stackSize;
      tid = nextThreadID();
}
```

这意味着子线程可以访问父线程中的 InheritableThreadLocal 实例，而且在子线程中调用 set() 方法时，会在子线程自己的 inheritableThreadLocals 字段中创建一个新的 Entry 对象，而不会影响父线程中的 Entry 对象。同时，根据源码我们也可以看到 Thread 的 init() 方法是在线程构造方法中拷贝的，在线程复用的线程池中是没有办法使用的。

- TransmittableThreadLocal 是阿里巴巴提供的解决跨线程传递上下文的 InheritableThreadLocal 子类，引入了 holder 来保存需要在线程间进行传递的变量，大致流程我们可以参考下面给出的时序图分析：



步骤可以总结为：① 装饰 Runnable，将主线程的 TTL 传入到 TtlRunnable 的构造方法中；② 将子线程的 TTL 的值进行备份，将主线程的 TTL 设置到子线程中（value 是对象引用，可能存在线程安全问题）；③ 执行子线程逻辑；④ 删除子线程新增的 TTL，将备份还原重新设置到子线程的 TTL 中，从而保证了 ThreadLocal 的值在多线程环境下的传递性。

TransmittableThreadLocal 虽然解决了 InheritableThreadLocal 的继承问题，但是由于需要在序列化和反序列化时对 ThreadLocalMap 进行处理，会增加对象创建和序列化的成本，并且需要支持的序列化框架较少，不够灵活。

- TransmissibleThreadLocal 是继承了 InheritableThreadLocal 类并重写了 get()、set() 和 remove() 方法，TransmissibleThreadLocal 的实现方式和 TransmittableThreadLocal 类似，主要的执行逻辑在 Transmitter 的

capture() 方法复制 holder 中的变量，replay() 方法过滤非父线程的 holder 的变量，restore() 来恢复经过 replay() 过滤后 holder 的变量：

```
public class TransmissibleThreadLocal<T> extends InheritableThreadLocal<T> {
      public static class Transmitter {
              public static Object capture() {
                      Map<TransmissibleThreadLocal<?>, Object> captured = new
HashMap<TransmissibleThreadLocal<?>, Object>();
      // 获取所有存储在 holder 中的变量
                      for (TransmissibleThreadLocal<?> threadLocal : holder.
get().keySet()) {
                              captured.put(threadLocal, threadLocal.
copyValue());
                      }
                      return captured;
              }
              public static Object replay(Object captured) {
                      @SuppressWarnings("unchecked")
                      Map<TransmissibleThreadLocal<?>, Object> capturedMap =
(Map<TransmissibleThreadLocal<?>, Object>) captured;
                      Map<TransmissibleThreadLocal<?>, Object> backup = new
HashMap<TransmissibleThreadLocal<?>, Object>();
                      for (Iterator<? extends Map.
Entry<TransmissibleThreadLocal<?>, ?>> iterator = holder.get().entrySet().
iterator();iterator.hasNext(); ) {
                              Map.Entry<TransmissibleThreadLocal<?>, ?> next =
iterator.next();
                              TransmissibleThreadLocal<?> threadLocal = next.
getKey();
                              // backup
                              backup.put(threadLocal, threadLocal.get());
                              // clear the TTL value only in captured
                              // avoid extra TTL value in captured, when run
task.
        // 过滤非传递的变量
                              if (!capturedMap.containsKey(threadLocal)) {
                                      iterator.remove();
                                      threadLocal.superRemove();
                              }
                      }
                      // set value to captured TTL
                      for (Map.Entry<TransmissibleThreadLocal<?>, Object> entry
: capturedMap.entrySet()) {
                              @SuppressWarnings("unchecked")
                              TransmissibleThreadLocal<Object> threadLocal =
(TransmissibleThreadLocal<Object>) entry.getKey();
                              threadLocal.set(entry.getValue());
                      }
                      // call beforeExecute callback
                      doExecuteCallback(true);
                      return backup;
              }
```

```
            public static void restore(Object backup) {
                    @SuppressWarnings("unchecked")
                    Map<TransmissibleThreadLocal<?>, Object> backupMap =
(Map<TransmissibleThreadLocal<?>, Object>) backup;
                    // call afterExecute callback
                    doExecuteCallback(false);
                    for (Iterator<? extends Map.
Entry<TransmissibleThreadLocal<?>, ?>> iterator = holder.get().entrySet().
iterator();
                                                                    iterator.
hasNext(); ) {
                            Map.Entry<TransmissibleThreadLocal<?>, ?> next =
iterator.next();
                            TransmissibleThreadLocal<?> threadLocal = next.
getKey();
                            // clear the TTL value only in backup
                            // avoid the extra value of backup after
restore
                            if (!backupMap.containsKey(threadLocal)) {
                                    iterator.remove();
                                    threadLocal.superRemove();
                            }
                    }
                    // restore TTL value
                    for (Map.Entry<TransmissibleThreadLocal<?>, Object> entry
: backupMap.entrySet()) {
                            @SuppressWarnings("unchecked")
                            TransmissibleThreadLocal<Object> threadLocal =
(TransmissibleThreadLocal<Object>) entry.getKey();
                            threadLocal.set(entry.getValue());
                    }
            }
    }
}
```

TransmissibleThreadLocal 不但可以解决跨线程的传递问题，还能保证子线程和主线程之间的隔离，但是目前跨线程拷贝 span 数据时，采用浅拷贝有丢失数据的风险。最后，我们可以根据下表综合对比：

特点	ThreadLocal	InheritableThreadLocal	TransmittableThreadLocal	TransmissibleThreadLocal
传递范围	仅限于当前线程	仅限于父子线程	可跨线程、跨线程池	可跨线程、跨线程池
传递方式	不传递	父线程到子线程	跟随线程池、手动传递	跟随线程池、手动传递
数据隔离	线程间数据隔离	线程间数据隔离	线程间数据隔离，但支持跨线程数据传递	线程间数据隔离，但支持跨线程数据传递
实现原理	ThreadLocalMap	ThreadLocalMap + 继承关系	ThreadLocalMap+TtlRunnable	ThreadLocalMap+Transmitter
使用场景	单线程、线程内部	父子线程间	异步、多线程	异步、多线程

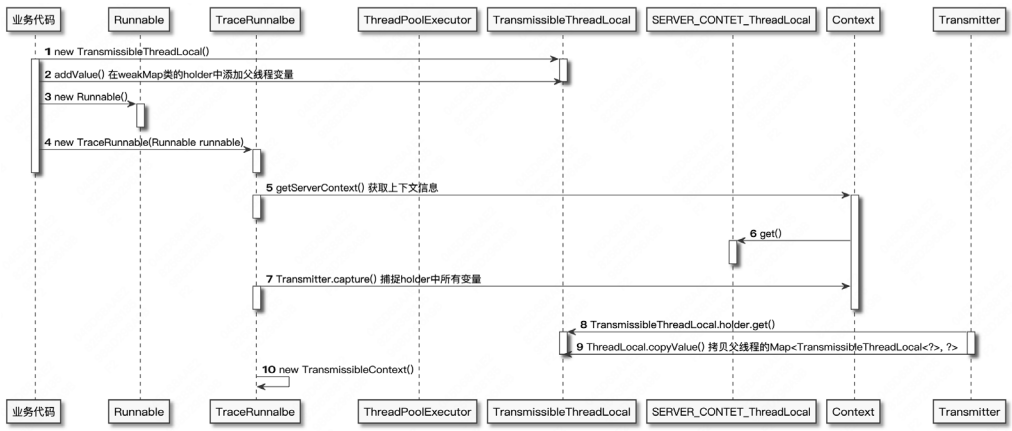
考虑到 TransmittableThreadLocal 并非标准的 Java API，而是第三方库提供的，存

在与其它库的兼容性问题，无形中增加了代码的复杂性和使用难度。因此，MTrace 选择自定义实现的 TransmissibleThreadLocal 类可以方便地在跨线程和跨服务的情况下传递追踪信息，透明自动完成所有异步执行上下文的可定制、规范化的捕捉传递，使得整个跟踪信息更加完整和准确。

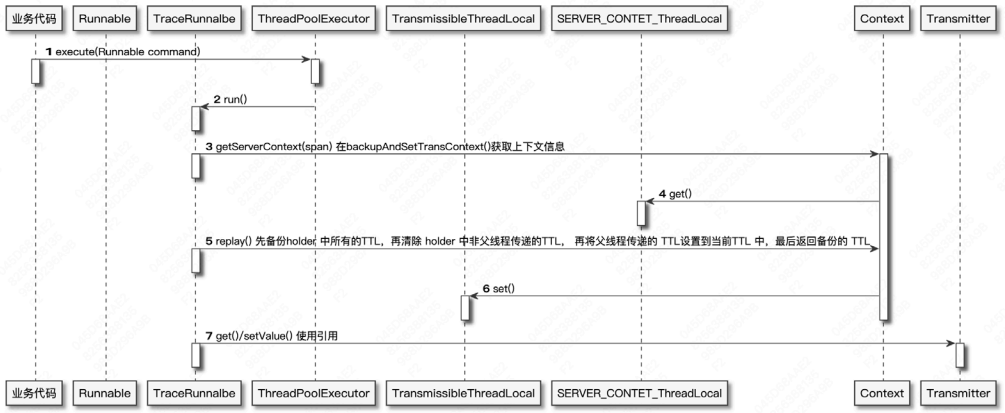
2.3.2 Mtrace 的跨线程传递方案

这一问题 MTrace 其实已经提供解决方案，主要的设计思路是在子线程初始化 Runnable 对象的时候首先会去父线程的 ThreadLocal 中拿到保存的 trace 信息，然后作为参数传递给子线程，子线程在初始化的时候设置 trace 信息来避免丢失。下面我们看看具体实现。

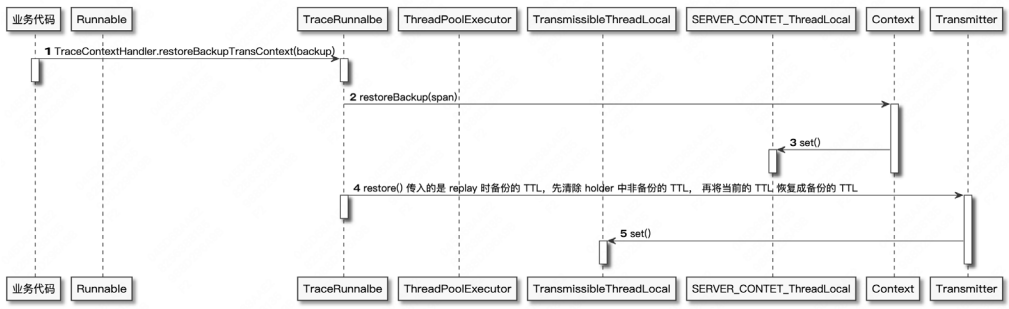
父线程新建任务时捕捉所有 TransmissibleThreadLocal 中的变量信息，如下图所示：



子线程执行任务时复制父线程捕捉的 TransmissibleThreadLocal 变量信息，并返回备份的 TransmissibleThreadLocal 变量信息，如下图所示：



在子线程执行完业务流程后会恢复之前备份的 TransmissibleThreadLocal 变量信息，如下图所示：



这种方案可以解决跨线程传递上下文丢失的问题，但是需要代码层面的开发会增加开发人员的工作量，对于一个分布式追踪系统而言并不是最优解：

```
TraceRunnable command = new TraceRunnable(runnable);
newThread(command).start();
executorService.execute(command);
```

因此，MTrace 同时提供无侵入方式的 javaagent&instrument 技术，可以简单理解成一个类加载时的 AOP 功能，只要在 JVM 参数添加 javaagent 的配置，不需要修饰 Runnable 或是线程池的代码，就可以在启动时增强完成跨线程传递问题。

回归到本次的问题中来，目前使用的 MDP 本身就已经集成了 MTrace-agent 的模式，但是为什么还是会“弄丢”TraceId 呢？查看 MTrace 的 ThreadPoolTransformer 类和 ForkJoinPoolTransformer 类我们可以知道，MTrace 修改了 ThreadPoolExec-

utor 类、ScheduledThreadPoolExecutor 类和 ForkJoinTask 类的字节码，顺着这个思路我们再看看 @Async 用到的 SimpleAsyncTaskExecutor 线程池是怎么一回事。

2.3.3 SimpleAsyncTaskExecutor 是怎么一回事

我们先深入 SimpleAsyncTaskExecutor 的代码中，看看执行逻辑：

```
public class SimpleAsyncTaskExecutor extends CustomizableThreadCreator
implements AsyncListenableTaskExecutor, Serializable {
      private ThreadFactory threadFactory;
      public void execute(Runnable task, long startTimeout) {
              Assert.notNull(task, "Runnable must not be null");
    //isThrottleActive 是否开启限流（默认 concurrencyLimit=-1，不开启限流）
              if(this.isThrottleActive() && startTimeout > 0L) {
                      this.concurrencyThrottle.beforeAccess();
                      this.doExecute(new SimpleAsyncTaskExecutor.
ConcurrencyThrottlingRunnable(task));
                      this.concurrencyThrottle.beforeAccess();
                      this.doExecute(new SimpleAsyncTaskExecutor.
ConcurrencyThrottlingRunnable(task));
                      this.concurrencyThrottle.beforeAccess();
                      this.doExecute(new SimpleAsyncTaskExecutor.
ConcurrencyThrottlingRunnable(task));
              } else {
                      this.doExecute(task);
              }
      }
      protected void doExecute(Runnable task) {
    // 没有线程工厂的话默认创建线程
              Thread thread = this.threadFactory != null?this.threadFactory.
newThread(task):this.createThread(task);
              thread.start();
      }
      public Thread createThread(Runnable runnable) {
    // 和线程池不同，每次都是创建新的线程
              Thread thread = new Thread(getThreadGroup(), runnable,
nextThreadName());
              thread.setPriority(getThreadPriority());
              thread.setDaemon(isDaemon());
              return thread;
      }
}
```

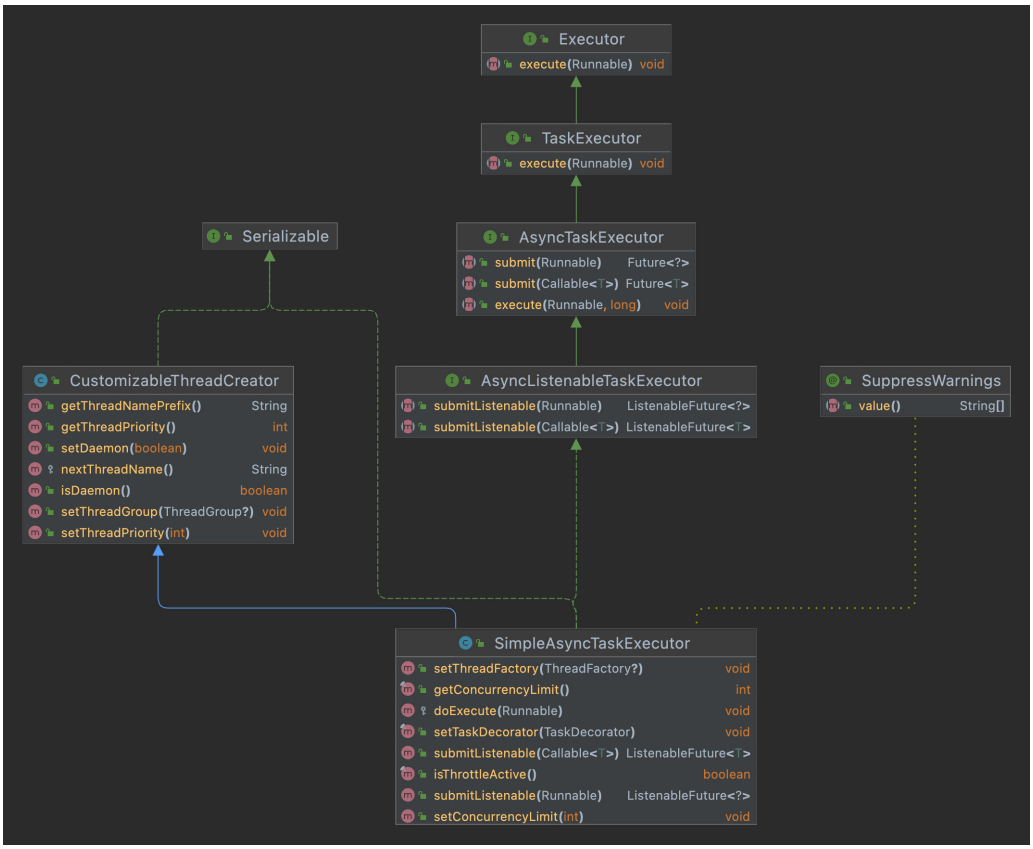
看到这里我们可以得出以下几个特性：

- SimpleAsyncTaskExecutor 每次执行提交给它的任务时，会启动新的线程，并不是严格意义上的线程池，达不到线程复用的功能。
- 允许开发者控制并发线程的上限（concurrencyLimit）起到一定的资源节流作用，

但默认 concurrencyLimit 取值为 −1，即不启用资源节流，有引发内存泄漏的风险。

- 阿里技术编码规约要求用 ThreadPoolExecutor 的方式来创建线程池，规避资源耗尽的风险。

结合之前说过的 MTrace 线程池代理模型，我们继续再来看看 SimpleAsyncTaskExecutor 的类图：



可以发现，其继承了 spring 的 TaskExecutor 接口，其实质是 java.util.concurrent.Executor，结合我们这次“丢失”的 TraceId 问题来看，我们已经找到了 Mtrace 的跨线程传递方案“失效”的原因：虽然 MTrace 已经通过 javaagent&instrument 技术可以完成 Trace 信息跨线程传递，但是目前只覆盖到 ThreadPoolExecutor 类、ScheduledThreadPoolExecutor 类和 ForkJoinTask 类的字节码，而 @Async 在未指定线程池的情况下默认会启用 SimpleAsyncTaskExecutor，其本质是 java.util.

concurrent.Executor 没有被覆盖到，就会造成 ThreadLocal 中的 get 方法获取信息为空，导致最终 TraceId 传递丢失。

3. 解决方案

实际上 @Async 支持我们使用自定义的线程池，可以手动自定义 Configuration 来配置 ThreadPoolExecutor 线程池，然后在注解里面指定 bean 的名称，就可以切换到对应的线程池去，可以看看下面的代码：

```
@Configuration
public class ThreadPoolConfig {
      @Bean("taskExecutor")
          public Executor taskExecutor() {
              ThreadPoolTaskExecutor taskExecutor = new ThreadPoolTaskExecutor();
              // 设置线程池参数信息
              taskExecutor.setCorePoolSize(10);
              taskExecutor.setMaxPoolSize(50);
              taskExecutor.setQueueCapacity(200);
              taskExecutor.setKeepAliveSeconds(60);
              taskExecutor.setThreadNamePrefix("myExecutor--");
              taskExecutor.setWaitForTasksToCompleteOnShutdown(true);
              taskExecutor.setAwaitTerminationSeconds(60);
              taskExecutor.setRejectedExecutionHandler(new ThreadPoolExecutor.
CallerRunsPolicy());
              taskExecutor.initialize();
              return taskExecutor;
      }
}
```

然后在注解中标注这个线程池：

```
@SpringBootTest
@RunWith(SpringRunner.class)
@EnableAsync
public class DemoServiceTest extends TestCase {
      @Resource
        private DemoService demoService;
      @Test
        public void testTestAsy() {
              Tracer.serverRecv("test");
              String mainThreadName = Thread.currentThread().getName();
              long mainThreadId = Thread.currentThread().getId();
              System.out.println("------We got main thread: "+ mainThreadName
+ " - " +  mainThreadId + "  Trace Id: " + Tracer.id() + "----------");
              demoService.testAsy();
      }
}
@Component
```

```
public class DemoService {
      @Async("taskExecutor")
        public void testAsy(){
              String asyThreadName = Thread.currentThread().getName();
              long asyThreadId = Thread.currentThread().getId();
              System.out.println("======Async====");
              System.out.println("------We got asy thread: "+ asyThreadName +
" - " +  asyThreadId + "  Trace Id: " + Tracer.id() + "----------");
      }
}
```

看看输出台的打印：

```
------We got main thread: main - 1  Trace Id: -3495543588231940494----------
======Async====
------We got asy thread: SimpleAsyncTaskExecutor-1 - 658  Trace Id:
3495543588231940494----------
```

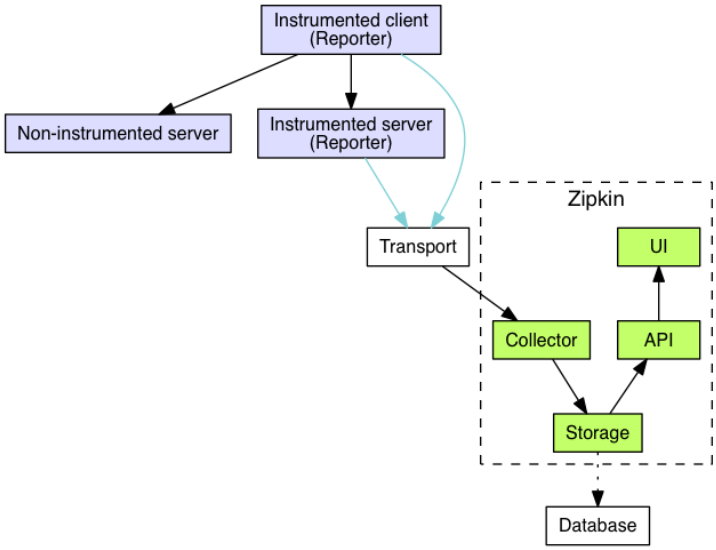
最终，我们可以通过这一方式“找回”在 @Async 注解下跨线程传递而“丢失”的 TraceId。

4. 其他方案对比

分布式追踪系统从诞生之际到有实质性的突破，很大程度受到 Google Dapper 的影响，目前常见的分布式追踪系统有 Twitter 的 Zipkin、SkyWalking、阿里的 Eagle-Eye、PinPoint 和美团的 MTrace 等，这些大多都是基于 Google Dapper 的设计思想，考虑到设计思路和架构特点，我们重点介绍 Zipkin、SkyWalking 和 EagleEye 的基本框架和跨线程解决方案（以下内容主要来源官网及作者总结，仅供参考，不构成技术建议）。

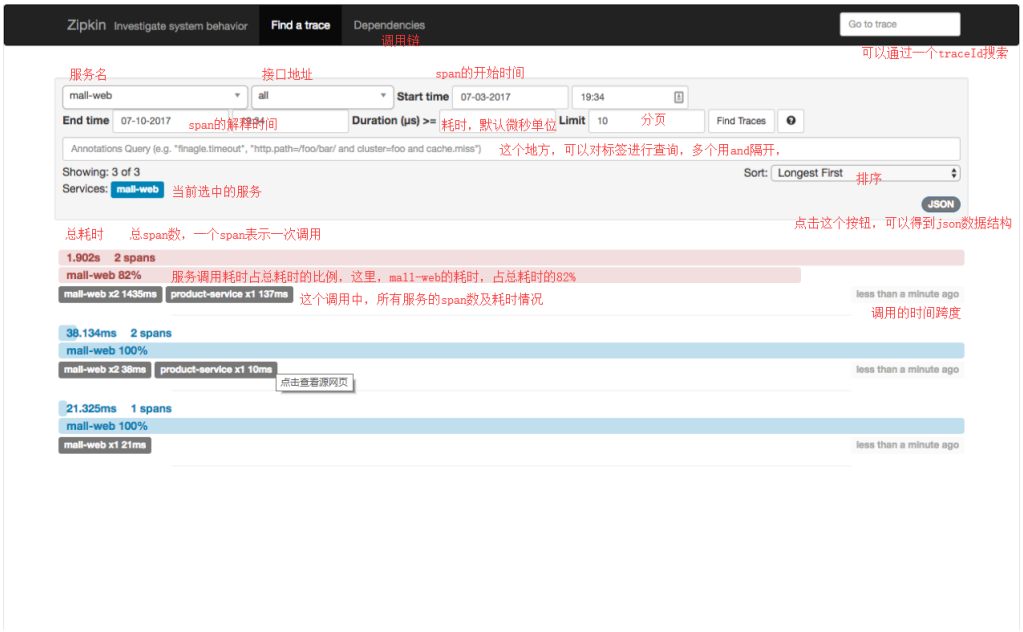
4.1 Zipkin

Zipkin 是由 Twitter 公司贡献开发的一款开源的分布式追踪系统，官方提供有基于 Finagle 框架（Scala 语言）的接口，而其他框架的接口由社区贡献，目前可以支持 Java、Python、Ruby 和 C# 等主流开发语言和框架，其主要功能是聚集来自各个异构系统的实时监控数据。主要由 4 个核心组件构成，如下图所示：



- **Collector：**收集器组件，它主要用于处理从外部系统发送过来的跟踪信息，将这些信息转换为 Zipkin 内部处理的 Span 格式，以支持后续的存储、分析、展示等功能。
- **Storage：**存储组件，它主要对处理收集器接收到的跟踪信息，默认会将这些信息存储起来，同时支持修改存储策略。
- **API：**API 组件，它主要用来提供外部访问接口，比如给客户端展示跟踪信息，或是外接系统访问以实现监控等。
- **UI：**UI 组件，基于 API 组件实现的上层应用，通过 UI 组件用户可以方便而有直观地查询和分析跟踪信息。

当用户发起一次调用的时候，Zipkin 的客户端会在入口处先记录这次请求相关的 trace 信息，然后在调用链路上传递 trace 信息并执行实际的业务流程，为防止追踪系统发送延迟与发送失败导致用户系统的延迟与中断，采用异步的方式发送 trace 信息给 Zipkin Collector，Zipkin Server 在收到 trace 信息后，将其存储起来。随后 Zipkin 的 Web UI 会通过 API 访问的方式从存储中将 trace 信息提取出来分析并展示。

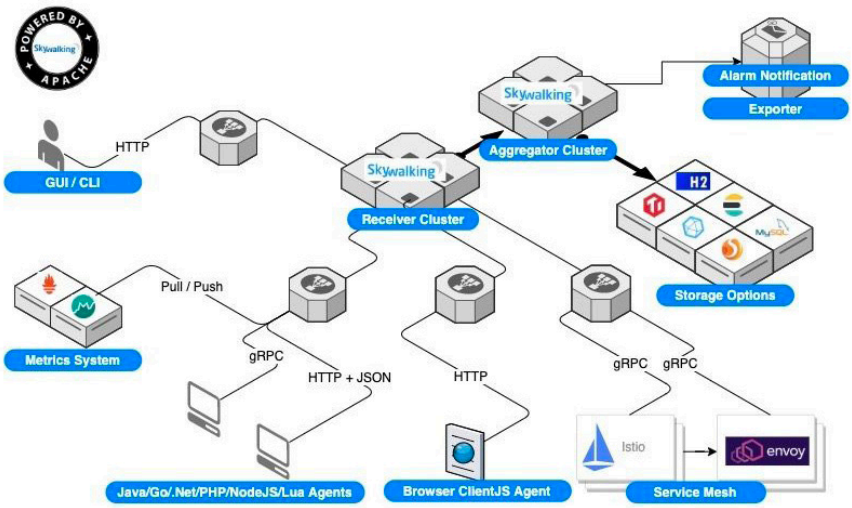


最后，我们看看 Zipkin 的跨线程传递方案的优缺点：在单个线程的调用中 Zipkin 通过定义一个 ThreadLocal<TraceContext> local 来完成在整个线程执行过程中获取相同的 Trace 值，但是当新起一个线程的时候 ThreadLocal 就会失效，对于这种场景，Zipkin 对于不提交线程池的场景提供 InheritableThreadLocal<TraceContext> 来解决父子线程 trace 信息传递丢失的问题。

而对于 @Async 的使用场景，Zipkin 提供 CurrentTraceContext 类首先获取父线程的 trace 信息，然后将 trace 信息复制到子线程来，其基本思路和上文 MTrace 的一致，但是需要代码开发，具有较强的侵入性。

4.2 SkyWalking

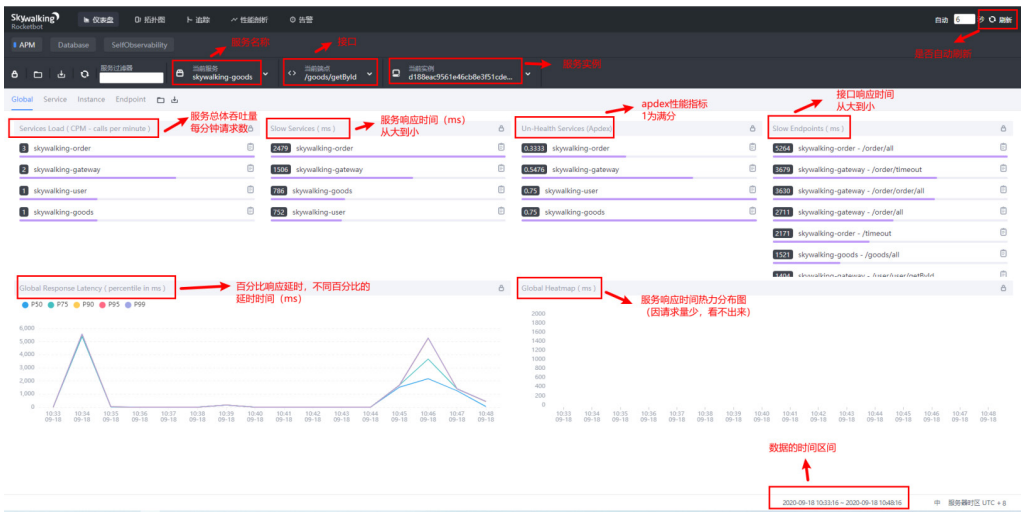
SkyWalking 是 Apache 基金会下面的一个开源的应用程序性能监控系统，提供了一种简便的方式来清晰地观测云原生和基于容器的分布式系统。具有支持多种语言探针；微内核 + 插件的架构；存储、集群管理和使用插件集合都可以自由选择；支持告警；优秀的可视化效果的特点。其主要由 4 个核心组件构成，如下图所示：



- **探针：**基于不同的来源可能是不一样的，但作用都是收集数据，将数据格式化为 SkyWalking 适用的格式。
- **平台后端：**支持数据聚合，数据分析以及驱动数据流从探针到用户界面的流程。分析包括 Skywalking 原生追踪和性能指标以及第三方来源，包括 Istio、Envoy telemetry、Zipkin 追踪格式化等。
- **存储：**通过开放的插件化的接口存放 SkyWalking 数据。用户可以选择一个既有的存储系统，如 ElasticSearch、H2 或 MySQL 集群（Sharding-Sphere 管理），也可以指定选择实现一个存储系统。
- **UI：**一个基于接口高度定制化的 Web 系统，用户可以可视化查看和管理 SkyWalking 数据。

SkyWalking 的工作原理和 Zipkin 类似，但是相比较于 Zipkin 接入系统的方式，SkyWalking 使用了插件化 +javaagent 的形式来实现：通过虚拟机提供的用于修改代码的接口来动态加入打点的代码，如通过 javaagent premain 来修改 Java 类，在系统运行时操作代码，让用户可以在不需要修改代码的情况下进行链路追踪，对业务的代码无侵入性，同时使用字节码操作技术（Byte-Buddy）和 AOP 概念来实现拦截追踪上下文的 trace 信息，这样一来每个用户只需要根据自己的需用定义拦截点，就可以实

现对一些模块实施分布式追踪。

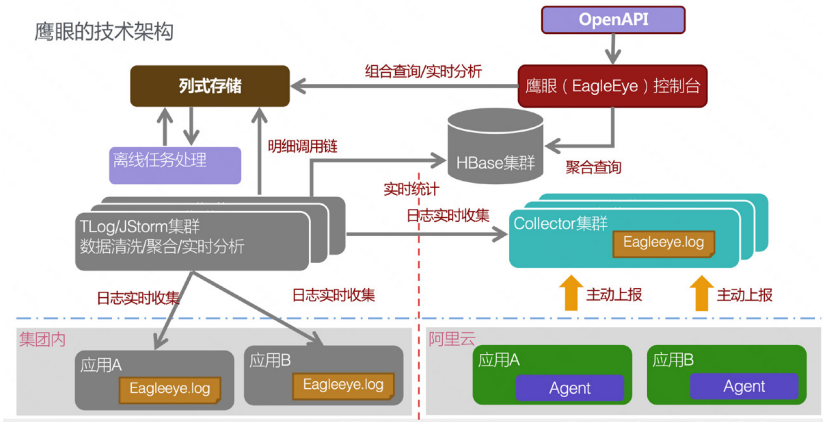


最后，我们总结一下 SkyWalking 的跨线程传递方案的优缺点：和主流的分布式追踪系统类似，SkyWalking 也是借助 ThreadLocal 来存储上下文信息，当遇到跨线程传输时也面临传递丢失的场景，针对这一问题 SkyWalking 会在父线程调用 ContextManager.capture() 将 trace 信息保存到一个 ContextSnapshot 的实例中并返回，ContextSnapshott 则被附加到任务对象的特定属性中，那么当子线程处理任务对象的时会先取出 ContextSnapshott 对象，将其作为入参调用 ContextManager.continued(contextSnapshot) 来保存到子线程中。

整体思路其实和主流的分布式追踪系统的相似，SkyWalking 目前只针对带有 @TraceCrossThread 注解的 Callable、Runnable 和 Supplier 这三种接口的实现类进行增强拦截，通过使用 xxxWrapper.of 的包装方式，避免开发者需要大的代码改动。

4.3 EagleEye

EagleEye 阿里巴巴开源的应用性能监控工具，提供了多维度、实时、自动化的应用性能监控和分析能力。它可以帮助开发人员实时监控应用程序的性能指标、日志、异常信息等，并提供相应的性能分析和报告，帮助开发人员快速定位和解决问题。主要由以下 5 部分组成：

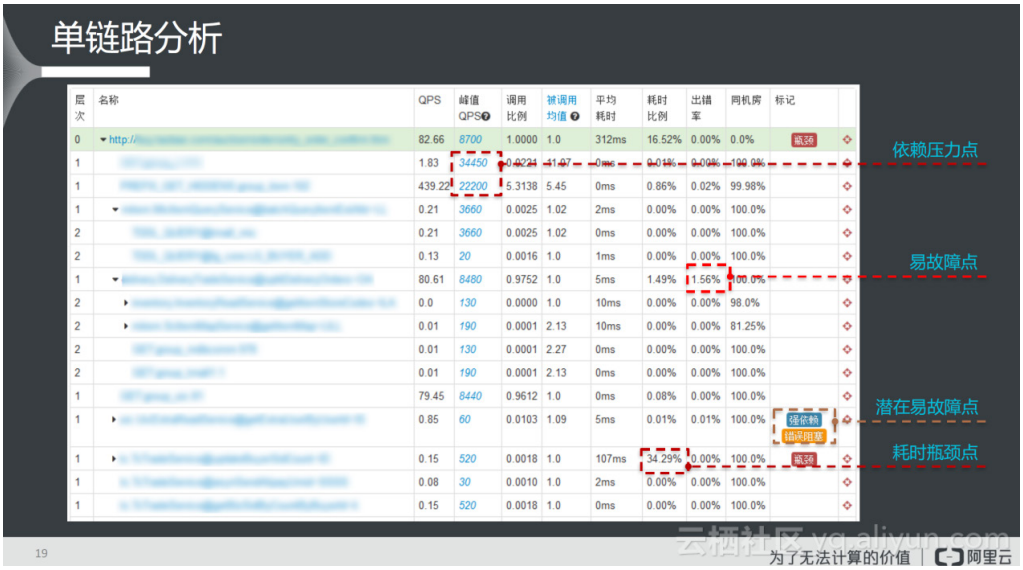


- **代理：**代理是鹰眼的数据采集组件，通过代理可以采集应用程序的性能指标、日志、异常信息等数据，并将其传输到鹰眼的存储和分析组件中。代理支持多种协议，如 HTTP、Dubbo、RocketMQ、Kafka 等，能够满足不同场景下的数据采集需求。
- **存储：**存储是鹰眼的数据存储组件，负责存储代理采集的数据，并提供高可用、高性能、高可靠的数据存储服务。存储支持多种存储引擎，如 HBase、Elasticsearch、TiDB 等，可以根据实际情况进行选择和配置。
- **分析：**分析是鹰眼的数据分析组件，负责对代理采集的数据进行实时分析和处理，并生成相应的监控指标和性能报告。分析支持多种分析引擎，如 Apache Flink、Apache Spark 等，可以根据实际情况进行选择和配置。
- **可视化：**可视化是鹰眼的数据展示组件，负责将分析产生的监控指标和性能报告以图形化的方式展示出来，以便用户能够直观地了解系统的运行状态和性能指标。
- **告警：**告警是鹰眼的告警组件，负责根据用户的配置进行异常检测和告警，及时发现和处理系统的异常情况，防止系统出现故障。

不同于 SkyWalking 的开源社区，EagleEye 重点面向阿里内部环境开发，针对海量实时监控的痛点，对底层的流计算、多维时序指标与交互体系等进行了大量优化，同时引入了时序检测、根因分析、业务链路特征等技术，将问题发现与定位由被动转为主动。

EagleEye 采用了 StreamLib 实时流式处理技术提升流计算性能，对采集的数据进行实时分析和处理，当监控一个电商网站时，可以实时地分析用户访问的日志数据，并根据分析结果来优化网站的性能和用户体验；参考 Apache Flink 的 Snapshot 优化齐全

度算法来保证监控系统确定性；为了满足不同的个性化需求，把一些可复用的逻辑变成了“积木块”，让用户按照自己的需求，拼装流计算的 pipeline。



最后总结一下 EagleEye 的跨线程传递方案优缺点：EagleEye 的解决思路和大多数分布式追踪系统一致，都是通过 javaagent 的方式修改线程池的实现，进而子线程可以获取到父线程到 trace 信息，不同于 SkyWalking 这种开源系统采用的字节码增强，EagleEye 大多数场景是内部使用，所以采用直接编码的方式，维护和性能消耗方面也是非常有优势的，但扩展性和开放性并不是非常友好。

5. 总结

本文意在从日常工作中一个很细微的问题出发，探究分析背后的设计思想和底层原因，主要涉及以下方面：

- **抓住问题本质：**在业务系统报警中抓住问题的核心代码并尝试再次复现问题，找到真正出问题的模块。
- **深入理解设计思想：**在查阅公司中间件的产品文档的基础上再继续追根溯源，学习业内领先者最开始的分布式链路追踪系统的设计思想和实现途径。
- **结合实际问题提出疑问：**结合了解到的分布式链路追踪系统的实现流程和设计思想，回归到一开始我们要解决的 TraceId 丢失情况分析是在什么环节出现问题。

- **阅读源码找到底层逻辑：**从 @Async 注解、SimpleAsyncTaskExecutor 和 ThreadLocal 类源码进行层层追踪，分析底层真正的实现逻辑和特点。
- **对比分析找到解决方案：**分析为什么 Mtrace 的跨线程传递方案“失效”了，找到原因提供解决方案并总结其他分布式追踪系统。

从本文可以看出，中间件的出现不仅为我们维护系统的稳定提供有力的支持，还已经为使用中可能发生的问题提供了更高效的解决方案，作为开发人员在享受这一极大便利的同时，还是要沉下心来认真思考其中的实现逻辑和使用场景，如果只是一味的低头使用不求甚解，那么在一些特定问题上往往会显得十分被动，无法发挥中间件真正的价值，甚至在没有中间件支撑时无法高效的解决问题。

本文作者

李祯，美团到店事业群 / 充电宝业务部工程师。

参考资料

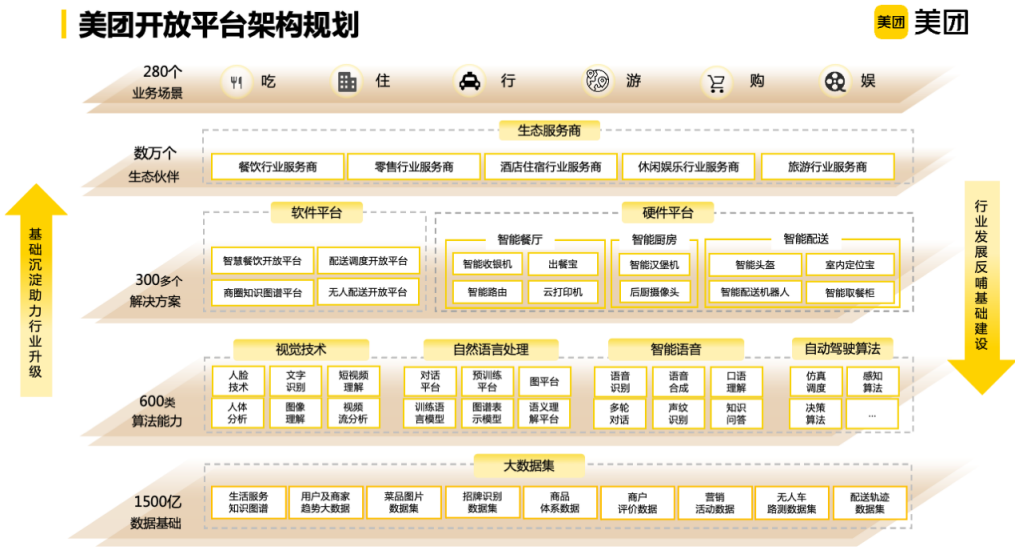
[1] Dapper, a Large-Scale Distributed Systems Tracing Infrastructure
[2] ThreadLocal
[3] Annotation Interface Async
[4] SkyWalking 8 官方中文文档
[5] Zipkin Architecture
[6] 阿里巴巴鹰眼技术解密

美团开放平台 SDK 自动生成技术与实践

作者：飞宏 照东 宇豪 王鸿

1. 引言

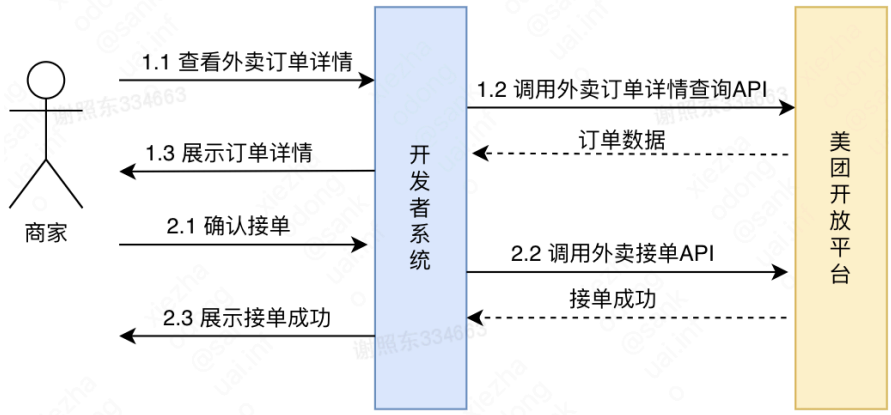
美团开放平台对外提供了外卖、团购、配送等 20 余个业务场景的 OpenAPI，供第三方开发者搭建应用时使用，是美团系统与外部系统通讯的最重要平台。本文主要讲述开放平台如何通过技术手段自动生成支持接口参数富模型和多种编程语言的 SDK，以提高开发者对接开放平台 API 的效率。



美团开放平台架构

1.1 背景

美团开放平台将美团各类业务提供的扩展服务封装成一系列应用程序编程接口（API）对外开放，供第三方开发者使用。开发者可通过调用开放平台提供的 OpenAPI 获取数据和能力，以实现自身系统与美团系统协同工作的业务逻辑。以外卖业务场景为例，开发者可以在自己为外卖商户开发的应用中通过调用美团开放平台提供的 API，提供外卖订单查询、接单、订单管理等一系列功能。如下图所示：



开放平台为开发者提供的 OpenAPI 以 HTTP 接口的形式提供。以平台提供的订单查询接口为例，对应的 HTTP 请求如下所示：

```
POST https://api-open-cater.meituan.com/api/order/queryById
Content-Type: application/x-www-form-urlencoded;charset=utf-8

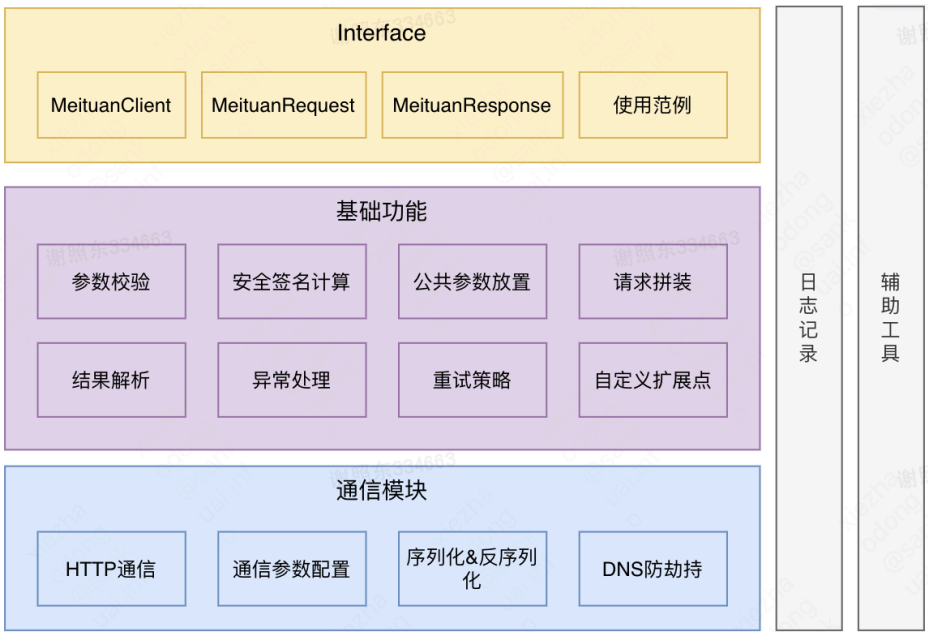
appAuthToken=eeee860a3d2a8b73cfb6604b136d6734283510c4e92282&
charset=utf-8&
developerId=106158&
sign=4656285a4c2493e279d929b8b9f4e29310da8b2b&
timestamp=1618543567&
biz={"orderId": "10046789912119"}

Response:{
  "orderId":"10046789912119",
  "payAmount":"45.67",
  "status":7,
  ......,
  "products":[{"pid":"8213","num":2,...,"price":"3.67"}
{"pid":"6556","num":1,...,"price":"11.99"}]
}
```

由上述示例可以看出，美团开放平台提供给开发者的接口契约较为复杂，其中包含了业务规则复杂及安全性要求高等原因。若开发者需要直接从 0 到 1 编码对接平台提供的 HTTP API，需要关注通信协议、接口契约规范、认证标识传递和安全签名等细节，成本较高。随着业务的发展，平台支持的 OpenAPI 数量在近两年增长约一倍，达到近 1000 个，平台运营和研发人员需要投入越来越多的精力去帮助开发者解决接口对接过程中的疑难问题。因此，提供 SDK 以帮助开发者提高开发对接效率，变得十分有必要。

1.2 SDK 目标概述

SDK，英文名称为 Software Development Kit，即软件开发工具包，广义上指辅助开发某一类软件的相关工具、文档和范例的集合。在开放平台的场景，我们为开发者提供的 SDK 应能为其屏蔽调用 OpenAPI 的通信协议、参数传递规范、接口基础契约（如时间戳、安全签名）等细节，以降低其对接平台 API 所需的开发成本。具备基本功能的开放平台 SDK 的架构和功能模块如下所示：



从使用 SDK 的开发者角度来看，基于 SDK 封装的基础功能来编写调用开放平台接口的代码，大致逻辑如下所示：

```
MeituanClient client = DefaultMeituanClient.builder(developerId, signKey).build();
// 设置请求参数
MeituanRequest request = new MeituanRequest("/api/order/queryById");
request.setParam("orderId","10046789912119");
MeituanResponse response = client.invokeApi(req);
if(response.isSuccess()) {
  long price = (long)response.getField("price");
  String phone = response.getField("customerPhone");
  int orderStatus = (int)response.getField("status");
  // 完成业务逻辑
} else {
  log.warn("query order failed with response={}", response);
  // 处理接口调用失败的逻辑
}
```

从上述代码可以看出，提供基础功能的 SDK 已经能够为使用者提供较大的便利。相比从零开始编码对接 OpenAPI，使用 SDK 可以帮助开发者省去处理通信协议、公共参数放置、安全签名计算和返回状态码解析的工作量。但开发者在编写代码设置 API 的业务参数字段的环节，仍需对照 API 文档逐个手工填充字段名并按字段类型赋值，并且在获取 API 返回的业务字段时也需自主填充字段名并解析数据类型，存在较大的不便且易出错。

为解决此问题，我们需要在 SDK 的能力上更进一步提供对参数富模型的支持，即为每个 API 提供模型化封装的请求参数和返回参数结构，让使用 SDK 的开发者可以更加专注于业务逻辑的开发。

在 SDK 加入参数富模型的支持后，从使用者的角度来看，需要编写的代码如下所示：

```
MeituanClient client = DefaultMeituanClient.builder(developerId, signKey).build();
// 设置请求参数
QueryOrderRequest request = new QueryOrderRequest();
request.setOrderId("10046789912119");
// 调用接口
MeituanResponse<QueryOrderResponse> response = client.invokeApi(req);
// 处理接口返回
if(response.isSuccess()) {
  QueryOrderResponse orderResponse = response.getData();
  long price = orderResponse.getPrice();
  String phone = orderResponse.getCustomerPhone();
  int orderStatus = orderResponse.getStatus();
  log.info("query order finish, price={}, orderStatus={}", price, phone,
orderStatus);
} else {
  log.warn("query order failed with response={}", response);
  // 处理接口调用失败的逻辑
}
```

可以看出，参数富模型功能可以进一步减少开发者使用 SDK 的复杂度。以 Java 语言版本为例，QueryOrderRequest 和 QueryOrderResponse 两个富模型类中封装了 API 的请求参数和返回参数的所有字段名、字段类型和字段校验规则等信息，开发者可简单使用字段的 getter 和 setter 方法完成对字段的赋值和取值操作，大幅降低了理解成本和出错可能。

虽然在 SDK 中支持参数富模型功能，可以有效提高使用者的效率，但也会带来 SDK 的开发和维护成本增加。如果采用纯人工的方式去开发维护 SDK 中支持的所有 API 的参数模型代码，需要投入的开发维护成本与 SDK 支持的编程语言数量和 API 数量呈正

相关性，其成本公式为：

$$Cost = F(API数量 * 支持编程语言数量)$$

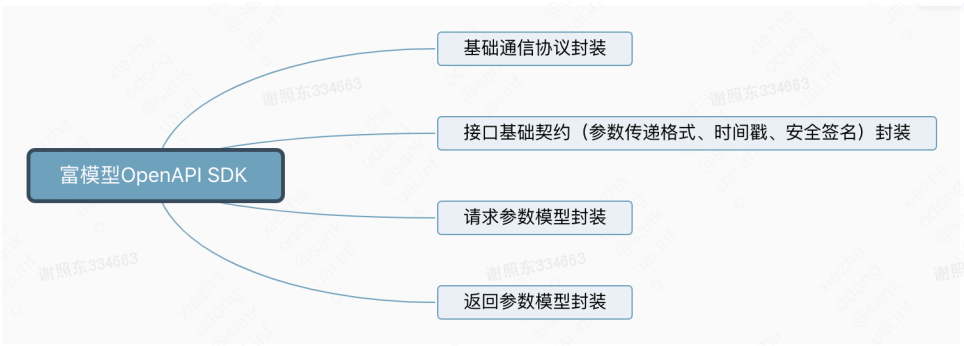
从上述公式可以看出，当 SDK 所需支持的 API 数量和编程语言数量达到一定数量时，通过纯人工编码去开发和维护 SDK 的成本会非常高。需要通过技术手段自动生成和测试 SDK 中的绝大部分代码，以达到在成本可控的前提下，为开发者提供支持多种编程语言版本的富模型 SDK 的目标。

2. SDK 自动生成技术详解

2.1 整体设计

要为开发者提供一个支持参数富模型功能的 OpenAPI SDK，我们需要实现以下主要功能：

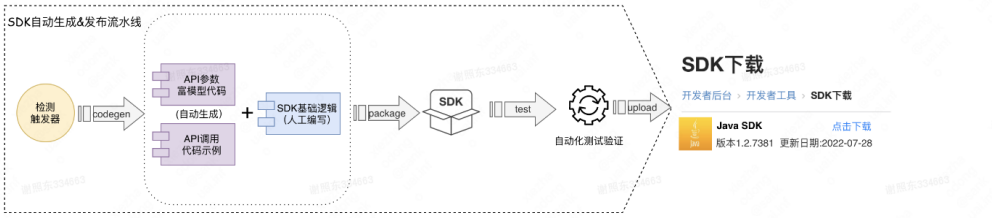
1. **通信协议封装：**让开发者无需关注调用 API 的通信协议和通信逻辑。
2. **接口基础契约封装：**让开发者无需关注调用 API 的参数传递格式、时间戳、安全签名、返回 Code 码处理等细节。
3. **请求参数模型封装：**让开发者便捷地设置 API 请求参数。
4. **返回参数模型封装：**让开发者便捷地使用 API 返回的数据。



其中，通信协议封装和接口基础契约封装是一次性工作，并且其逻辑是相对稳定的。对于 SDK 所需支持的每一种编程语言，只需投入有限的成本开发一次对应代码逻辑，即可支撑 SDK 的整个生命周期。而要为平台开放的 1000 余个 API 提供支持多种编程语

言的参数富模型功能，靠人工编写和维护代码是极其低效的，我们考虑通过代码自动生成技术，对 SDK 中的参数富模型代码进行自动化生成。

更进一步，在实现了参数富模型代码自动生成后，我们可以通过持续集成（Continious Integration）和持续发布（Continuous Delivery）技术，将 SDK 的生成、测试和发布流程也尽可能地做到自动化。整体的 SDK 自动生成流程设计如下图所示：



实现了以上流程后，即可做到在开放平台的任意 API 的参数模型发生变化时，由系统自动生成和发布最新版本的 SDK 供开发者使用。我们将在下文详述如何通过代码自动生成、持续集成和持续发布等技术手段实现上述流程。

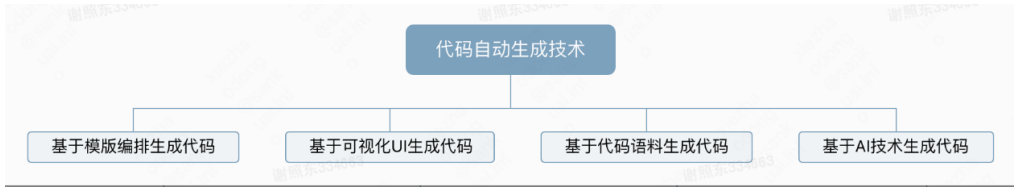
2.2 自动生成参数模型代码

我们最终的目标是为开放平台的每个 OpenAPI，自动生成供 SDK 使用的请求参数模型代码（Request 类）、返回参数模型代码（Response 类）和调用示例代码（Example），并且代码自动生成机制要支持 SDK 适配的多种编程语言。以 Java 和 C# 编程语言为例，我们要生成的目标代码如下图所示：



从上面的示例中可以看出，在请求参数模型（Request 类）中需要生成 Request Path、鉴权配置、字段强类型定义、字段取值、赋值及校验逻辑等代码。在返回参数模型（Response 类）中，需要生成接口返回的各个数据字段的强类型定义、取值逻辑及校验规则。调用示例代码则需要包含请求参数赋值、发起接口调用和处理接口返回数据等相关逻辑。

要达成上述目标，首先需要考虑的是代码自动生成技术的选型，目前业界主流的代码生成技术分为以下几类：

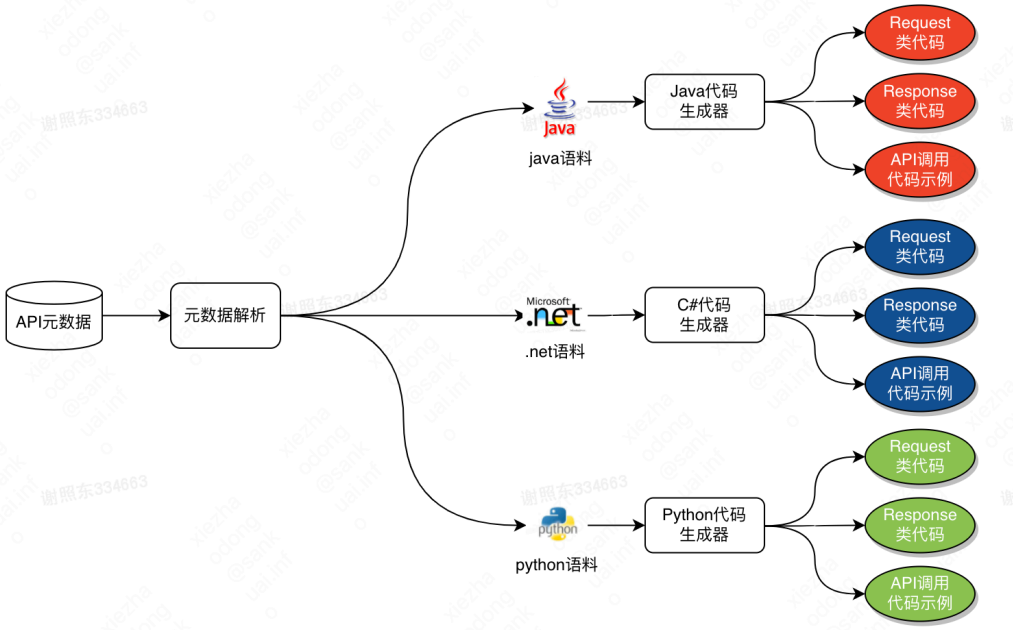


1. **基于模版编排生成代码：**最原始最简单也是目前应用最广泛的一种代码生成方式。包括后端 MVC 框架的 Controller、Service、DAO 层模式化代码一键生成，还有前端 Vue CLI 和 Create-React-App 两款脚手架的代码生成，都属于此类。

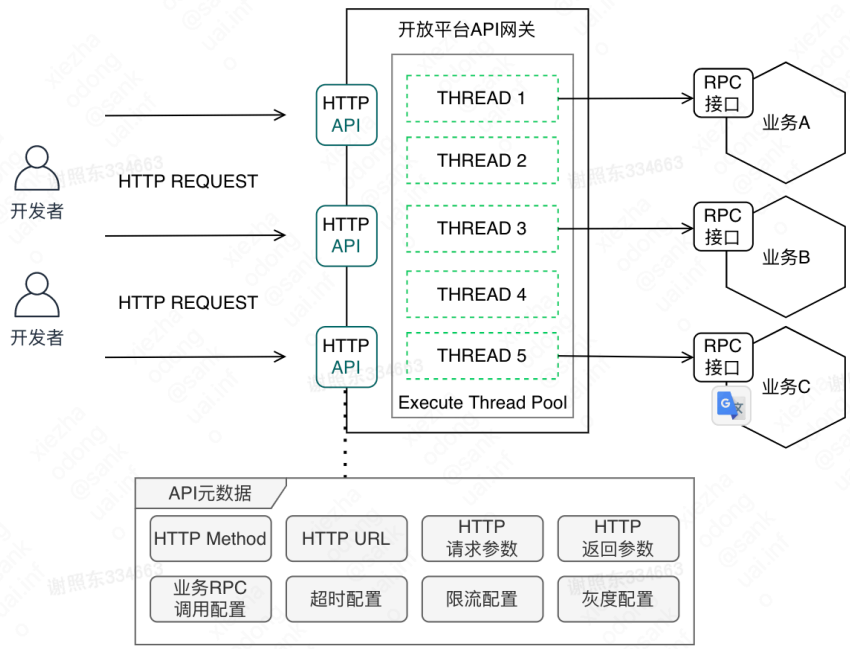
2. **基于可视化 UI 生成代码：**目前市场上运用得很广的一门技术，也被称为代码可视化生成工具。从 Eclipse 的 Web 可视化编辑器，到 .NET Framework 提供的 MVC，及 Winform 界面及控件代码可视化拖拽生成，到汽车行业广泛使用的可视化原型搭建工具（自动生成 C 代码）都属于此类。在近几年比较火的低代码平台（如 aPaaS）中，通过可视化 UI 生成代码的技术也被大量使用。
3. **基于代码语料生成代码：**基于代码语料生产代码的前提是要有足够的语料，例如伪代码 / 中间语言 / 描述性代码模板，再基于一套生成规则去生成目标代码。常见的落地场景包括 RPC 框架中基于 IDL（Interface description language，接口描述语言）自动生成多种编程语言的 RPC Client 和 Service 代码，以及 IDE 插件中的代码自动生成功能（例如 Eclipse 的 telosys 插件可通过 DSL 生成多种语言代码）。
4. **基于人工智能技术生成代码：**属于比较前沿的技术范畴，多和 AI 领域的图像识别和机器学习技术结合。现有的一些典型案例包括：微软开发的可将手绘图转化 HTML 代码的智能化代码生成工具 sketch2code，基于 AI 技术自动生成 UI 逻辑的 teleporthq。

考虑到开放平台 SDK 中，需要自动生成的 OpenAPI 参数富模型代码和调用示例代码均具备相对较强的规则性和模式性，我们选择基于代码语料自动生成代码的技术路线。

基于代码语料自动生成代码需要“语料”+“规则”两个核心元素，我们可以通过解析 API 元数据并结合领域专用语言（DSL）作为语料模板，生成代码语料，再基于语料特性为不同的编程语言定制代码生成规则，最终将“语料”+“规则”输入代码生成器以完成目标代码的生成。整体流程如下图所示：



在上述流程中，首先关注作为代码语料生成数据源的 API 元数据，其来源于开放平台实现的零编码 API 网关底层维护的基础配置。开放平台网关基于 API 元数据配置化的技术，可做到零编码将业务服务的 RPC 接口转化为 HTTP 协议的 API 进行开放。其基本运行结构如下图所示：



作为驱动开放平台网关运行的核心数据，API 元数据中包含了 HTTP Method、URL、请求参数、返回参数等信息。在参数信息中，又以树形结构记录了每个参数字段的字段名、字段类型、字段描述、校验规则和示例值。我们以“按订单 id 查询订单详情”的 API 为例，其元数据中和 SDK 生成相关的数据如下所示：

```
APIGroup:waimai
APISubGroup:order
APIName: order_query_by_id
HTTP METHOD: POST
HTTP PATH: /api/order/queryById
Description: 按订单 id 查询订单详情
Request
  |- orderId LONG NOT_NULL 要查询的订单的 id example:1000224201796844308
Response
  |- orderId  LONG NOT_NULL 订单 id  example:1000224201796844308
  |- price  LONG NOT_NULL 订单金额（单位为人民币 " 分 "）example:3308
  |- phone  STRING  顾客联系电话  example:"13000000002"
  |- products  ARRAY<Product>  订单商品列表
     |- pid  LONG  商品 id  example:"13000000002"
     |- name  String  商品名  example:" 珍珠奶茶 "
     |- num  INTEGER  商品数量  example:1
     |- price  LONG  商品单价  example:1199
     |- properties  ARRAY<Property>  商品属性列表
        |- name STRING 商品属性名  example:" 甜度 "
        |- value STRING 商品属性值  example:" 七分糖 "
     |- remark  STRING  商品备注  example:" 请做常温的 "
  |- status  INTEGER  订单状态  example:7
```

以上信息足以支撑我们为 SDK 生成参数富模型和调用示例代码。下一步我们需要开始处理代码语料，并为最终的代码自动化生成做好准备。不同编程语言所需的代码语料有所差异，但同一类编程语言（如 Java 和 C# 都是面向对象的编程语言）大致相同。

以生成 Java SDK 中的参数富模型代码为例，需要用到的代码语料包含两部分。第一部分为类的基本信息，由元数据解析器在解析 API 的元数据时生成，其包含的内容和具体生成方式如下表所示：

数据类型	说明	生成规则	示例
packageName	Java类所属的包名	{固定前缀}.{APIGroup}.{APISubGroup}.{APIName}	com.meituan.sdk.model.waimai.order.OrderQueryById
className	类名	{ApiName} + Request/Response	OrderQueryByIdRequest
baseClass	基类名	暂无	-
classDesc	类描述	/*{Description}*/	/*按订单id查询订单详情*/
staticFields	静态字段信息列表	暂无	-
privateFields	私有字段信息列表	从元数据的树形结构参数信息中递归解析。树中的叶子结点均为基本类型（如int、string）的字段，直接生成字段信息。而非叶子结点均为复合类型（如array、结构体）字段，需要做递归处理，再生成一个Java类作为字段类型	orderId LONG NOT_NULL 订单id example:1000224201796844308 products ARRAY<Product> 订单商品列表
importPackages	需要import的依赖包列表	在递归解析参数过程中，按字段类型记录需依赖的包信息	java.util.List，javax.validation.constraints.NotNull
interfaces	类实现的接口列表	MeituanRequest<{Response类名}>	MeituanRequest<OrderQueryByIdResponse>
JavaApiMeta	类注解中需要的元数据信息	从元数据解析结果中中直接获取	path=/api/order/queryById , needAuth=false

第二部分为语料模板，我们以 DSL（Domain Specific Language）作为中间语言加以描述，如下所示：

```
<@class className=className metaInfo=javaApiMeta baseClass=baseClass
interfaces=interfaces
classDesc=classDesc package=packageName importPackages=importPackages>
    <#-- 静态字段   -->
    <#if staticFields?? && (staticFields?size > 0) >
        <#list staticFields as param>
            <@staticField param=param/>
        </#list>
    </#if>
    <#-- 字段   -->
    <#if privateFields?? && (privateFields?size > 0) >
        <#list privateFields as param>
            <@field param=param/>
        </#list>
    </#if>
   <#-- Getter/Setter -->
    <#if privateFields?? && (privateFields?size > 0) >
        <#list privateFields as param>
            <@getterMethod param=param/>
            <@setterMethod param=param/>
        </#list>
    </#if>

    <#-- 静态字段 Getter -->
    <#if staticFields?? && (staticFields?size > 0) >
        <#list staticFields as param>
            <@getterMethod param=param/>
        </#list>
    </#if>

    <#if javaApiMeta?has_content>
        <@deserializeResponse metaInfo=javaApiMeta/>
        <@serializeToJson metaInfo=javaApiMeta/>
    </#if>
```

```
    <#-- toString 方法 -->
    <#if privateFields?? && (privateFields?size > 0) >
        <@toString className=className params=privateFields/>
    </#if>
</@class>
```

有了上述的代码语料，我们即可通过语言转换引擎生成 Java 代码。我们将解析好的 API 元数据作为输入，执行基于 DSL 的语言转换引擎。语言转换引擎通过执行宏命令将要生成的代码类的基本信息在 DSL 语料模板中进行填充，最终得到 Java 编程语言的目标类及其附属类的代码。以生成 Response 类代码为例，代码生成的具体执行过程如下图所示：



Request 和 Response 类中其余的 `getter` 方法、`setter` 方法、类注解等元素的生成原理和步骤均和以上相同，此处不再赘述。在 DSL 语料模板中所有的元素处理完成后，我们即可得到供 Java 编程语言使用的请求参数类和返回参数类的完整代码。

对于其他的编程语言（例如 Python），我们使用的 API 元数据和元数据解析逻辑和 Java 是一致的，不同点在于 DSL 语料模板和语言转换引擎。当需要对 SDK 新增一种编程语言的支持时，我们只需要对目标语言建立 DSL 语料模板并提供相应的转换逻辑，即可支持该语言的请求参数类和返回参数类的代码自动生成。

2.3 自动生成 API 调用示例代码

通过同样的技术手段，我们还可以自动生成每个 OpenAPI 的调用示例代码，并将示例代码展示接口文档中供开发者参考。

调用示例代码的生成的逻辑相对参数模型代码更加简单。我们使用 API 元数据中的类名和字段信息（元数据中也包含了每个字段的 examle 值，可用于在代码示例中生成字

段赋值的逻辑）填入代码语料中，再执行语言转换引擎生成目标代码即可。以 Java 编程语言为例，用于生成 API 调用示例代码的 DSL 语料模板如下所示：

```
<#setting number_format="computer">
MeituanClient meituanClient = DefaultMeituanClient.builder(10000L, "xxxxx").
build();

<#assign reqVarName = className?uncap_first/>
${className} ${reqVarName} = new ${className}();

<#if privateFields?? && (privateFields?size > 0)>
<#list privateFields as field>
${reqVarName}.set${field.fieldName?cap_first}(${field.exampleValue!""});
</#list>
</#if>

<#if javaApiMeta.needAuth>
String appAuthToken = "xxxx";
MeituanResponse<${javaApiMeta.responseClass}> response = meituanClient.
invokeApi(request,
appAuthToken);
<#else >
MeituanResponse<${javaApiMeta.responseClass}> response = meituanClient.
invokeApi(request);
</#if>

if (response.isSuccess()) {
<#if javaApiMeta.responseClass == "Void">
    System.out.println(" 调用成功 ");
<#else>
    ${javaApiMeta.responseClass} resp = response.getData();
    System.out.println(resp);
</#if>
} else {
    System.out.println(" 调用失败 ");
}
```

在使用 API 元数据和代码语料模板执行基于 DSL 的语言转换引擎后，生成的 API 调用示例代码如下所示：

```
MeituanClient client = DefaultMeituanClient.builder(developerId, signKey).build();
// 设置请求参数
OrderQueryByIdRequest request = new OrderQueryByIdRequest();
request.setOrderId(100022420179684308L);
// 调用接口
MeituanResponse<OrderQueryByIdResponse> response = client.invokeApi(req);
// 处理接口返回
if(response.isSuccess()) {
  OrderQueryByIdResponse orderResponse = response.getData();
  System.out.println(orderResponse);
```

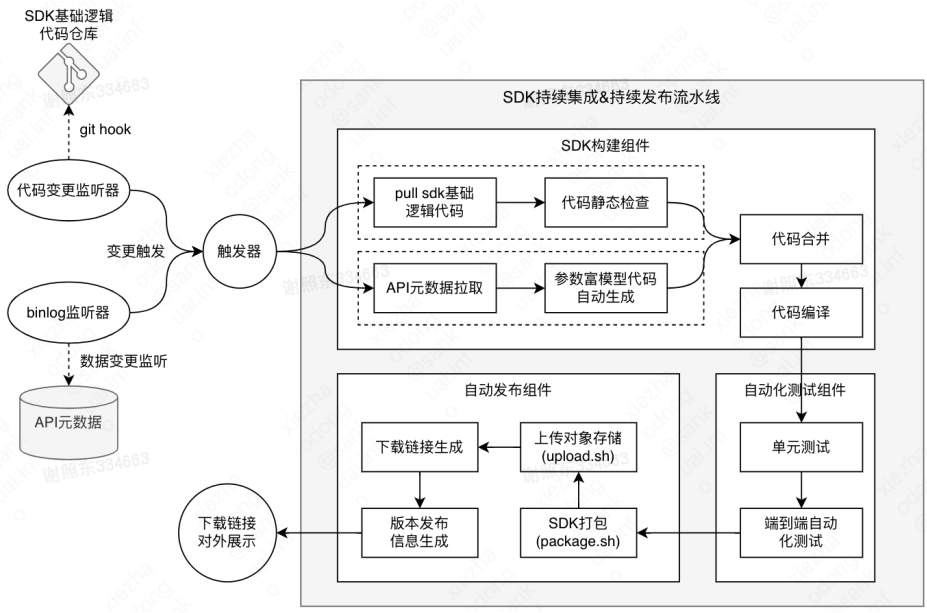
```
} else {
  System.out.println(" 调用失败 ");
}
```

可以看出，我们生成的 API 调用示例代码可以为开发者呈现出每个请求参数赋值的示例逻辑，可有效降低开发者在对接 API 时的理解成本。后续我们可以进一步优化 DSL 语料模板，在示例代码中增加对返回数据结构中各个字段的取值逻辑示范，以进一步降低开发者在处理 API 返回数据时的理解和开发成本。

2.4 持续集成和持续发布

搞定参数富模型代码和调用示例代码的自动生成后，下一步是通过持续集成和持续发布技术，确保开发者在任何时刻均能获取到最新版本的 SDK。传统由人工编译、测试和上传发布 SDK 的模式，开发者得到 SDK 版本更新的周期短则数周，长则数月。我们的目标是将这个周期缩短到分钟级别：当 SDK 的基础逻辑和 API 参数模型有任何变更发生时，通过持续集成和持续发布的能力，在数分钟内将包含此变更的新版本 SDK 发布给开发者使用。

我们基于美团自研的流水线引擎来驱动 SDK 的持续集成和持续发布。流水线的执行可以看作是对生成 SDK 的“原材料”一步步加工，最终交付到线上的过程。先通过下图了解整体流程：



首先我们监听可能导致 SDK 需要发布的变更，包括通过 Binlog 机制监听 API 元数据的变更，以及通过 Git Hook 机制监听 SDK 基础逻辑代码仓库 Master 分支的变更。一旦监听到有变更产生，通过触发器去触发 SDK 持续集成和发布流水线的运作。

流水线开始运作后，首先执行 SDK 构建组件，SDK 构建组件会并发执行两个操作：

1. 获取 SDK 基础逻辑代码（人工编写）并完成静态代码检查；
2. 拉取 API 元数据并自动生成参数富模型代码。

以上两个操作完成后，执行代码合并和代码编译，将结果提交到流水线执行下一个步骤。接下来由自动化测试组件完成对 SDK 的单元测试和端到端自动化测试，通过后提交到流水线执行下一个步骤。最后由自动发布组件完成 SDK 的打包、上传、下载链接生成和版本信息生成等一系列操作，并最终将最新版本 SDK 发布到官网供开发者下载。

3. 结语

通过上述能力的建设，我们打通了 SDK 自动生成的整个环节，以自动化的方式完成代码生成、构建、测试、集成、发布等一系列行为，最终实现了在低人力投入的前提下持续向开发者交付最新版本 SDK 的目标。

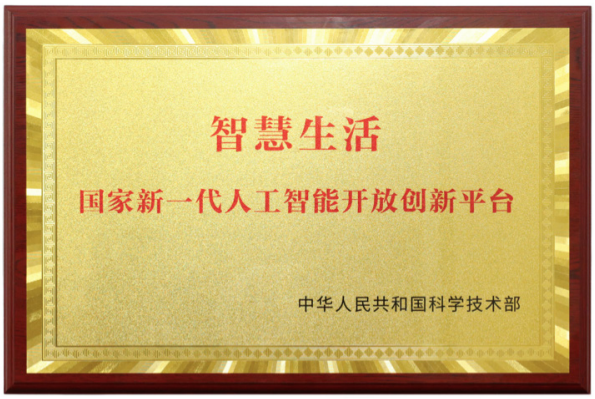
通过最近半年数据的对比，我们可以看出开发者使用 SDK 后在接口对接环节遇到的疑难问题明显减少。基本达到了我们最初提高开发者接入效率，降低平台研发和运营处理工单成本的目标。



后续，我们将会计划继续完善 SDK 的代码自动生成逻辑，并为 SDK 添加更多编程语言的支持，为接入美团开放平台的开发者提供更好的体验。

4. 写在后面

不久前，美团独立申报的智慧生活国家新一代人工智能开放创新平台正式获得中华人民共和国科学技术部（以下简称“科技部”）批复。这是美团第一个国家级科研平台。



国家新一代人工智能开放创新平台被称为“人工智能国家队”，是聚焦人工智能重点细分领域，充分发挥行业领军企业的引领示范作用，有效整合技术资源、产业链资源和金融资源，持续输出人工智能核心研发能力和服务能力的重要创新载体。此前，已有百度、阿里、腾讯等 15 家公司先后获批建设。本次美团成功申报，标志着美团的科研创新能力获得了国家层面认可，达到“国家队水平”。

5. 本文作者

飞宏、照东、宇豪、王鸿等，均来自美团到店事业群 / 餐饮 SaaS 事业部。

MJDK 如何实现压缩速率的 5 倍提升?

作者：艳梅

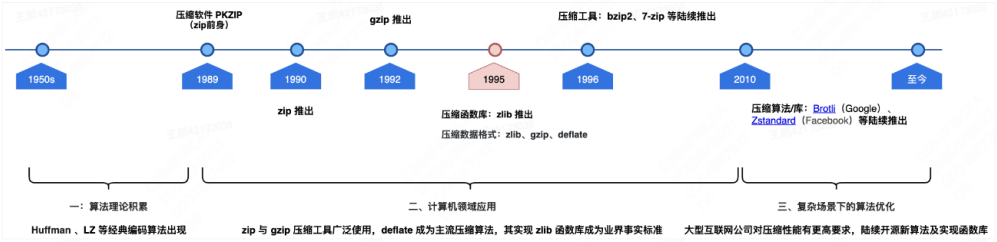
1. 前言

数据压缩技术 [1] 因可有效降低数据存储及传输成本，在计算机领域有非常广泛的应用（包括网络传输、文件传输、数据库、操作系统等场景）。主流压缩技术按其原理可划分为无损压缩 [2]、有损压缩 [3] 两类，工作中我们最常用的压缩工具 zip 和 gzip ，压缩函数库 zlib，都是无损压缩技术的应用。Java 应用中对压缩库的使用包括：处理 HTTP 请求时对 body 的压缩 / 解压缩操作、使用消息队列服务时对大消息体（如 >1M）的压缩 / 解压缩、数据库写入前及读取后对大字段的压缩 / 解压缩操作等。常见于监控、广告等涉及大数据传输 / 存储的业务场景。

美团基础研发平台曾经开发过一种基于 Intel 的 isa-l 库优化的 gzip 压缩工具及 zlib[4] 压缩库（又称：mzlib[5] 库），优化后的压缩速度可提升 10 倍，解压缩速度能提升 2 倍，并已在镜像分发、图片处理等场景长期稳定使用。遗憾的是，受限于 JDK[6] 对压缩库调用的底层设计，公司 Java8 服务一直无法使用优化后的 mzlib 库，也无法享受压缩 / 解压缩速率提升带来的收益。为了充分发挥 mzlib 的性能优势为业务赋能，在 MJDK 的最新版本中，我们改造并集成了 mzlib 库，完成了 JDK 中 java.util.zip.* 原生类库的优化，可实现在保障 API 及压缩格式兼容性的前提下，将内存数据压缩速率提升 5-10 倍的效果。本文主要介绍该特性的技术原理，希望相关的经验给大家带来一些启发或者帮助。

2. 数据压缩技术

计算机领域的数据压缩技术的发展大致可分为以下三个阶段：



详细时间节点如下：

- 20 世纪 50 ~ 80 年代，香农创立信息论，为数据压缩技术奠定了理论基础。期间出现多种经典算法，如 Huffman 编码、LZ 系列编码等。
- 1989 年，Phil Katz 推出文件归档软件 PKZIP（zip 前身），并公开文件归档格式 zip 及其使用的数据压缩算法 deflate（Huffman 与 LZ77 的组合算法）的所有技术参数。
- 1990 年，Info-ZIP 小组基于公开的 deflate 算法编写了可移植的、免费的、开源实现 zip 和 unzip，极大地扩展了 .zip 格式的使用。
- 1992 年，Info-ZIP 小组基于 zip 的 deflate 算法代码，推出了文件压缩工具 gzip（GUN zip），用于替代 Unix 下的 compress（有专利纠纷）。通常 gzip 会与归档工具 tar 结合使用来生成压缩的归档格式，文件扩展名为 .tar.gz。
- 1995 年，Info-ZIP 小组成员 Jean-loup Gailly 和 Mark Adler 基于 gzip 源码中的 deflate 算法实现，推出了压缩库：zlib 。通过库函数调用的方式，为其他场景（如 PNG 压缩）提供通用的压缩 / 解压缩能力。同年，在 RFC 中发布了 DEFLATE、ZLIB、GZIP 三种数据压缩格式。其中 DEFLATE 是原始压缩数据流格式，ZLIB、GZIP 则是在前者的基础上包装数据头及校验逻辑等。此后随着 zip、gzip 工具及 zlib 库的广泛应用，DEFLATE 成为互联网时代数据压缩格式的事实标准。
- 2010 年后，各大型互联网公司陆续开源了新的压缩算法及实现，如：LZFSE（Apple）、Brotli（Google）、Zstandard（Facebook）等，在压缩速度和压缩比方面均有不同程度的提升。常见的压缩库如下（需要注意的是：由于压缩算法协议的差异，这些函数库不能交叉使用，数据压缩 / 解压缩必须使用同一种算法操作）：

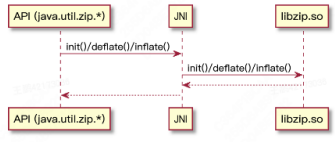
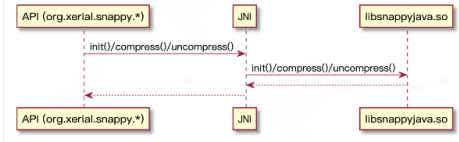
	压缩库	推出时间	一句话描述	支持的压缩数据格式	开发语言
1	zlib	1995年	zlib 是基于 deflate 算法实现的，一套完全开源、通用的无损数据压缩库。 **特点**：事实的业界标准，应用最广泛。	RFC：DEFLATE RFC：ZLIB RFC：GZIP	C
2	snappy	2011年	Google基于LZ77实现的压缩库，广泛应用于 Google 孵化项目，包括BigTable、MapReduce等。 **特点**：压缩速度高，压缩率低。	snappy	C++
3	LZ4	2011年	基于LZ算法实现的无损数据压缩库，提供单核500MB/s的压缩速度及GB/s的解压速度。 **特点**：压缩速度高，压缩率略低于 gzip	LZ4	C
4	Zopfli	2013年	Google 基于 deflate 实现的压缩库。 **特点**：压缩率高，压缩速度慢	RFC：DEFLATE RFC：ZLIB RFC：GZIP	C
5	Brotli	2015年	Google 使用 LZ77算法变体、Huffman编码和二阶上下文建模的组合算法实现的压缩库。 **特点**：压缩速度与 deflate 基本持平，但压缩率更高	RFC：Brotli	C
6	Zstandard	2016年	Facebook 基于Huffman 编码和 FSE 实现的快速压缩库。**特点**：压缩速度快	RFC：Zstandard	C

3. 压缩技术在 Java 中的应用及优化思路

前面我们介绍了压缩技术的基础知识，本章节主要介绍 MJDK8_mzlib 版本实现压缩速率 5 倍提升的技术原理。分两部分进行阐述：第一部分，介绍原生 JDK 中压缩 / 解压缩 API 的底层原理；第二部分，分享 MJDK 的优化思路。

3.1 Java 语言中压缩 / 解压缩 API 实现原理

Java 语言中，我们可以使用 JDK 原生压缩类库（java.util.zip.*）或第三方 Jar 包提供的压缩类库两种方式来实现数据压缩 / 解压缩，其底层原理是通过 JNI (Java Native Interface) 机制，调用 JDK 源码或第三方 Jar 包中提供的共享库函数。详细对比如下：

对比项	JDK 原生压缩类库（zlib 压缩库）	第三方压缩类库（此处以 Google 推出的snappy 压缩库举例，其他第三方类库原理基本类似）
执行原理		
共享库路径	jre/lib/amd64/libzip.so	jar 包下 native/linux/x86/libsnappyjava.so
压缩算法	deflate 算法（基于LZ77 + 霍夫曼编码的变种算法，目前应用最广泛的无损压缩算法，也是业界事实标准）	snappy 算法（基于LZ77的变种算法，广泛应用于 Google 内部项目，包括BigTable、MapReduce等）
支持的压缩数据格式	• RFC：DEFLATE • RFC：ZLIB • RFC：GZIP	• snappy（不同的压缩算法对应不同的压缩数据格式）

其中在使用方式上，两者区别可参考如下代码。

（1）JDK 原生压缩类库（zlib 压缩库）

zip 文件压缩 / 解压缩代码 demo（Java）

```
public class ZipUtil {
      //压缩
    public void compress(File file, File zipFile) {
        byte[] buffer = new byte[1024];
        try {
            InputStream     input  = new FileInputStream(file);
            ZipOutputStream zipOut = new ZipOutputStream(new
FileOutputStream(zipFile));
            zipOut.putNextEntry(new ZipEntry(file.getName()));
            int length = 0;
            while ((length = input.read(buffer)) != -1) {
                zipOut.write(buffer, 0, length);
            }
            input.close();
            zipOut.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

  //解压缩
    public void uncompress(File file, File outFile) {
        byte[] buffer = new byte[1024];
        try {
            ZipInputStream input  = new ZipInputStream(new
FileInputStream(file));
            OutputStream   output = new FileOutputStream(outFile);
            if (!outFile.getParentFile().exists()) {
                outFile.getParentFile().mkdir();
            }
            if (!outFile.exists()) {
                outFile.createNewFile();
            }

            int length = 0;
            while ((length = input.read(buffer)) != -1) {
                output.write(buffer, 0, length);
            }
            input.close();
            output.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

gzip 文件压缩 / 解压缩代码 demo（Java）

```
public class GZipUtil {
    public void compress(File file, File outFile) {
        byte[] buffer = new byte[1024];
        try {
            InputStream      input  = new FileInputStream(file);
            GZIPOutputStream gzip   = new GZIPOutputStream(new
FileOutputStream(outFile));
            int              length = 0;
            while ((length = input.read(buffer)) != -1) {
                gzip.write(buffer, 0, length);
            }
            input.close();
            gzip.finish();
            gzip.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void uncompress(File file, File outFile) {
        try {
            FileOutputStream out    = new FileOutputStream(outFile);
            GZIPInputStream  ungzip = new GZIPInputStream(new
FileInputStream(file));
            byte[]           buffer = new byte[1024];
            int              n;
            while ((n = ungzip.read(buffer)) > 0) {
                out.write(buffer, 0, n);
            }
            ungzip.close();
            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

（2）第三方压缩类库（此处以 Google 推出的 snappy 压缩库举例，其他第三方类库原理基本类似）分成两步。

第一步：pom 文件中添加依赖 Jar 包（C 语言）

```
        <dependency>
            <groupId>org.xerial.snappy</groupId>
            <artifactId>snappy-java</artifactId>
            <version>1.1.8.4</version>
        </dependency>
```

第二步：第二步，调用接口进行压缩 / 解压缩操作（C 语言）

```
public class SnappyDemo {
    public static void main(String[] args) {
        String input = "Hello snappy-java! Snappy-java is a JNI-based
wrapper of "
                + "Snappy, a fast compresser/decompresser.";
        byte[] compressed = new byte[0];
        try {
            compressed = Snappy.compress(input.getBytes("UTF-8"));
            byte[] uncompressed = Snappy.uncompress(compressed);
            String result = new String(uncompressed, "UTF-8");
            System.out.println(result);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

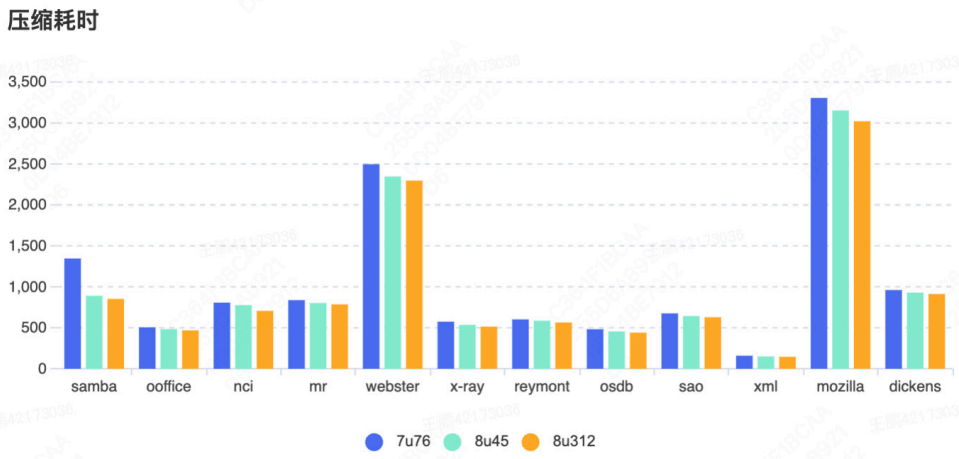
综上所述，JDK 中默认使用的压缩库是 zlib，虽然业务可以通过第三方 Jar 包的方式使用其他的压缩库算法，但是因为 Snappy 等算法的压缩数据格式与 zlib 支持的 DEFLATE、ZLIB、GZIP 不同，混合使用会有兼容性问题。

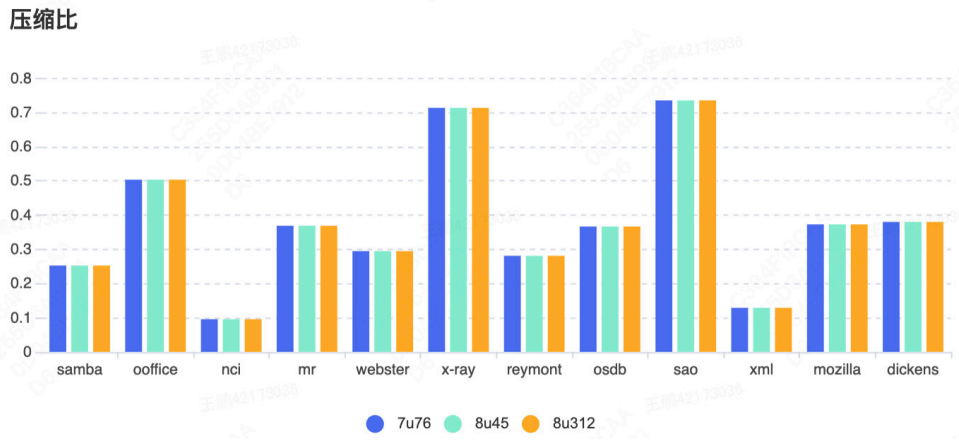
除此之外，zlib 库（1995 年推出）本身的迭代速度非常缓慢（原因：应用范围广且稳定、无商业组织维护），这里使用测试集 Silesia corpus 测试了 OpenJDK 7u76（2014 年发行）、8u45（2015 年发行）、8u312（2022 年发行）中内置压缩类库的性能，从图表中可看出，三者在压缩耗时、压缩比两方面均未有明显的优化效果，难以满足业务日益增长的压缩性能需求场景。因此，我们选择在 MJDK 中集成 zlib 优化，实现既兼容原生接口实现，又能提升压缩性能的效果。

Silesia corpus 是压缩方法性能基准测试集，提供一套涵盖现时使用的典型资料类别的档案资料。文件的大小在 6 MB 到 51 MB 之间，文件格式包括 text、exe、html、picture、database、bin data 等。测试数据类别如下：

Filename	Description	Type	Source	Raw size [B]	Bzipped size [B]
dickens	Collected works of Charles Dickens	English text	Project Gutenberg	10,192,446	2,799,528
mozilla	Tarred executables of Mozilla 1.0 (Tru64 UNIX edition)	exe	Mozilla Project	51,220,480	17,914,392
mr	Medical magnetic resonanse image	picture	Hospital image	9,970,564	2,441,280
nci	Chemical database of structures	database	CACTVS Chemical Information Services at LMC/NCI	33,553,445	1,812,734
ooffice	A dll from Open Office.org 1.01	exe	Open Office	6,152,192	2,862,526
osdb	Sample database in MySQL format from Open Source Database Benchmark	database	Open Source Database Benchmark Project	10,085,684	2,802,792
reymont	Text of the book Ch�opi by W�adys�aw Reymont	Polish pdf	Virtual Library of Polish Literature	6,627,202	1,246,230
samba	Tarred source code of Samba 2-2.3	src	Samba Project	21,606,400	4,549,790
sao	The SAO star catalog	bin data	Astronomical Catalogs and Catalog Formats	7,251,944	4,940,524
webster	The 1913 Webster Unabridged Dictionary	html	Project Gutenberg	41,458,703	8,644,714
xml	Collected XML files	html	XMLPPM: XML-Conscious PPM Compression	5,345,280	441,186
x-ray	X-ray medical picture		Hospital image	8,474,240	4,051,112
Total				211,938,580	54,506,808

Silesia corpus 测试集说明





3.2 MJDK 优化方案

通过 3.1 章节，我们知道 Java 原生的 java.util.zip.* 类库中的数据压缩 / 解压缩能力最终是调用 zlib 库实现的，因此 JDK 的压缩性能提升问题就可转换为对 JDK 使用的 zlib 库的优化。

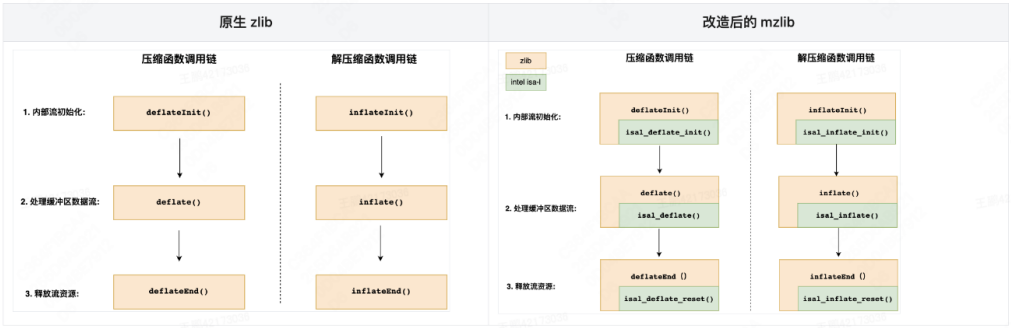
3.2.1 优化思路

除原生 zlib 外，同样使用 deflate 算法的压缩库有 Intel ISA-L、Intel IPP、Zopfli，直接基于 zlib 源码优化的项目有 zlib-cloudflare，它们与 zlib 间的对比如下：

压缩库	性能表现（与 zlib 相比）	兼容性表现（与 zlib 相比）
ISA-L	压缩性能是 zlib 的 5-10 倍，解压缩性能是 zlib 的 2-3 倍	• 仅压缩数据格式兼容
IPP	压缩速度可提升至原始版本的 2 倍左右	• API 兼容 • 压缩数据格式兼容
zlib-cloudflare	压缩速度可提升至原始版本的 2 倍左右	• API 兼容 • 压缩数据格式兼容
Zopfli	最大压缩级别下，压缩率可降低 3%-8%，但压缩速度可能降低几十倍	• 仅压缩数据格式兼容

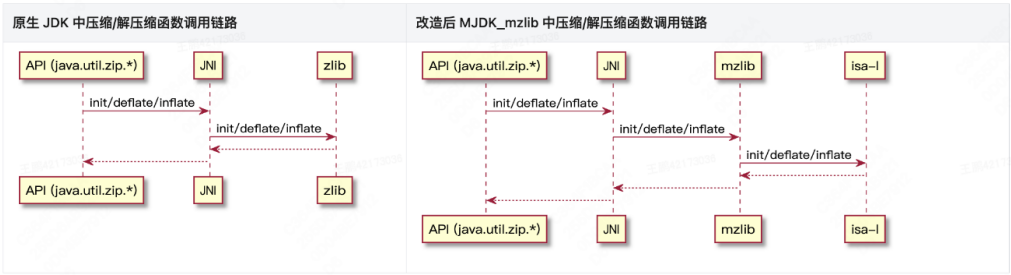
综上，我们选择基于 Intel 开源的 ISA-L（原理是使用 intel sse/avx/avx2/avx256 的扩展指令，并行运算多个流来提升底层函数的执行性能）来完成 zlib 的改造优化。

1. zlib 改造流程（重点在 API 的兼容性改造）



优化后的 mzlib 库在线上稳定运行 3 年以上，压缩速率提升在 5 倍以上，有效解决了上文提到基础研发平台曾在镜像构建、图片处理等场景面临过压缩 / 解压缩耗时较高的问题。

2. JDK 层面变更



3.2.2 优化效果

测试说明

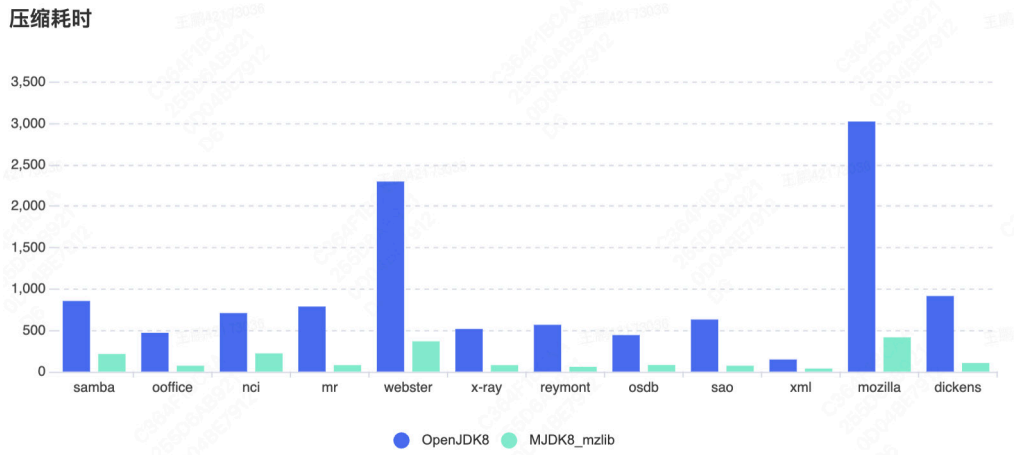
测试集：Silesia corpus

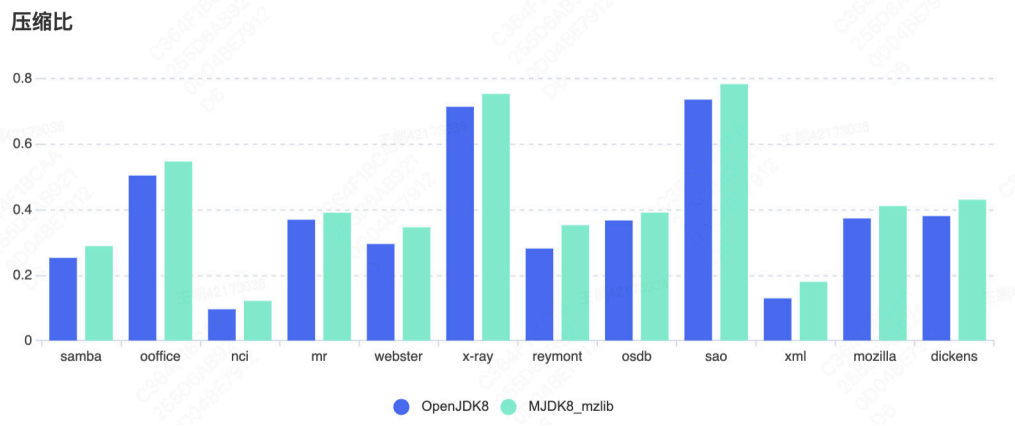
测试内容：GZip 压缩 / 解压缩文件、Zip 压缩 / 解压缩文件

测试结论

兼容性测试（通过）：改造后的 Java 类库的 Zip、Gzip 压缩 / 解压缩接口可正常使用，与原生 JDK 中的接口交叉进行压缩 / 解压缩操作验证通过。

性能测试（通过）：在同一基准 update 版本下，MJDK8_mzlib 数据压缩耗时比 OpenJDK8 降低 5–10 倍，压缩比无较大波动（增加 3% 左右）。





目前，美团内部的文档协同服务已使用该 MJDK 版本，进行用户协同编辑记录数据（> 6M）的压缩存储，验证了该功能在线上的稳定运行，压缩性能提升在 5 倍以上。

4 本文作者

艳梅，来自美团基础研发平台。

5 参考文献

[1] Comparison of Brotli, Deflate, Zopfli, LZMA, LZHAM and Bzip2 Compression Algorithms
[2] zip、gzip、zlib 的区别

注释

[1] 数据压缩技术：在不丢失有用信息的前提下，通过相应的算法缩减信源数据冗余，从而提高数据存储、传输和处理效率的技术。

[2] 无损压缩：利用数据的统计冗余进行压缩，常见的无损压缩编码方法有 Huffman 编码，算术编码，LZ 编码（字典压缩）等。数据统计冗余度的理论限制为 2:1 到 5:1，所以无损压缩的压缩比一般比较低。这类方法广泛应用于文本数据、程序等需要精确存储数据的压缩，

[3] 有损压缩：利用了人类视觉、听觉对图像、声音中的某些频率成分不敏感的特性，允许压缩的过程中损失一定的信息，以此换来更大的压缩比。广泛应用于语音、图像和视频数据的压缩。

[4] zlib：zlib 是基于 DEFLATE 算法实现的，一套完全开源、通用的无损数据压缩库。也是目前应用最广泛的压缩库。在网络传输、操作系统、图像处理等领域均有大量使用。比如：
Linux kernel：使用 zlib 以实作网路协定的压缩、档案系统的压缩以及开机时解压缩自身的核心。
libpng一 ：用于 PNG 图形格式的一个实现，对 bitmap 数据规定了 DEFLATE 作为流压缩方法。
HTTP 协议：使用 zlib 对 HTTP 响应头数据进行压缩 / 解压缩。
OpenSSH、OpenSSL：以 zlib 达到最佳化加密网路传输。
Subversion、Git 和 CVS 等版本控制系统，使用 zlib 来压缩和远端仓库的通讯流量。
dpkg 和 RPM 等包管理软件：以 zlib 解压缩 RPM 或者其他封包。

[5] mzlib：美团基于 Intel 的 isa-l 库优化的 zlib 压缩库。

[6] JDK：Java Development Kit，是 Sun 公司针对 Java 开发人员发布的免费软件开发工具包，是 Java 开发的核心组件之一，包含了 Java 编译器、Java 虚拟机、Java 类库等开发工具和资源。

[7] JNI (Java Native Interface)：JNI 是一个本地编程接口。它允许在 Java 虚拟机中运行的 Java 代码与用其他编程语言（如 C、C++ 和汇编）编写的应用程序和库进行互操作。

基于 AI+ 数据驱动的慢查询索引推荐

作者：彭淦

1. 背景

随着美团业务量的不断增长，慢查询的数量也日益增多。目前，日均慢查询数量已经超过上亿条，如果仅依靠 DBA 和开发人员手动地对这些慢查询进行分析并建立合适的索引，显然是不太现实的。为了解决这一难题，美团内部 DAS（数据库自治服务）平台已经集成了基于代价的慢查询优化建议来自动地为慢查询推荐索引。然而，仍然存在一些问题：

- 基于代价的慢查询优化建议是借助于优化器的代价估计，来推荐出对于查询代价改善最大的索引，但优化器的代价估计并不是完全准确 [1]，因此可能存在着漏选或者错选推荐索引的问题。
- 基于代价的慢查询优化建议需要计算查询在不同索引下查询代价的改善程度，因此需要进行大量的增删索引操作，但真实增删索引的代价是非常大的，需要借助于假索引 [2] 技术，假索引技术并不创建真实的物理索引文件，只是通过模拟索引存在时的查询计划来估算索引对于查询的收益。目前，美团大部分业务都是运行在 MySQL 实例上的，不同于商业数据库 SQL Server 和开源数据库 PostgreSQL，MySQL 内部并没有集成假索引技术，因此需要自己构建支持假索引的存储引擎，其开发成本较高，这也是目前 DAS 平台基于代价的慢查询优化建议所采用的方案。

为了解决上述两个问题，美团数据库研发中心与华东师范大学数据科学与工程学院展开了《基于数据驱动的索引推荐》的科研合作，双方通过在 DAS 平台上集成基于 AI+ 数据驱动的索引推荐，来与基于代价的方法并行地为慢查询推荐索引，以提升推荐效果。

- 首先，基于代价的方法每天会为慢查询推荐索引，并在采样库上评估推荐的索引是否真正地改善了查询的执行时间，这为 AI 方法积累了大量可信的训练数据，根据此数据训练的 AI 模型，可以在一定程度上弥补基于代价的方法漏选或错选索引的问题。

- 其次，基于 AI 的方法将针对慢查询的索引推荐看作是二分类问题，通过分类模型直接判别在某一列或某些列上建立索引是否能够改善查询的执行性能，并不借助于查询优化器和假索引技术，这使得 AI 方法更加通用，且开发成本更低。

2. 索引推荐介绍

索引推荐可以划分为两个级别：Workload 级别和 Query 级别：

- 在 Workload 级别，索引推荐是在限制的索引存储空间或索引个数下，推荐出一组最优的索引集合来使得整个 Workload 的代价最低。
- Query 级别的索引推荐可以被视为 Workload 级别索引推荐的简化版本，在 Query 级别，索引推荐是为单个慢查询推荐缺失的索引，以改善其性能。

2.1 基于代价的索引推荐

基于代价的索引推荐 [3] 大多聚焦于 Workload 级别的索引推荐，出现在查询中每一列或者列的组合都可以看作是一个能够改善 Workload 代价的候选索引，所有的候选索引构成了一个巨大的搜索空间（候选索引集合）。

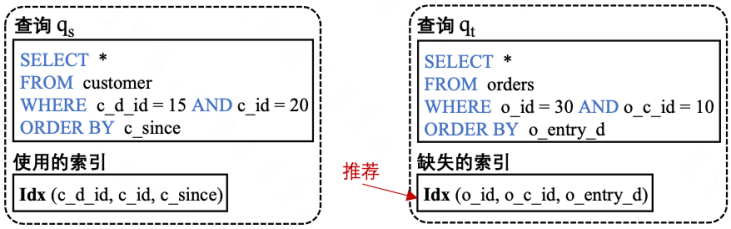
基于代价的索引推荐的目标，是在候选索引集合中搜索出一组最优索引集合，以最大程度地改善 Workload 代价。如果候选索引的个数 N，限制的最大推荐索引个数是 M，那么最优索引集合的搜索空间是：

$$C_N^M = \frac{N*(N-1)\dots(N-M+1)}{M!}$$

这是一个属于 NP-hard 范畴的搜索问题 [4]。目前，基于代价的索引推荐方法大多会采用“贪心策略”来简化搜索过程，但这可能会导致最后推荐出的索引是次优解 [5]。

2.2 基于 AI+ 数据驱动的索引推荐

基于 AI+ 数据驱动的索引推荐聚焦于 Query 级别的索引推荐，出发点是在某个数据库中因为缺失索引导致的慢查询，在其它数据库中可能有相似的索引创建案例：这些查询语句相似，因此在相似位置上的列创建索引也可能带来类似的收益。例如下图中，查询 q_s 和 q_t 在语句结构和列类型上非常相似。因此，我们可以通过学习查询 q_s 的索引创建模式来为查询 q_t 推荐缺失的索引。

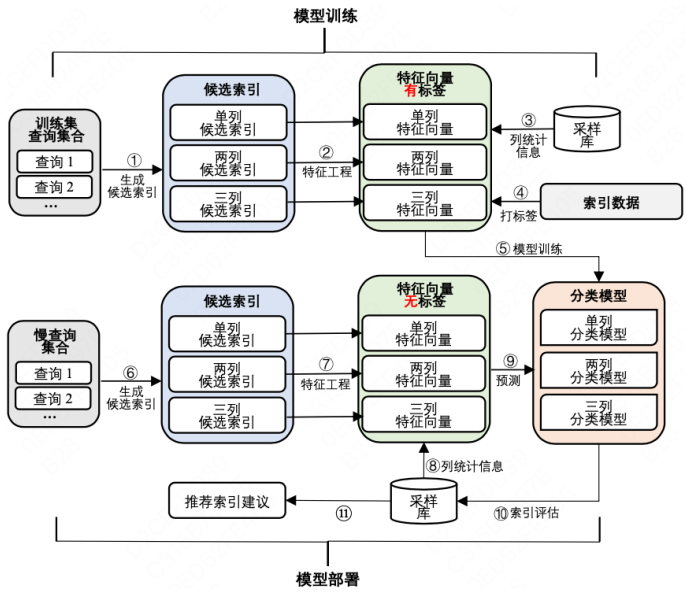


对于不同列数的索引推荐，我们会分别训练基于 XGBoost 的二分类模型。例如，我们目前最高支持三列的索引推荐，因此会分别训练一个单列索引推荐模型、一个两列索引推荐模型和一个三列索引推荐模型。对于给定的一个单列候选索引和它对应的慢查询，我们使用单列索引推荐模型来判断该单列候选索引是否能够改善该慢查询的性能。

同样的，对于给定的一个两列（三列）候选索引和它对应的慢查询，我们使用两列（三列）索引推荐模型来判断这个两列（三列）候选索引是否能够改善该慢查询的性能。如果一条慢查询中包含的候选索引个数为 N，那么则需要 N 次模型预测来完成对这条慢查询的索引推荐。

3. 整体架构

基于 AI+ 数据驱动的索引推荐的整体架构如下图所示，主要分为两个部分：模型训练和模型部署。



3.1 模型训练

如上文所述，我们收集 DAS 平台基于代价的慢查询优化建议每天的索引推荐数据（包括慢查询和被验证有效的推荐索引）作为训练数据。我们生成每条查询的单列、两列和三列候选索引，并通过特征工程来为每个候选索引构建特征向量，使用索引数据来为特征向量打标签。之后，单列、两列和三列特征向量将分别用于训练单列、两列和三列索引推荐模型。

3.2 模型部署

针对需要推荐索引的慢查询，我们同样生成候选索引并构建特征向量。接下来，我们使用分类模型来预测特征向量的标签，即预测出候选索引中的有效索引。随后，我们在采样库上创建模型预测出的有效索引，并通过实际执行查询来观察建立索引前后查询性能是否得到改善。只有当查询性能真正得到改善时，我们才会将索引推荐给用户。

4. 建模过程

4.1 生成候选索引

我们提取查询中出现在聚合函数、WHERE、JOIN、ORDER BY、GROUP BY 这些关键词中的列作为单列候选索引，并对这些单列候选索引进行排列组合来生成两列和三列候选索引。同时，我们会获取查询所涉及的表中已经存在的索引，并将其从候选索引集合中删除。这一步骤遵循索引的最左前缀原则：如果存在索引 $Idx(col1,col2)$，那么候选索引 $(col1)$ 和 $(col1,col2)$ 都将从候选索引集合中删除。

4.2 特征工程

一个候选索引的特征向量包括语句特征和统计特征两部分。语句特征描述了候选索引列在查询中的出现位置（采用 one-hot 的编码方式），统计特征描述了候选索引列的统计信息，如所在表的表行数、Cardinality 值、选择率等，这些是判断是否需要在候选索引列上建立索引的重要指标。

下表以单列候选索引 $(col1)$ 为例，展示了它的部分重要特征及其含义：

语义特征	含义		统计特征	含义
join	col1是join列		Cardinality	col1列中不同值的个数
equal	col1出现在等值查询条件中		Nullnumber	col1列中空值的个数
range	col1出现在范围查询条件中		Selectivity	col1列的选择率
orderby	col1是ORDER BY条件后的列		Tablecount	col1列所在表的表行数
...				

两列候选索引 $(col1,col2)$ 的特征是通过对单列候选索引 $(col1)$ 和 $(col2)$ 的特征进行拼接而成的，此外，我们还会计算 $col1$ 和 $col2$ 共同的 Cardinality 值作为两列候选索引 $(col1,col2)$ 的额外统计特征，以更加全面地描述其统计信息。同样地，我们也会采用使用这种方式来构建三列候选索引 $(col1,col2,col3)$ 的特征。在生成完一条查询的特征向量之后，我们使用这条查询使用到的索引来为生成的特征向量打标签。

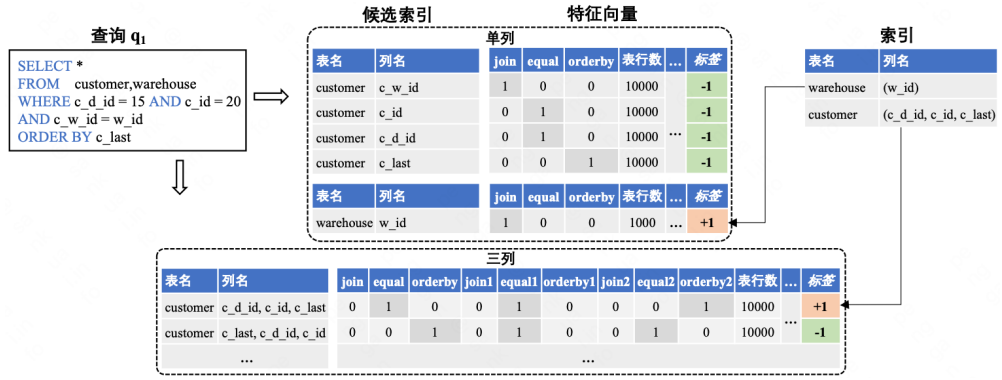
4.3 建模举例

下图以查询 q_1 为例，展示我们为训练集中的一条查询生成特征向量并打标签的过程。查询 q_1 涉及两张表 customer 表和 warehouse 表，其中 customer 表的 c_w_id、c_id、c_d_id、c_last 四列参与到查询中，因此对应生成四条单列特征向量；warehouse 表的 w_id 列参与到查询中，因此只生成了一条单列特征向量。查询 q_1 使用的单列索引为 Idx(w_id)，所以单列候选索引 (w_id) 对应的特征向量被标记为正样本，其余特征向量则被标记为负样本。

接下来，我们对单列候选索引进行排列组合来生成多列候选索引及其特征向量。由于查询 q_1 使用到的多列索引只有一个三列索引 Idx(c_d_id, c_id, c_last)，因此我们跳过生成两列候选索引，只生成三列候选索引。这是因为我们是基于查询使用到的索引来为特征向量打标签的，如果查询没有使用到两列索引，那么生成的所有两列特征向量均为负样本，这可能会导致训练集正负样本不均衡的问题。

最后，基于查询使用到的三列索引，我们将三列候选索引 (c_d_id, c_id, c_last) 对应的特征向量标记为正样本。以上就是我们为查询 q_1 生成特征向量并打标签的整个过程，查询 q_1 为单列索引推荐模型的训练集贡献了五条样本（一条正样本，四条负样本），为

三列索引推荐模型的训练集贡献了六条样本（一条正样本，五条负样本）。



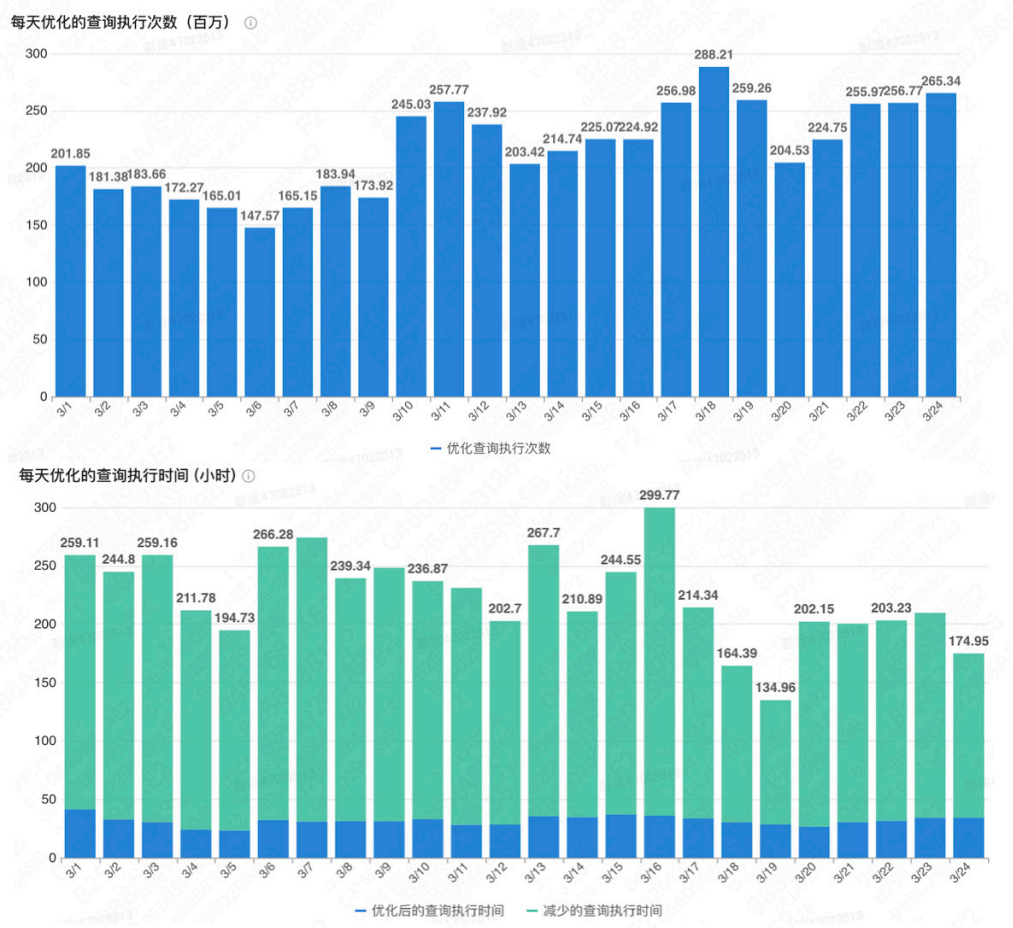
4.4 模型预测和索引评估

在为一条慢查询推荐索引时，我们依次生成慢查询中所有的单列、双列和三列候选索引，并通过上述的特征工程来构造特征向量。然后，我们将特征向量输入给对应的分类模型进行预测，并从三个分类模型的预测结果中分别挑选出一个预测概率最高的候选索引（即一个单列索引、一个两列索引和一个三列索引）作为模型推荐的索引。

虽然推荐的索引越多，慢查询的性能就越有可能得到改善，但是模型推荐的部分索引可能是无效的，这些无效索引带来的存储空间开销和更新索引的开销是不可忽视的。因此，直接将模型推荐的索引全部推荐给用户是不合理的。为此，在将索引推荐给用户之前，我们会首先将三个分类模型推荐的索引建立在采样库上进行验证，采样库是线上数据库的一个 mini 版本，它抽取了线上数据库的部分数据。在采样库上，我们会观察在建立推荐的索引之后，查询的执行时间是否得到改善。如果得到改善，我们就把查询使用到的一个或多个模型推荐的索引作为索引建议推荐给用户。

5. 项目运行情况

正如前文所述，美团 DAS 平台目前采用代价方法和 AI 模型并行为慢查询推荐索引。具体来说，AI 模型可以在某些场景下，弥补代价方法漏选或错选推荐索引的问题。就在刚过去的 3 月份，在代价方法推荐索引的基础上，AI 模型有额外 12.16% 的推荐索引被用户所采纳。



这些额外补充的索引对于查询的改善情况如上图所示：上半部分展示了优化的查询执行次数，下半部分展示了查询在使用推荐的索引之后的执行时间以及减少的执行时间，这些索引总计约优化了 52 亿次的查询执行，减少了 4632 小时的执行时间。

6. 未来规划

目前，大模型技术（如 GPT-4）已经得到了越来越多的认可，几乎可以胜任各种领域的任务。我们计划尝试通过 Fine-Tune 开源的大型语言模型（如 Google 开源的 T5 模型）来解决索引推荐的问题：输入一条慢查询，让模型来生成针对慢查询的索引建议。

在推荐索引无法改善慢查询的情况下，后续我们可以提供一些文本建议来帮助用户优化 SQL，比如减少返回不必要的列，使用 JOIN 代替子查询等。

7. 本文作者

彭淦，美团基础研发平台工程师，主要负责美团数据库自治服务 DAS 的 SQL 优化建议工作。

8. 特别感谢

在这里特别感谢华东师范大学数据科学与工程学院的蔡鹏教授，教授在 VLDB、ICDE、SIGIR、ACL 等领域重要国际会议上发表多篇论文。目前的研究方向为内存事务处理，以及基于机器学习技术的自适应数据管理系统。本文也是美团数据库研发中心跟蔡鹏教授展开科研合作后的具体实践。

美团科研合作致力于搭建美团技术团队与高校、科研机构、智库的合作桥梁和平台，依托美团丰富的业务场景、数据资源和真实的产业问题，开放创新，汇聚向上的力量，围绕机器人、人工智能、大数据、物联网、无人驾驶、运筹优化等领域，共同探索前沿科技和产业焦点宏观问题，促进产学研合作交流和成果转化，推动优秀人才培养。面向未来，我们期待能与更多高校和科研院所的老师和同学们进行合作。欢迎老师和同学们发送邮件至：meituan.oi@meituan.com。

9. 参考文献

[1] Leis V, Gubichev A, Mirchev A, et al. 2015. How good are query optimizers, really? Proc. VLDB Endow. 9, 3 (2015), 204–215.

[2] https://github.com/HypoPG/hypopg

[3] Kossmann J, Halfpap S, Jankrift M, et al. 2020. Magic mirror in my hand, which is the best in the land? an experimental evaluation of index selection algorithms. Proc. VLDB Endow. 13,12 (2020), 2382–2395.

[4] Piatetsky-Shapiro G. 1983. The optimal selection of secondary indices is NP-complete. SIGMOD Record. 13,2 (1983), 72–75.

[5] Zhou X, Liu L, Li W, et al. 2022. Autoindex: An incremental index management system for dynamic workloads. In ICDE. 2196–2208.

美团视觉 GPU 推理服务部署架构优化实践

作者：张旭 赵铮 岸青 林园 志良 楚怡等

0. 导读

美团视觉面向本地生活服务，在众多场景上落地应用了文字识别、图像质量评价、视频理解等视觉 AI 技术。此前，在线推理服务使用的 GPU 资源不断增加，但服务 GPU 利用率普遍较低，浪费大量计算资源，增加了视觉 AI 应用成本，这是美团也是很多企业亟需解决的问题。

美团视觉智能部通过实验分析发现，造成视觉推理服务 GPU 利用率低下的一个重要原因是模型结构问题：模型中预处理或者后处理部分 CPU 运算速度慢，导致推理主干网络无法充分发挥 GPU 运算性能。基于此，视觉研发团队通过模型结构拆分和微服务化，提出一种通用高效的部署架构，解决这种常见的性能瓶颈问题。

目前，该解决方案已经在多个核心服务上成功应用。以“图像检测 + 分类”服务为例，优化后的服务压测性能指标 GPU 利用率由 40% 提升至 100%，QPS 提升超过 3 倍。本文将会重点介绍推理服务部署架构优化的工程实践，希望对从事相关工作的同学们有所帮助或启发。

1. 背景

随着越来越多的 AI 应用进入生产应用阶段，推理服务所需要的 GPU 资源也在迅速增加。调研数据表明，国内 AI 相关行业推理服务的资源使用量占比已经超过 55%，且比例未来还会持续升高。但很多公司面临的实际问题是，线上推理服务 GPU 利用率普遍较低，还具备很大的提升空间。

而造成服务 GPU 利用率低的重要原因之一是：推理服务本身存在性能瓶颈，在极限压力情况下也无法充分利用 GPU 资源。在这种背景下，“优化推理服务性能、提高 GPU 资源使用效率、降低资源使用成本”具有非常重要的意义。本文主要介绍如何通过架构部署优化，在保证准确率、推理时延等指标的前提下，提升推理服务的性能和 GPU 利用率。

2. 视觉模型服务的特点与挑战

近年来，深度学习方法在计算机视觉任务上取得显著进展，已经成为主流方法。视觉模型在结构上具有一些特殊性，如果使用现有推理框架部署，服务在性能和 GPU 利用率方面可能无法满足要求。

2.1 模型优化工具与部署框架

深度学习模型部署前通常会使用优化工具进行优化，常见的优化工具包括 TensorRT、TF-TRT、TVM 和 OpenVINO 等。这些工具通过算子融合、动态显存分配和精度校准等方法提高模型运行速度。模型部署是生产应用的最后一环，它将深度学习模型推理过程封装成服务，内部实现模型加载、模型版本管理、批处理以及服务接口封装等功能，对外提供 RPC/HTTP 接口。业界主流的部署框架有以下几种：

- **TensorFlow Serving**：TensorFlow Serving（简称 TF-Serving）是 Google 发布用于机器学习模型部署的高性能开源框架，内部集成了 TF-TRT 优化工具，但是对非 TensorFlow 格式的模型支持不够友好。
- **Torch Serve**：TorchServe 是 AWS 和 Facebook 联合推出的 Pytorch 模型部署推理框架，具有部署简单、高性能、轻量化等优点。
- **Triton**：Triton 是 Nvidia 发布的高性能推理服务框架，支持 TensorFlow、TensorRT、PyTorch 和 ONNX 等多种框架模型，适用于多模型联合推理场景。

在实际部署中，无论选择哪种框架，需要同时考虑模型格式、优化工具、框架功能特点等多种因素。

2.2 视觉模型特点

与传统方法不同，深度学习是一种端到端的方法，不需要单独设计特征提取、分类器等模块，用单个模型取代传统方法多模块任务。深度学习模型在分类、检测、分割、识别等视觉任务上呈现出巨大优势，做到了传统方法无法达到的精度。常用的视觉分类模型（如 ResNet、GoogleNet 等）和检测模型（如 YOLO、R-FCN 等）具有如下特点：

- **网络层数多（适合用 GPU 运算）**：以 ResNet-50 为例，网络包含 49 个卷积层和 1 个全连接层，参数量高达 2 千 5 百万，计算量达到 38 亿 FLOPs（浮点运算数）。模型推理包含大量矩阵计算，适合用 GPU 并行加速。

- **输入图像尺寸固定（需要预处理）**：同样以 ResNet-50 为例，网络的输入是图像浮点类型矩阵，尺寸大小固定为 224x224。因此二进制编码图片在送入网络前，需要做解码、缩放、裁切等预处理操作。

2.3 视觉推理服务面临的问题与挑战

由于视觉模型存在的上述特点，导致模型在部署和优化上存在 2 个问题：

1. **模型优化不彻底**：TensorRT、TF-TRT 等工具主要针对主干网络优化，但忽略了预处理部分，因此整个模型优化并不充分或者无法优化。例如分类模型中 ResNet-50 所有网络层都可以被优化，但预处理中的图像解码（通常是 CPU 运算）操作”tf.image.decode”会被 TF-TRT 忽略跳过。
2. **多模型部署困难**：视觉服务经常存在组合串接多个模型实现功能的情况。例如在文字识别服务中，先通过检测模型定位文字位置，然后裁切文字所在位置的局部图片，最后送入识别模型得到文字识别结果。服务中多个模型可能采用不同训练框架，TF-Serving 或 Troch Serve 推理框架只支持单一模型格式，无法满足部署需求。Triton 支持多种模型格式，模型之间的组合逻辑可以通过自定义模块（Backend）和集成调度（Ensemble）方式搭建，但实现起来较为复杂，并且整体性能可能存在问题。

这两点常见的模型部署问题，导致视觉推理服务存在性能瓶颈、GPU 利用率低，即便 Triton 这种高性能部署框架也难以解决。

通用部署框架重点关注的是“通信方式、批处理、多实例”等服务托管方面的性能问题，但如果模型本身中间某个部分（如图像预处理或后处理）存在瓶颈，优化工具也无法优化，就会出现“木桶效应”，导致整个推理过程性能变差。因此，如何优化推理服务中的模型性能瓶颈，仍然是一件重要且具有挑战性的工作。

3. GPU 服务优化实践

分类和检测是两种最基础的视觉模型，常应用在图像审核、图像标签分类和人脸检测等场景。下面以两个典型服务为案例，单个分类模型和“分类 + 检测”多模型组合的使用情况，来介绍具体性能的优化过程。

3.1 图像分类模型服务优化

美团每天有千万量级的图片需要审核过滤有风险内容，人工审核成本太高，需要借助图像分类技术实现机器自动审核。常用的分类模型结构如图 1，其中预处理部分主要包括“解码、缩放、裁切”等操作，主干网络是 ResNet-50。预处理部分接收图像二进制流，生成主干网络需要的矩阵数据 Nx3x224x224（分别代表图片数量、通道数、图像高度和图像宽度 ），主干网络预测输出图片分类结果。

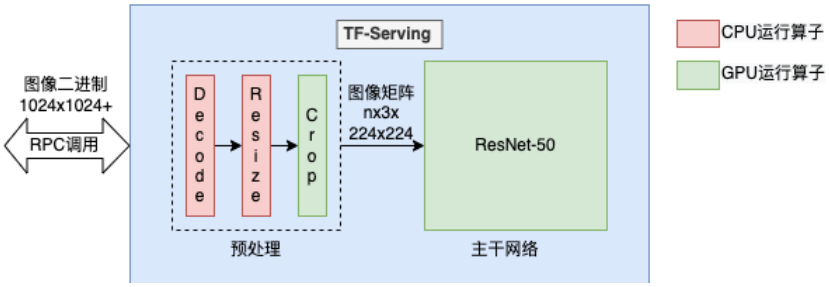


图 1　图像分类模型结构示意图

模型经过 TF-TRT 优化后的实际结构如下图 2 所示，主干网络 ResNet 被优化为一个 Engine，但预处理部分的算子不支持优化，所以整个预处理部分仍然保持原始状态。

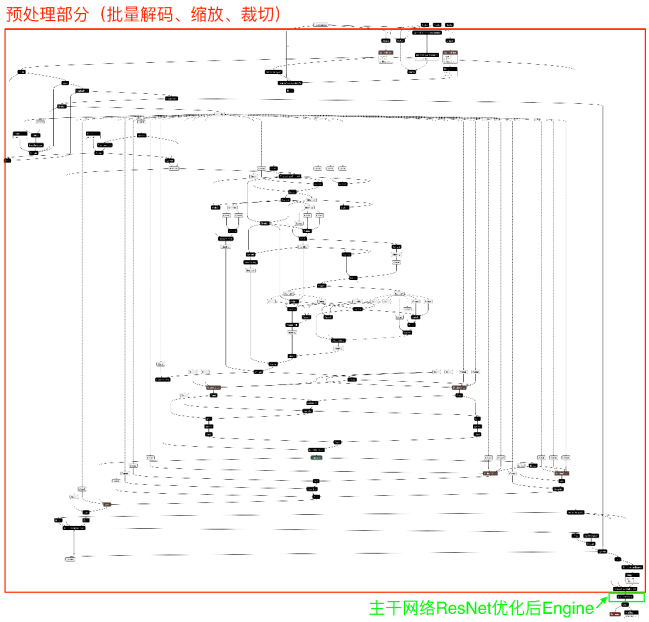


图 2　图像分类模型 TF-TRT 优化结构图

3.1.1 性能瓶颈

模型经过 TF-TRT 优化后使用 TF-Serving 框架部署，服务压测 GPU 利用率只有 42%，QPS 与 Nvidia 官方公布的数据差距较大。排除 TF-TRT 和 Tensorflow 框架等可能影响的因素，最终聚焦在预处理部分。Nvidia 进行性能测试的模型没有预处理，直接输入 Nx3x224x224 矩阵数据。但这里的线上服务包含了预处理部分，压测指标 CPU 利用率偏高。查看模型中各个算子的运行设备，发现模型预处理大部分是 CPU 运算，主干网路是 GPU 运算（具体细节参见图 1）。

如下图 3 所示，通过 NVIDIA Nsight System（nsys）工具查看模型运行时的 CPU/GPU 执行情况，可以发现 GPU 运行有明显间隔，需要等待 CPU 数据准备完成并拷贝到 GPU 上，才能执行主干网络推理运算，CPU 处理速度慢导致 GPU 处于饥饿状态。结合服务压测的 CPU/GPU 利用率数据可以看出：预处理部分 CPU 消耗高、处理速度慢，是推理服务的性能瓶颈。

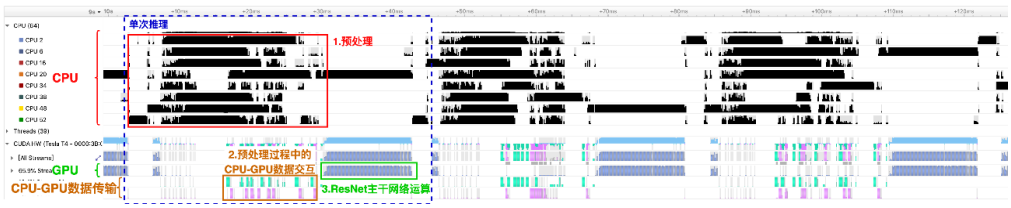


图 3　分类模型 nsys 性能诊断图

3.1.2 优化方法

如下图 4 所示，针对预处理 CPU 消耗过高影响推理服务性能的问题，提出几种优化方法：

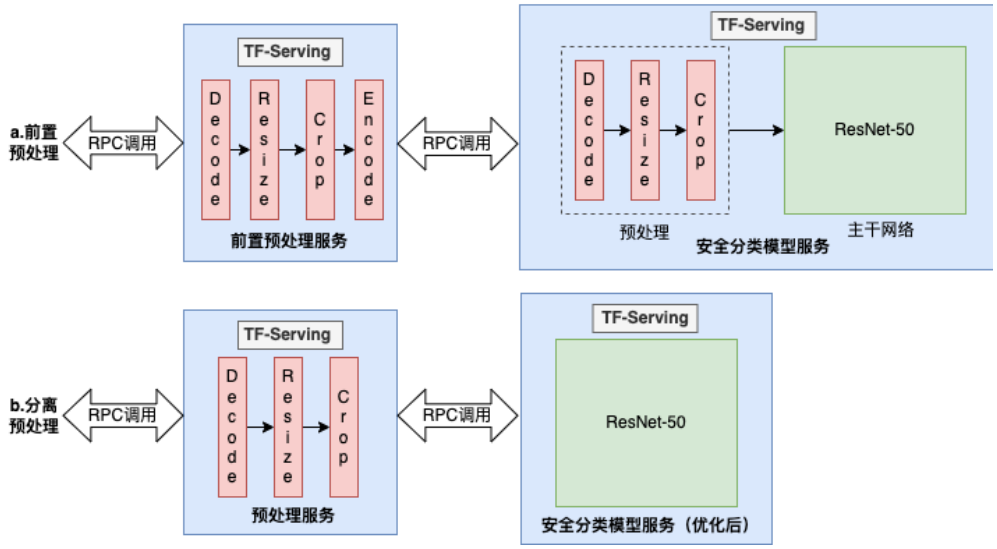


图 4　优化方法对比

1. **增加 CPU**：增加机器 CPU 数量是最简单的做法，但是受限于服务器硬件配置，1 个 GPU 通常只配置 8 个 CPU。所以增加 CPU 的方法只能用于性能测试数据对比，无法实际应用部署。
2. **前置预处理**：大尺寸图片的解码、缩放操作 CPU 消耗较高，相较而言小尺寸图片的处理速度会快很多。因此考虑对输入图片提前进行预处理，将预处理后的小图再送入服务。经过验证 Tensorflow 中缩放、裁切操作具有叠加不变性，即多次操作和单次操作对最终结果没有影响。预处理后的小图像经过编码后，再送入原始分类服务。需要注意的是图像编码必须选择无损编码（如 PNG），否则会导致解码数据不一致，引起预测误差。前置预处理方式的优点是不需要修改原始模型服务，可操作性高、预测结果没有误差。缺点是需要重复预处理，导致整体流程时延增加、CPU 资源浪费。
3. **分离预处理**：另外一种思路是将模型预处理部分和主干网络拆分，预处理部分单独部署到 CPU 机器上，主干网络部署到 GPU 机器上。这种做法让 CPU 预处理服务可以水平无限扩容，满足 GPU 处理数据供给，充分利用 GPU 性能。更重要的是将 CPU 和 GPU 运算进行解耦，减少了 CPU-GPU 数据交换等待时间，理论上比增加 CPU 数量效率更高。唯一需要考虑的是服务间的通信效率和时间，裁切后图像大小为 224x224x3，采用无符号整型类型数据大小是 143KB，传输时间和带宽在 1 万 QPS 以下没有问题。

3.1.3 优化结果

如下图 5 所示，我们利用 NVIDIA Nsight System，对比上述几种方法优化后的模型运行情况。增加 CPU 和前置预处理的方法都可以缩短 CPU 预处理时间，减少 GPU 数据等待时间，提升 GPU 利用率。但相较而言，分离预处理的方法优化更加彻底，CPU 到 GPU 的数据拷贝时间最短，GPU 利用最为充分。

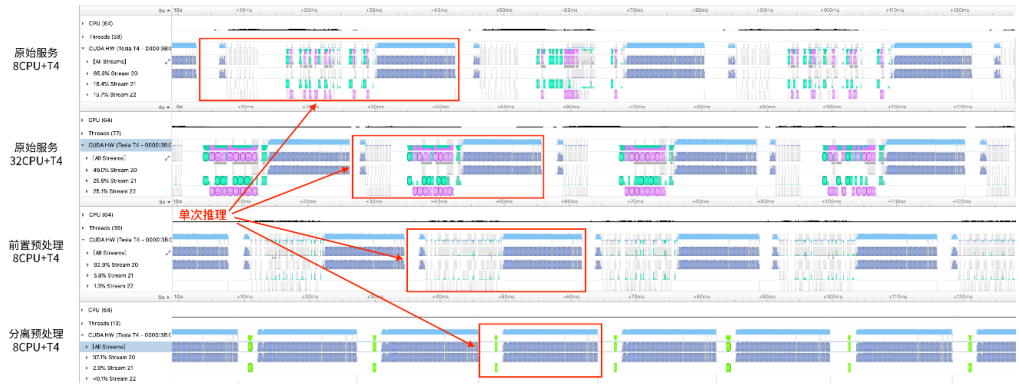


图 5　优化方法 nsys 性能诊断对比图

各种方法优化后的线上服务压测性能结果见下图 6（其中前置预处理和分离预处理中的 CPU 预处理服务额外使用 16 个 CPU），机器配置的 CPU 型号是 Intel　Xeon Gold 5218 CPU@2.30GHz、GPU 型号是 Tesla T4。从压测结果可以看出：

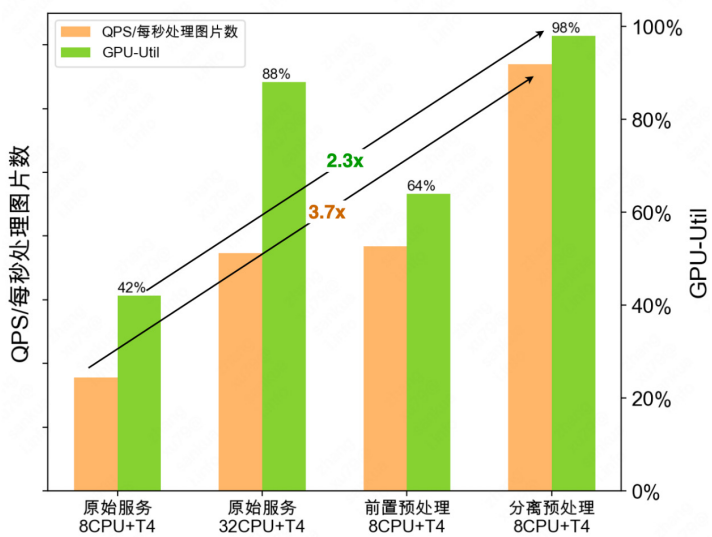


图 6　优化结果性能对比

1. 服务 CPU 增加到 32 核，QPS 和 GPU 利用率（通过 `nvidia-smi` 命令获取的 GPU-Util 指标）提升超过 1 倍，GPU 利用率提升至 88%；
2. 前置预处理方法优化后服务 QPS 提升超过 1 倍，略优于增加 CPU 的方法，但从 GPU 利用率角度看仍有较大优化空间；
3. 分离预处理方法优化后 QPS 提升 2.7 倍，GPU 利用率达到 98% 接近满载状态。

增加 CPU 并不能完全解决服务性能瓶颈问题，虽然 GPU 利用率达到 88%，但是 CPU-GPU 数据传输时间占比较大，QPS 提升有限。前置预处理方法也没有完全解决预处理性能瓶颈问题，优化并不彻底。相较而言，分离预处理方法充分发挥了 GPU 运算性能，在 QPS 和 GPU 利用率指标上都取得了更好的优化效果。

3.2 图像“检测 + 分类”模型服务优化

在一些复杂任务场景下（如人脸检测识别、图像文字识别等），通常是检测、分割、分类等多个模型组合实现功能。本节介绍的模型便是由“检测 + 分类”串接而成，模型结构如下图 7 所示，主要包括以下几个部分：

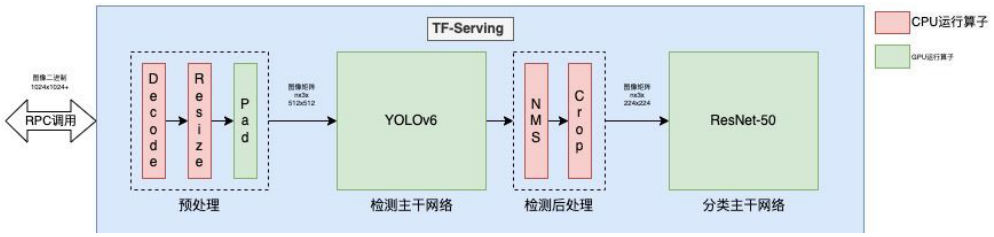


图 7　原始模型结构示意图

- **预处理**：主要包括图像解码、缩放、填充等操作，输出 Nx3x512x512 图像矩阵，大部分算子运行在 CPU 设备上。
- **检测模型**：检测网络结构是 YOLOv6，算子运行设备为 GPU。
- **检测后处理**：使用 NMS（非极大值抑制）算法去除重复或误检目标框，得到有效检测框，然后裁切目标区域子图并缩放，输出 Nx3x224x224 图像矩阵，检测后处理大部分算子运行在 CPU 设备上。
- **分类模型**：分类网络结构是 ResNet-50，算子运行设备为 GPU。

其中检测和分类两个子模型是单独训练的，推理时合并成单个模型，部署框架采用

TF-Serving，优化工具采用 TF-TRT。

3.2.1 性能瓶颈

模型中预处理和检测后处理大部分是 CPU 运算，检测和分类模型主干网络是 GPU 运算，单次推理过程中需要进行多次 CPU-GPU 之间数据交换。同样地，CPU 运算速度慢会导致 GPU 利用率低，推理服务存在性能瓶颈。

实际线上服务压测 GPU 利用率 68%，QPS 也存在较大优化空间。服务配置为 1GPU+8CPU（机器 CPU 型号是 Intel Xeon Gold 5218 CPU@2.30GHz，GPU 型号是 Tesla T4）。

3.2.2 优化方法

仍然采用模型拆分的思路，将 CPU 和 GPU 运算部分单独部署微服务。如下图 8 所示，我们将原始模型拆分为 4 个部分，预处理和检测后处理部署成 CPU 微服务（可根据流量大小动态扩容），检测模型和分类模型部署成 GPU 微服务（同一张 GPU 卡上）。为了保持调用方式简单，上层采用调度服务串接各个微服务，提供统一调用接口，对上游屏蔽调用差异。拆分后的检测模型和分类模型经过 TensorRT 优化后采用 Triton 部署。

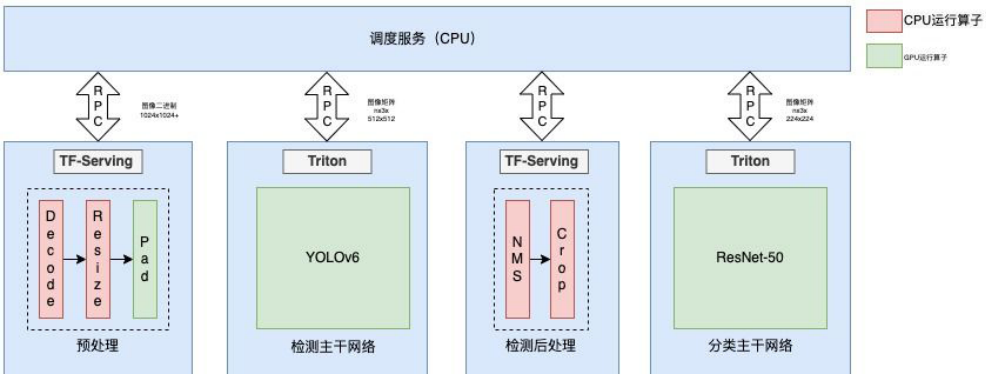


图 8 模型优化部署结构示意图

3.2.3 优化结果

如下图 9 所示，除了原始服务和微服务拆分，另外还对比了增加 CPU 和 Triton Ensemble 两种方式的性能测试结果。Triton Ensmeble 方式将所有子模型（包括预处

理和后处理）部署在一个机器上，实现多模型流水线调度。对比可以看出：

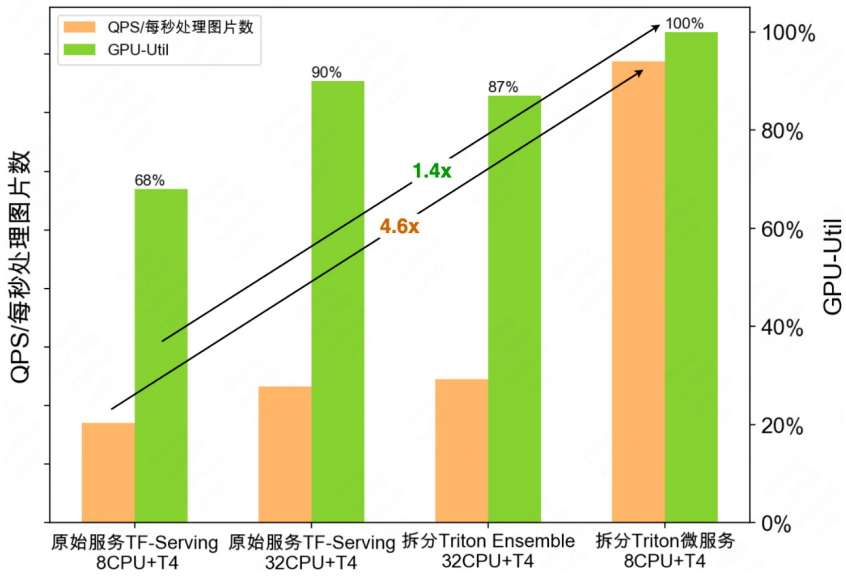


图 9　优化结果性能对比

- 原始服务 CPU 增加到 32 核，GPU 利用率提升到 90%，但 QPS 提升只有 36%；
- Triton Ensemble 方式与原始 TF-Serving 服务相比性能差距不大；
- 模型拆分微服务方式优化后 QPS 提升 3.6 倍，GPU 利用率达到 100% 满载状态。

增加 CPU 的方法对服务 GPU 利用率提升较大、QPS 提升不明显，原因在于 CPU 预处理和后处理时间缩短，但 CPU-GPU 数据传输时间在整个推理过程中仍然占比较大，GPU 运算时间较少。

将模型拆分使用 Triton Ensmeble 方式部署实现多模型调度流水线，效率并不比模型拆分前高。因为多模型流水线是在同一台机器上，CPU 和 GPU 相互影响的问题并没有解决，GPU 处理效率受 CPU 制约。模型拆分微服务部署方式在服务层面实现了调用流水线，CPU 预处理和后处理微服务可以根据流量动态增加资源，满足 GPU 模型微服务吞吐需求，实现了 GPU/CPU 处理解耦，避免了 CPU 处理成为瓶颈。

总而言之，拆分微服务的方式充分发挥了 GPU 运算性能，在 QPS 和 GPU 利用率指

标上都取得了更好的效果。

4. 通用高效的推理服务部署架构

除了上述两个具有代表性的推理服务优化案例，视觉其它很多服务也采用了这种优化方式。这种优化方式具有一个核心思想：**在 CPU 运算是瓶颈的推理服务中，将模型的 CPU 和 GPU 运算部分拆分，单独部署成 CPU/GPU 微服务**。

根据这种思想，我们提出一种通用高效的推理服务部署架构。如下图 10 所示，底层使用通用的部署框架（TF-Serving/Triton 等），将模型中 CPU 运算部分如预处理、后处理拆分单独部署成 CPU 微服务，主干网络模型部署成 GPU 服务，上层调度服务串联模型微服务实现整体功能逻辑。

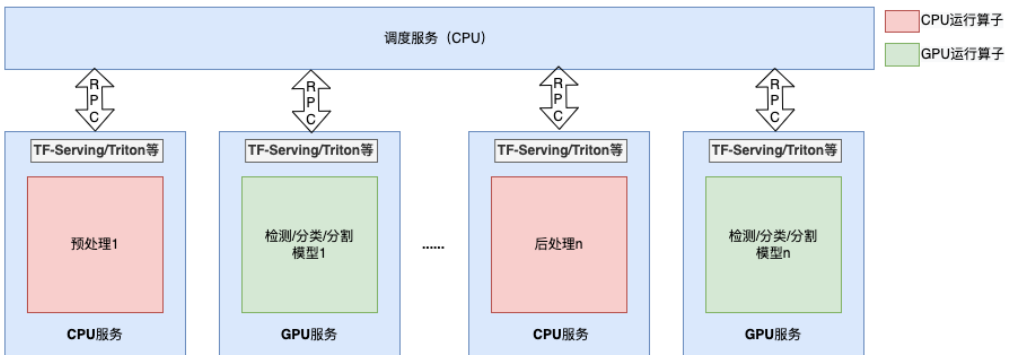


图 10　通用服务部署架构示意图

这里需要解释一个重要问题，为什么这种部署架构是高效的?

首先在宏观层面，模型拆分部署成微服务，通过调度服务实现了子模型流水线处理。拆分的子模型 CPU 微服务可以根据流量和处理能力动态扩容，避免模型预处理或后处理 CPU 运算能力不足成为性能瓶颈，满足了 GPU 模型微服务的吞吐需求。

其次在微观层面，如果模型包含多个 CPU/GPU 运算部分，那么 GPU 运算之间会存在间隔。如下图 11 所示，单次推理过程中两次 GPU 运算之间需要等待 CPU 后处理部分完成，CPU 预处理也会影响 GPU 运算。模型拆分后，预处理和后处理部分独立部署成 CPU 服务，GPU 服务中推理过程仅包含两个子模型运算部分，并且子模型间运算相互独立无数据关联，CPU-GPU 数据传输部分可以与 GPU 运算过程并行，理

论上 GPU 可以达到 100% 运行效率。

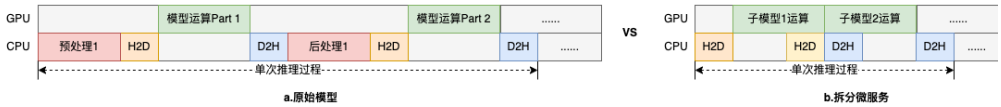


图 11　推理过程对比示意图

另外，在时延方面，拆分微服务的部署方式增加了 RPC 通信和数据拷贝时间开销，但从实践来看这部分时间占比很小，对端到端的延迟没有显著影响。例如对于上面 3.1 节中的分类模型，优化前服务的平均耗时是 42ms，优化后微服务形式（RPC 通信协议为 Thrift）的服务整体平均耗时是 45ms。对于大部分视觉服务应用场景而言，这种级别的延迟增加通常并不敏感。

5. 总结与展望

本文以两个典型的视觉推理服务为案例，介绍了针对模型存在性能瓶颈、GPU 利用率不高的问题进行的优化实践，优化后服务推理性能提升 3 倍左右，GPU 利用率接近 100%。根据优化实践，本文提出了一种通用高效的推理服务部署架构，这种架构并不局限于视觉模型服务，其它领域的 GPU 服务也可参考应用。

当然，这种优化方案也存在一些不足，比如优化过程中模型如何拆分比较依赖人工经验或者实验测试，没有实现优化流程的自动化与标准化。后续我们计划建设模型性能分析工具，自动诊断模型瓶颈问题，支持自动拆分优化全流程。

6. 作者

| 张旭、赵铮、岸青、林园、志良、楚怡等，来自基础研发平台视觉智能部。
| 晓鹏、驶飞、松山、书豪、王新等，来自基础研发平台数据科学与平台部。

MRCP 在美团语音交互中的实践和应用

作者：唐锐 森彬 子丰 亚男 王程 国桥 俞涛

一、背景

智能语音对话作为人工智能领域最先落地的分支之一，可以实现与人进行自然语言多轮对话，其核心技术在近年来不断地发展成熟，包括语音识别、自然语言理解、对话管理等。伴随着技术的成熟，越来越多的电话机器人开始走进我们的生活，这些电话机器人在客户服务、智能销售、通知触达等场景发挥着重要的作用。

当你和智能语音机器人对话交互时，你是否好奇电话背后的机器人如何“听懂”你的意思，又如何像人一样“回答”你的问题？经典的技术实现路径是：机器人首先通过“语音识别（ASR）”将用户输入语音识别成文字，再通过“自然语言理解（NLU）”识别意图，之后根据意图、系统信号等输入结合对话管理技术得到相应的回复，最后通过“语音合成（TTS）”生成语音播报给电话对端的用户。但要将 ASR、TTS 这些技术应用到电话系统上，还需要一些额外的工作和技术支撑，其中比较重要的技术之一也就是本文将要介绍的 **MRCP**。

备注：本文涉及较多的专业名词，我们特别在文末附上了名词解释，以帮助大家进行阅读。

1.1 MRCP 是什么

MRCP（Media Resource Control Protocol, MRCP）是一种通讯协议，中文定义是：媒体资源控制协议，用于语音服务器向客户端提供各种语音服务（如语音识别和语音合成）。该协议定义了控制媒体处理资源所必需的请求（Request）、应答（Response）和事件（Event）等消息，它需要借助 RTP（Real-Time Transport Protocol, 实时传输协议）创建一个媒体会话、借助 SIP（Session Initiation Protocol, 会话初始化协议）和 SDP（Session Description Protocol, 会话描述协议）创建一个控制会话来实现媒体资源服务器端和客户端之间的控制 [1]。

在传统的语音应用中，各集成商必须针对不同的 ASR/TTS 厂商提供的 API 接口进行

专门的集成开发，不同 ASR/TTS 引擎的接口各不相同，从而导致了集成过程的复杂性和局限性。因此，在实现语音和网络技术集成方面必须需要比较规范的协议来进行处理。MRCP 协议是目前针对媒体资源和 IP 网络起草的标准协议。有了 MRCP 协议后，ASR/TTS 厂商提供 MRCP 协议的标准统一接口，语音集成开发商们不必再针对特定的 ASR/TTS 进行开发，为各种语音应用开发提供了更加灵活的选择，并有效地降低业务开发周期和成本 [2]。

以语音合成（TTS）为例，图 1 展示了一个 MRCP 客户端连接媒体资源服务器的流程，在连接中会创建一个媒体会话和一个控制会话。假设连接成功，此时 MRCP 客户端对服务器端发起了一个 SPEAK 语音合成的请求，处理此请求的过程中包括了 3 个消息指令：`SPEAK` 指令表示发送语音合成请求，`IN-POGRESS` 指令表示正在处理中，`SPEAK—COMPLETE` 指令表示处理完成 [3]。

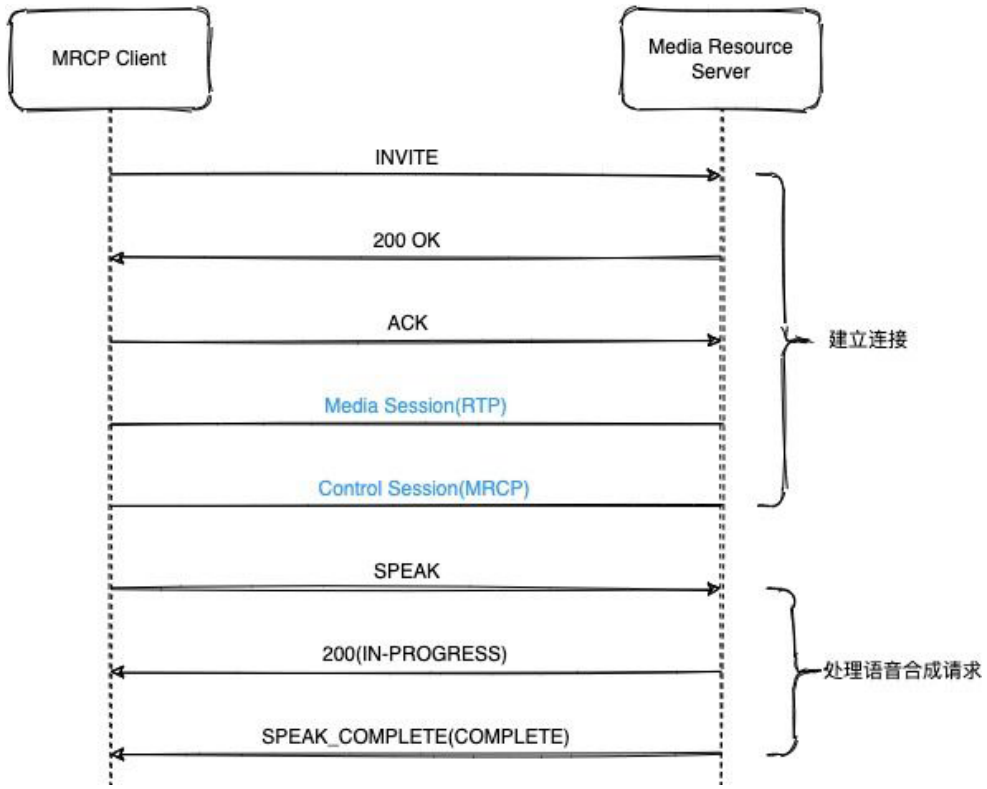


图 1　MRCP 协议交互

MRCP 协议的消息格式和 HTTP 消息格式类似，如下以一次语音合成的 MRCP 消息

为例，展示了 MRCP 消息的主要构成。

发起语音合成请求：

```
MRCP/2.0 380 SPEAK 14321
Channel-Identifier: 43b9ae17@speechsynth
Content-Type: application/ssml+xml
Content-Length: 253

您好，有什么能帮助您?
```

IN-PROGRESS 响应消息：

```
MRCP/2.0 119 14321 200 IN-PROGRESS
# 消息长度是 119 bytes,ID 是 14321，200 表示成功，正在处理中
Channel-Identifier: 43b9ae17@speechsynth
Speech-Marker: timestamp=857206027059
```

COMPLETE 响应消息：

```
MRCP/2.0 157 SPEAK-COMPLETE 14321 COMPLETE
Channel-Identifier: 43b9ae17@speechsynth
Speech-Marker: timestamp=861500994355
# 表示 SPEAK 请求的正常处理结束
Completion-Cause: 000 normal
```

1.2 MRCP 使用场景

目前，语音行业几乎所有的重要厂商都承诺支持 MRCP。MRCP 使用场景丰富，它支持目前最热门的开源语音通信平台 Asterisk 和 FreeSWITCH ，并且提供了丰富的接口文档，其中呼叫中心就是一个典型的案例。一个简单的呼叫中心如下图 2 所示：

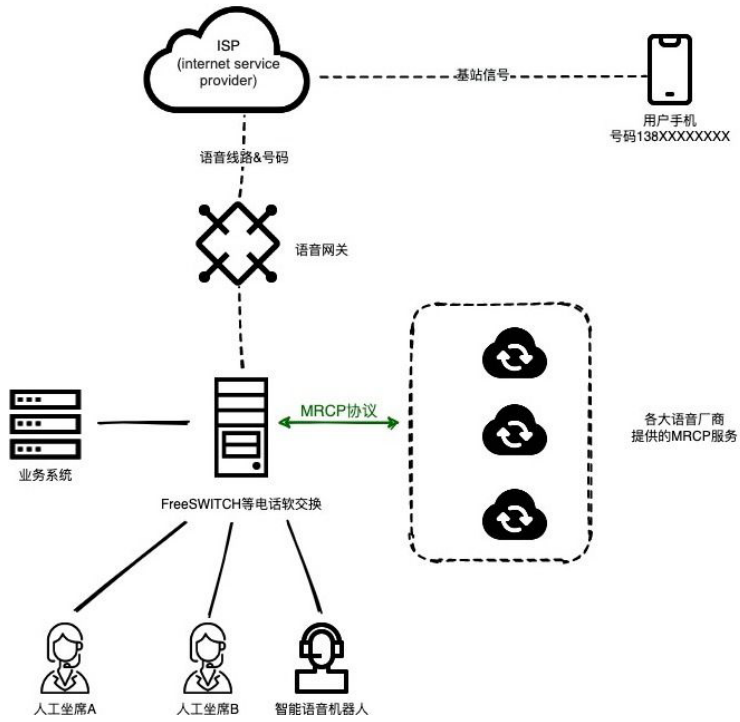


图 2　简单呼叫中心系统示意

当分机拨打外呼用户时，需要从运营商那里获得一条电话通讯线路，FreeSWITCH 指定路由通过网关到达运营商线路，通过运营商基础网络、连接用户通讯设备终端。

智能语音服务通过 MRCP 协议与 FreeSWITCH 进行对接，用户接通电话后，智能语音服务从呼叫中心设备中实时获取声音讯号，将语音讯号转化为文本流实时输出，并将要回复的文本话术经过语音合成转化为语音讯号，交由呼叫中心进行语音播报。

在此场景中可以看到，采用 MRCP 协议，**调用方仅需要面向 MRCP 接口撰写程序，而无需考虑不同语音引擎产品之间的差异**，可以真正做到**一次开发、多种环境下应用**，任何支持 MRCP 标准的语音引擎都可以被无缝集成和调用。

1.3 美团自研的 ASR/TTS 能力

自 2018 年起至今，美团语音交互部持续投入语音识别（ASR）和语音合成（TTS）的自主研发，目前已形成平台级的服务能力。美团语音识别重点针对美团场景进行优化，相比通用场景的识别率更高；参考 2022 年的数据，在电话呼叫场景的测试集中，美团语音识别的字准率达到 94.6%（很多业界头部厂商的字准率在 89% 左右）。在骑手语

音助理、客服中心语音转译、美团 App / 外卖 App 语音助理等典型业务场景中进行了落地和应用。

美团语音合成从美团各场景出发，建立起从端到云一体化，全面覆盖客服、配送、听书等各个方向的合成音色群，并支持不同数据量级的语音定制化能力，做到了通用场景好、特色场景精、定制周期短。其中现有声音的客服场景效果已领先于行业，具有小样本声音克隆、强表现力的配音能力，在性能和效果层面达到了业界一流水准；同时，美团语音合成在美团外卖配送、美团商家端、美团打车、美团客服等核心业务场景落地，支持日均亿级别的调用。

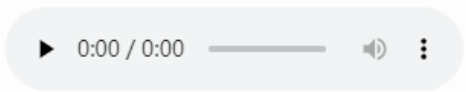
1.4 我们为什么需要 MRCP

1.4.1 赋能内部业务

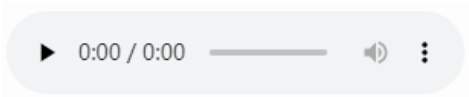
随着美团自研的 ASR/TTS 逐步达到业界一流水平，美团内部越来越多业务接入美团自研的 TTS 和 ASR 能力。特别是 TTS，在应用的业务场景中取得超过外采系统捷通华声的效果，但在业务对接和优化过程中，也存在一些问题，可以概括为音色机械、音色不统一、合成延时过高等。

这几类问题，主要是在业务升级替换音色过程中，采取了将业务系统（如外呼系统）与语音的合成和识别能力的 HTTP/RPC 接口直接对接的方式，这种方式不仅投入大，需要逐个业务系统、逐个流程的对接，也容易因为系统复杂性、运营疏漏等问题出现音色不统一等体验问题。因此，按行业通用标准，以 MRCP 将语音合成和识别与电话系统直接对接的方式，可以有效地降低业务使用、升级语音能力的成本，平滑地提升用户体验。

音色不统一示例：对话流程中，一部分固定的话术文本使用的提前合成好的音频文件，另一部分动态的话术文本（如，录音中“请问你是某某店吗”）采用的实时合成的音频，两部分拼接在一起播放，音色不统一。



延时过高：部分业务对于动态话术文本，特别是本身句子较长的话术，待一整句合成完成后再播放到用户，给用户带来严重的迟滞感。



1.4.2 支撑对外商业化

另一方面，越来越多美团外部企业如中通天鸿、微呼科技、马上消费金融等，认可并计划接入美团语音自研的 TTS 和 ASR 能力；预期以标准的 MRCP 协议完成对接，在客户服务、通知触达、电话提示语音识别等场景，提升其呼叫中心的基础用户体验。

二、设计与实现

2.1 设计目标

如下图 3 所示，美团的小美机器人平台、木星通讯平台分别提供不同类型的语音对话机器人能力。以语音合成（TTS）为例，这些机器人平台直接调用 TTS 引擎的服务接口，将合成好的语音文件交由电话软交换平台（如 FreeSWITCH）去播报，如链路 ① 所示。

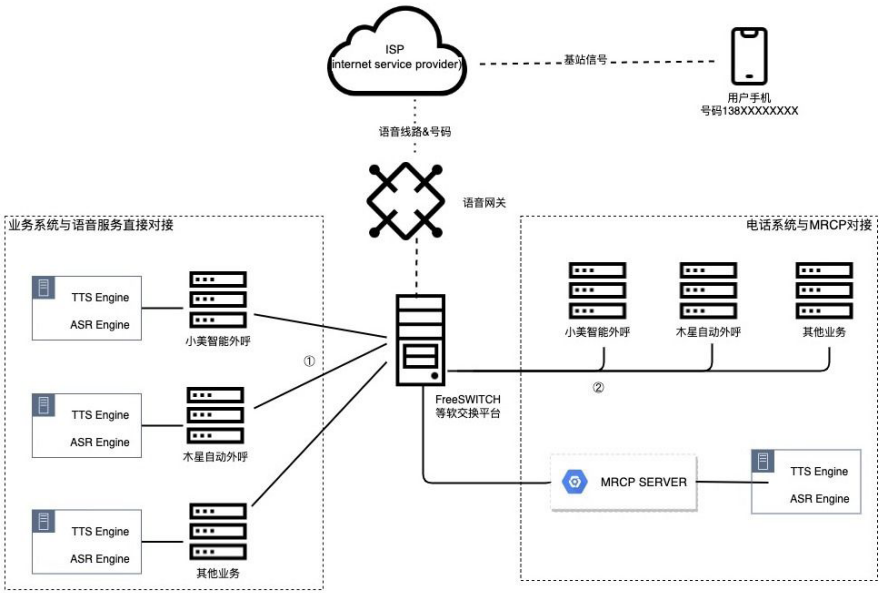


图 3　语音能力与业务系统对接方式比较

我们的目标是将这种调用关系简化，以标准 MRCP 下的语音合成服务对接电话软交换平台，那么上述机器人平台则只用核心关注机器人的对话逻辑，将具体的语音合成逻辑解耦出来，那么链路 ② 所传递的内容即为机器人待播报的话术文本，由电话软交换平台去调用和处理语音合成。

总而言之，目标结合自研流式 TTS 和 ASR 能力，建设 MRCP 协议下标准的语音合成和识别服务，达到：

- 支持标准协议下的 TTS 和 ASR 能力对接方式，追齐业界主流厂商能力。
- 以横向可扩展、业务解耦的方式，支持和助力美团内部业务，在智能外呼、内呼热线等场景下提升用户体验；并为美团语音能力的对外商业化探索，提供更好的支撑。

2.2 总体系统架构

2.2.1 系统层次架构

在系统层次上，围绕联络场景，美团语音交互部正在建设全联络场景智能化的平台化解决方案。具体来说，美团智能对话平台 AICC（AI for Contact Center），基于美团语音交互部领先的语音识别、自然语言理解、多轮对话、知识图谱等人工智能技术，为美团业务提供智能文字客服（在线客服）、智能语音客服（热线客服）、智能外呼、智能营销、智能质检等完整解决方案；帮助业务从传统服务模式向智能服务模式转型，助力美团业务的服务成本优化、客户服务体验提升，实现客户服务及营销智能化升级。

AICC 的层次架构如下图 4 所示，从整个 AICC 架构来看，TTS 和 ASR 处于语音技术平台，提供语音合成和识别的 PaaS 级能力；相应地，MRCP-TTS、MRCP-ASR 扩充已有的 HTTP/RPC 接口的能力范围。

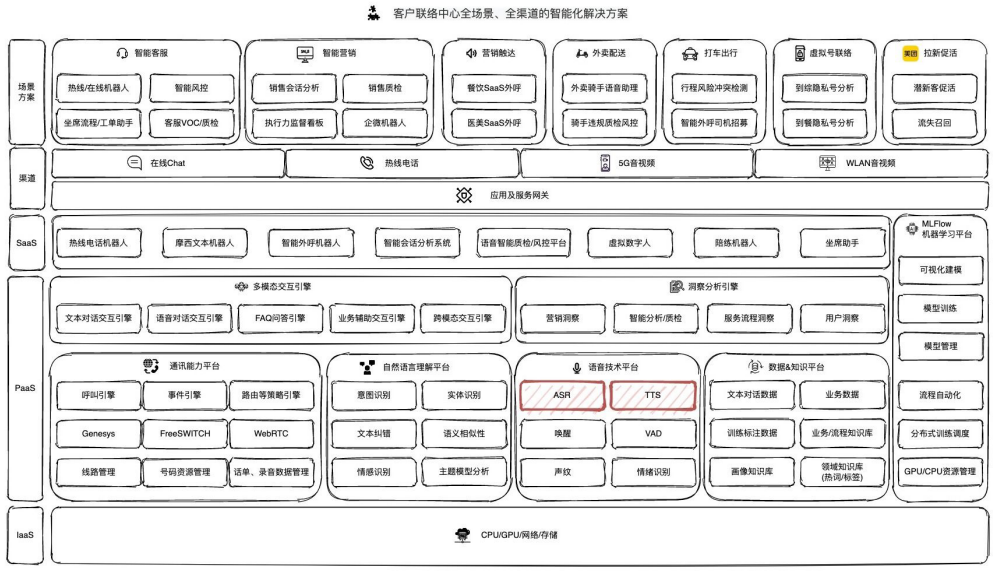


图 4　MRCP 在 AICC 层次架构中的位置关系

2.2.2 系统间调用关系

以热线电话机器人的系统调用过程为例，MRCP 在系统中所处的位置以及同其他各环节的配合关系如下图 5 所示：

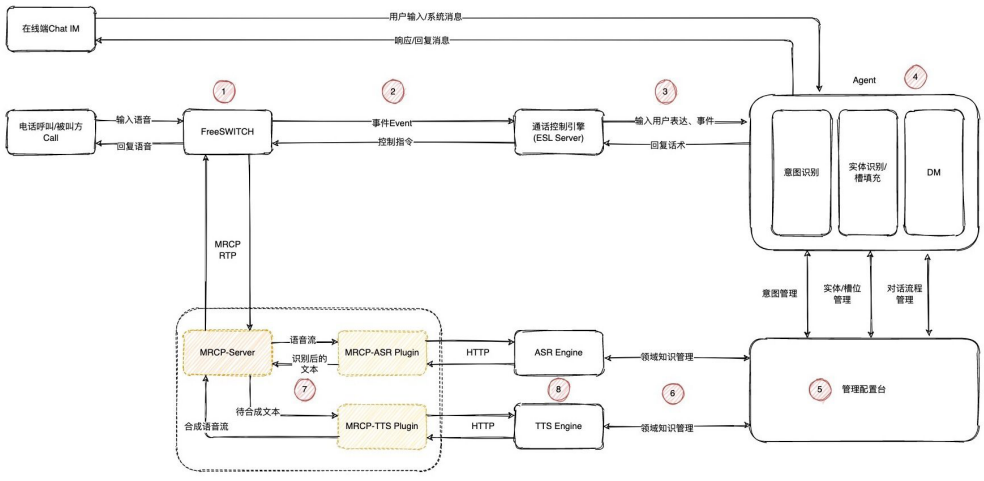


图 5　MRCP 服务调用关系

1. FreeSWITCH 电话软交换平台，负责和运营商打通通讯线路，以具备基础的电话通讯能力。

2. FreeSWITCH 除了以内置模块（如 mod_java）的方式开发控制接口外，也以 ESL（Event Socket Library）的 Inbound/Outbound 方式开放接口，提供事件监听、通话控制等能力。
3. ESL Server 将监听到的事件、消息传递给具体的业务逻辑，可以提供通讯层所有的事件供监听和处理，籍此实现机器人的语音对话交互能力。
4. 将调语音相关的事件和信号处理解耦以后来看，热线、在线机器人的交互逻辑则可以简化、抽象为统一的模型和系统。
5. 管理配置台主要负责一般对话机器人所需要的意图、槽位的定义、管理和配置任务型对话流程。
6. 配置管理也可对 ASR、TTS 引擎需要的领域知识进行管理，比如客服领域的词库、样本数据集的持续标注等。
7. 本文所述的 MRCP 在系统调用中处于此位置：在 FreeSWITCH 收到合成 / 识别请求后，发起与 MRCP-Server 的交互，MRCP-Server 调用内部实现的 MRCP-TTS Plugin 与 MRCP-ASR Plugin 分别完成相应的合成或识别结果。
8. ASR Engine 和 TTS Engine 指美团语音自研的语音合成和语音识别引擎，MRCP 通过 HTTP/RPC 接口与之完成通讯。

2.3 关键技术组件

要实现一个工业可用的 MRCP Server，有两个关键技术能力：一是 MRCP 协议本身的支持，二是 MRCP Server 的高可用，如多节点的负载均衡。

2.3.1 MRCP 协议实现

对于 MRCP 协议的实现，不仅仅需要支持 MRCP 协议本身，也需要支持一套完整的协议栈，包括文章开头部分提到的 SIP、RTP 协议等，这是一个复杂且庞大的工作。

参考业界的一般做法，我们选择基于开源的 UniMRCP 完成这些工作。UniMRCP 是一个开源的、跨平台的 MRCP 协议实现，由 C/C++ 语言编写，包括了 MRCP 客户端和服务端两个部分，它封装了 SIP、SDP、MRCPv1、MRCPv2、RTP/RTCP 协议栈，并为语音服务集成商提供了一致的 API[4]。

UniMRCP 完成了 MRCP 协议栈的封装，并没有实现 ASR 或 TTS ；基于其对底层协议栈的完整支持，我们在 UniMRCP 的框架下实现相应的 Recog（ASR）和 Synth

（TTS）插件（即 MRCP-TTS Plugin，MRCP-ASR Plugin），并改造适配美团日志框架、监控打点等基础技术组件，从而保障服务的稳定性和可维护性。

2.3.2 MRCP Server 负载均衡

对于实现 MRCP-Server 的负载均衡，我们选用开源的 Kamailio 来完成。Kamailio 的前身叫 Openser，作为出色的 SIP proxy，在大并发量使用时常常用于负载均衡媒体服务器，对 Asterisk、FreeSWITCH、MRCP 等实现集群能力[5]。Kamailio 经常与 FreeSWITCH 配合使用，最常用的场景是 Kamailio 作呼叫负载均衡服务器（一般主备配置），FreeSWITCH 做媒体相关的处理如转码、放音、录音、呼叫排队等。

2.4 MRCP-Server 模块结构

由于 UniMRCP 提供的基础框架已经实现了服务接口、连接管理、协议管理，如何加载自定义插件，以及系统层的日志框架；并且 TTS 引擎与 ASR 引擎作为基础的依赖，已以 HTTP/RPC 协议的方式提供稳定的基础语音服务。因此，开发工作是在 UniMRCP 的基础上进行 TTS/ASR 插件的开发，模块结构如下图 6 所示，主要新增的模块已在图中标灰，其中：

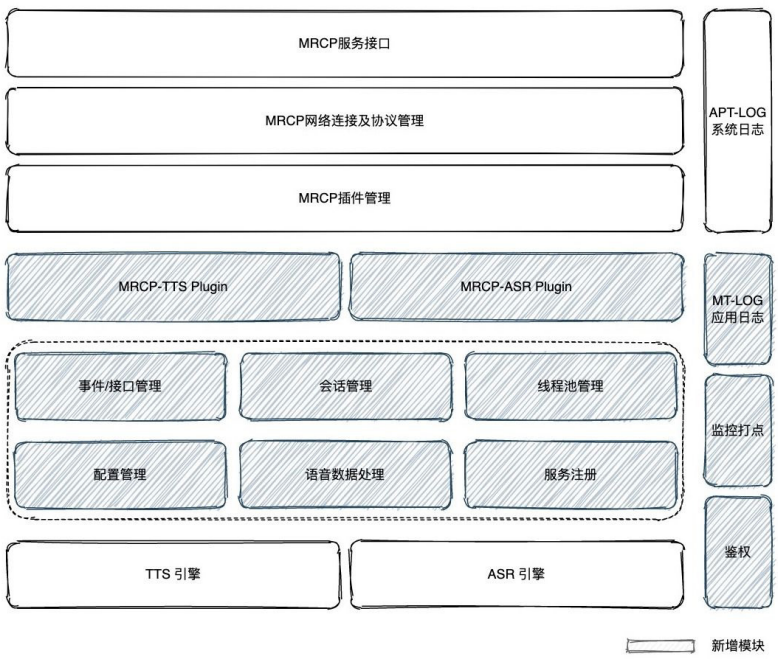


图 6 MRCP 服务模块结构

- MRCP-TTS Plugin 和 MRCP-ASR Plugin 两个为核心的模块，以插件的形式加载到 MRCP-Server。
- MRCP-TTS Plugin 和 MRCP-ASR Plugin 抽象出公共的模块：
  - 事件 / 接口管理：订阅和注册 MRCP-Server 收到事件和消息，如信道建立事件、请求消息等等。
  - 会话管理：管理会话的创建、打开、关闭、销毁过程。
  - 线程池管理：管理调用 TTS 和 ASR 引擎服务接口所需的任务线程池。
  - 配置管理：读取配置文件对各自 Plugin 定义的配置项。
  - 语音数据处理：包括语音数据缓冲、时长计算等。
  - 服务注册：负责将服务注册到 OCTO，支持 MRCP 服务实例激活、摘除的自动发现。
- 应用日志模块，将插件这一层的日志与美团的日志框架打通，将应用日志接入美团日志中心。
- 监控打点模块，将合成首包延时、状态事件等支持，接入美团 Raptor 监控平台。
- 鉴权模块，使用美团语音开放平台的统一认证服务，完成调用 TTS 和 ASR 引擎前的鉴权过程。

2.5 部署方案

在语音交互场景中，用户可以连续不间断的“说话”并听到“回应”，是保证本次交互顺利完成的重要基础。因此，高可用是 MRCP-Server 在部署方案设计中核心关注和考虑的维度，部分节点故障不影响整体的可用性，并可以快速摘除、恢复、替换故障节点。其中，关键的措施包括：

- **统一的服务入口：**使用美团 MGW（美团内部自研的四层负载均衡网关）创建固定的虚拟 IP (VIP)，提供唯一的服务访问入口。
- **资源隔离：**为了减少语音合成和语音识别服务彼此间的影响，我们将 MRCP Server 拆分成两个服务，即 MRCP-TTS 和 MRCP-ASR，对二者进行隔离部署。
- **跨地域、多机房部署：**将 MRCP-TTS 和 MRCP-ASR 服务部署在多个机房，并且跨地域部署，以提高服务稳定性。

- **Kamailio 热备：**为了防止 Kamailio 出现故障以影响整个链路，我们对 Kamailio 也采用了跨地域、多机房的部署，并且由 MGW 完成对 Kamailio 的多机热备。
- **负载均衡：**显而易见地，需要对 MRCP-TTS 和 MRCP-ASR 进行负载均衡。如何进行负载均衡？起初我们打算直接使用 MGW 对 MRCP-TTS 进行负载均衡，MGW 是支持到传输层的四层负载均衡，可以路由分发到具体的服务。在我们场景中所使用的 SIP、MRCP 均属于应用层的协议，数据传输基于 TCP/UDP，理论上四层负载均衡能够满足需要。但实际协议交互过程中，支持七层负载均衡的 Kamailio 能够在 SIP 协议经过的路由点时，在协议头中加入 `<Via address>` 和 `<Contact address>` 等字段，层层路由能够始终保持与同一台机器进行交互。比如，客户端通过 SIP-INVITE 经由 Kamalio 与 MRCP 服务完成握手后，结束服务、客户端发送 SIP-BYE 消息时，Kamailio 可以准确路由到当时接受 INVITE 消息的同一个 MRCP 服务。因此，最终使用 Kamailio 完成对 MRCP-TTS 和 MRCP-ASR 的负载均衡。

对应地，整体部署如图 7 所示，在美团通过这种部署方式为业务提供稳定可用的服务。

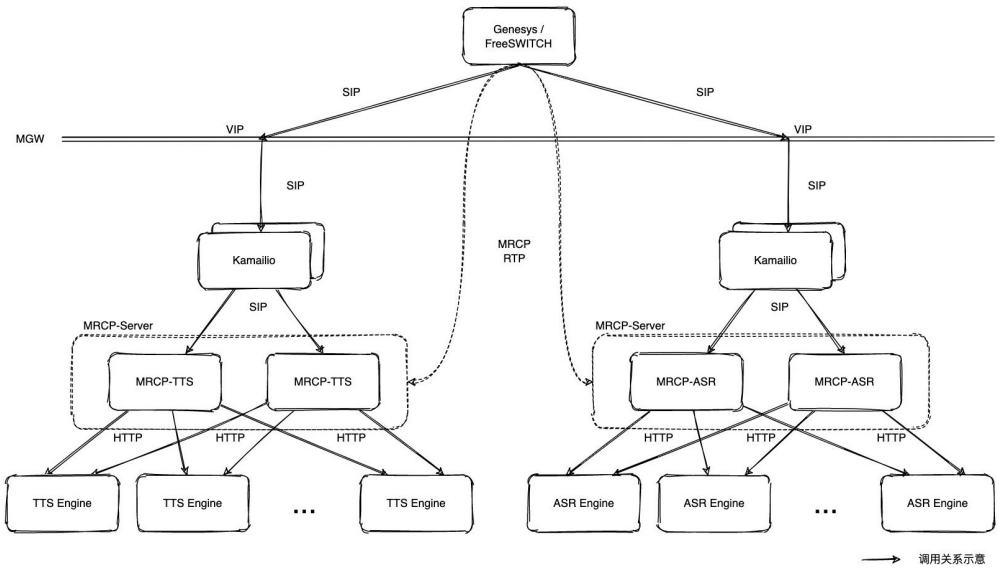


图 7　MRCP 整体系统部署方案

三、实践与应用效果

3.1 应用范围

3.1.1 美团内部业务赋能

目前 MRCP 已经在美团内外呼热线电话下的多个业务场景中广泛应用：

- **美团外呼业务语音合成：**结合小美智能外呼与木星自动外呼，在美团内部的太平洋客服坐席辅助外呼、营销类外呼、通知触达类外呼等场景，逐步接入 MRCP 协议下的流式语音合成。目前已接入 MRCP 的外呼机器人日均百万次合成调用，峰值约千路并发。
- **美团内呼业务语音合成：**内呼主要是支持美团呼入热线的语音合成，包括：10107888- 美团客服热线、10109777- 外卖客服热线、10105777- 美团商服热线、10101777- 美团骑手热线以及其他长尾业务热线。目前已接入 MRCP 的热线呼入机器人日均近千万次合成调用量，峰值千余路并发。

3.1.2 外部商业化应用

在外部商业化落地过程中，客户一般要求将 MRCP-Server 私有化部署到客户机房，服务的运维工作由客户自行保证；MRCP-Server 本身在 8C16G 容器中测试，可承载约 600 路并发访问。目前已支持的美团外部客户包括：

1. 微呼科技，接入了美团提供的 MRCP 语音合成和识别服务，在通知触达等场景使用美团语音提供的流式语音合成能力，在“空号”、“忙音”、“助手秘书提示”等场景使用美团提供的语音识别能力。
2. 中通天鸿，接入了美团提供的 MRCP 语音合成服务，支持其外呼机器人等业务下的客户选择美团语音提供的多种场景化音色。

3.2 应用效果

MRCP 将语音合成和识别能力接口标准化以后，在美团对接了外采的 Genesys 电话系统（主要支持内呼电话），以及自研的木星通讯系统（主要支持外呼电话），也在支持语音合成和识别能力商业化的过程中对接了中通天鸿、微呼科技等外部公司的电话系统。在这些业务对接和支持的过程中，一次开发多处复用，相较于早期的对接方式，极

大提高了业务支持的效率。

此外，对于使用 MRCP-TTS 的业务，在语音合成的性能有明显提升，具体体现在：首包延时显著降低，且与待合成的话术文本句长无关。此前，使用 HTTP 接口进行整句合成播报，以大约 6~8 字每秒的合成速度、需要整句完成后用户才能听到声音。目前，MRCP-TTS 支持一边合成一边为用户播报语音。

从线上观测来看，内呼机器人的端到端延时降低了约 55%，外呼机器人的端到端延时降低了约 33%，并且做到了音色统一、无变量导致的音色跳跃问题。从业务使用 MRCP-TTS 前后的业务指标对比来看，有较为显著的效果：

- 在美团电话热线的呼入场景，使用 MRCP-TTS 升级原来的语音播报服务后，不满意度下降 0.25pp~3.92pp，平均电话服务时长缩短 2.19s~5.3s，可见 MRCP-TTS 的应用带来了更好用户体验。
- 在美团的智能外呼场景，结合 MRCP-TTS 提供流式语音合成服务后，对外呼机器人进行替换自研音色「美凡楠」的 A/B 试验，数据表明智能外呼通话成功率有效提升了 15% 。

在业务支持中，我们结合 MRCP 提供了多种音色，音色更加丰富真实，尽可能满足各个业务场景。下面列举了部分实际业务过程的录音，以及前文提到的语音合成问题改善情况。

卡顿优化

播报文案为：“我是美团网的，本次来电是想确认一下商家信息，方便用户找到您家的，请问您这边是许吉凉水吗?”

| 升级前：



| 升级后：



机械优化

::: block-1 播报文案为：“非常抱歉，查询到您的订单收货已超过 48 小时，由于生鲜商品容易腐烂，因此暂不支持超 48 小时的售后退款，下次遇到问题还请您尽早反馈，给您带来的不便我们深感抱歉。” :::

| 升级前：



| 升级后：



延时优化

对话过程如下：**客服：为尽快找到合适的客服人员，请用一句话描述您想咨询的服务及问题，我的外卖到哪了？酒店如何退单？**

用户：我要退款。

客服：好的，请问是好邻居连锁便利这个外卖订单吗？

用户：不是。

客服：好的，那请问您要咨询的是达美乐比萨这个外卖订单吗？

用户：不是。

客服：为尽快找到合适的客服人员，请问您是要找美团外卖、休闲娱乐、还是酒店住宿

的客服呢?

用户：不是。

客服：好的，那请问您要找账户登录手机解换绑小美果园及手机充值问题的客服吗?

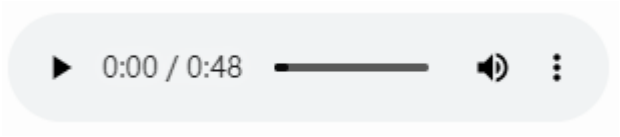
用户：我要退款。

客服：正在为您转接人工。

| 升级前：



| 升级后：



四、结语

MRCP 协议及相关技术已经相对成熟，围绕技术本身而言的迭代和演进相对较少。未来，除了语音合成（TTS）和语音识别 (ASR) 外，语音交互部自研的声纹识别（VPR）技术也逐步成熟；其中，娇喘识别效果大幅提升、目前性能已超过头部厂商服务，年龄识别误差小于 7.5 岁（公开测试集评测）。预计结合 MRCP 在电话智能等场景提供相应技术能力，支持优化和提升业务流程及效果（如金服生意贷场景，通过声纹识别判断是否为虚假申贷人，减少恶意申贷造成的损失）。

从技术运用上看，一方面 MRCP 协议下的 TTS、ASR 为美团智能外呼机器人、智能呼入机器人等多个业务场景提供稳定的服务，并带来可观的业务效果提升；我们预期将其接入更多的业务场景，以客户为中心、给每一个用户带来流畅丝滑的人机对话体验，助力业务优化。另一方面，在标准化的 MRCP 协议接口的承载下，持续推进美团 TTS

和 ASR 能力的商业化。

五、名词解释

名词	解释
RTP	RTP（Real-Time Transport Protocol）实时传输协议，适用于传输实时数据的应用程序，如音频、视频或模拟数据，可通过 RTP 的子协议 RTCP（RTP Control Protocol）保证服务质量。在 MRCP 中负责对媒体资源进行传输。
SIP	SIP（Session Initiation Protocol）会话初始化协议，是一个基于文本的应用层控制协议，用于创建、修改和释放一个或多个参与者的会话。
SDP	SDP（Session Description Protocol）会话描述协议，定义了会话描述的统一格式，与 SIP 配合使用在 MRCP 中达到创建一个 Media Control Channel 的目的。
Asterisk	用于构建通信应用程序的开放源码框架。可以将普通计算机转换为通信服务器，实现 IP PBX 系统、VoIP 网关、会议服务器和其他定制解决方案。
FreeSWITCH	一个跨平台的、伸缩性极好的、开源免费的、多协议的电话软交换平台。
ESL	Event Socket Library，外部的程序通过 Socket 方式使用 FreeSWITCH 提供的所有的 App 程序和 API 命令，控制 FreeSWITCH。
OCTO	美团基础研发团队开发的分布式服务通信框架及服务治理平台，为业务提供标准化微服务技术方案。
MGW	Meituan Gateway 缩写，是美团基础研发团队自研的四层负载均衡网关，并针对美团自己的需求做了额外的功能定制。
小美机器人平台	美团语音交互部自研的智能外呼机器人平台。
木星通讯平台	美团服体技术部自研的集人工呼叫，智能外呼功能于一体，支持快速配置接入的全自主研发通讯平台。
首包延时	从 MRCP 接到合成请求，到用户首次听到音频内容之间的时间间隔。
端到端延时	从用户侧发起合成请求到开始播放之间的延时。

六、本文作者

唐锐、森彬、子丰、亚男、王程、国桥、俞涛等，均来自美团平台 / 语音交互部。

七、参考文献

[1] Media Resource Control Protocol (2022) Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Media_Resource_Control_Protocol.

[2] 史俊波 , 詹舒波 (2010) MRCPv2 协议及其在分布式语音资源解决方案中的应用 , 中国科技论文在线 – 科技论文 . Available at: http://www.paper.edu.cn/releasepaper/content/201009-322.

[3] zhu, james (2018) MRCP 概 要 , CTI. Available at: http://www.ctiforum.com/news/guonei/534456.html.

[4] UniMRCP Introduction (no date) Home. Available at: https://www.unimrcp.org/.

[5] Kamailio Introduction (2022) The Kamailio SIP Server Project | The Open Source SIP Server. Available at: https://www.kamailio.org/w/.

前端

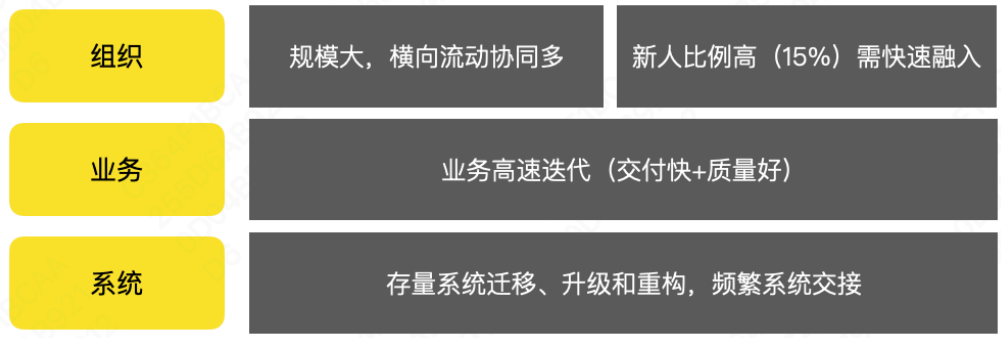
美团前端研发框架 Rome 实践和演进趋势

作者：陈伟 吴婧 永康

1. 背景介绍

1.1 业务背景

首先介绍一下业务的背景，这里主要从 3 个维度展开。第一个维度是组织维度，在立项之初，恰逢美团的多个事业群合并，因前端规模比较大，横向的流动协同比较多（需要跨部门支持需求，进行跨系统协作等等）。此外，美团到店事业群新人比例比较高，校招和新员工比例很高，我们会帮助新同学快速融入团队，需要完成一些较为基础的开发工作。

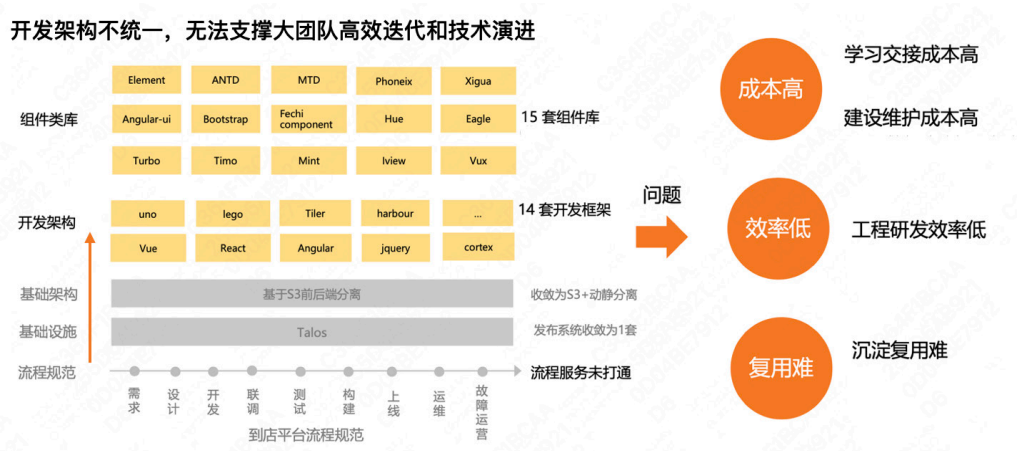


第二维度是业务维度，美团到店业务迭代频次比较高，基础工程框架不仅要保证交付速度快，同时还对质量有很高的要求。

第三个维度是系统维度，因业务周期比较长，到店还存在大量的存量系统，需要考虑迁移升级和重构等问题，同时会有频繁的系统交接。

1.2 技术背景

在 Rome 整体立项时，我们已经准备好了相关的基础设施，包括发布系统的收敛、基础架构，统一为基于 S3（美团内部存储服务）加动静分离的技术架构，但是上层开发框架、组件类库种类繁多且开发方式不统一。存在问题包括：整个团队中人数比较多，学习交接、建设维护成本相对较高，而整体开发的效率比较低，跨团队之间的工程能力也很难进行复用等等。



建设之初，我们基于纯静态 S3（美团内部存储服务）架构进行前端框架的建设，这源于我们早期大量基于 Node.js 的前后端一体架构存在一些问题：首先，事业群早期以中后台场景业务为主，对页面的秒开、SEO 的诉求比较低；其次，当时 Node.js 生态基建还没有那么完善，前端同学需要做动态扩缩容、峰值流量处理等操作，整体的业务风险比较高。同时还存在机器成本高、开发人员能力要求高、招聘难度大等问题。

因此，在整体的建设思路和路径上，我们不会建设类 Egg.js 这样的前后端一体的框架；同时因为我们的框架层要解决研发流程不规范、交付质量不高等问题，也需要联动上下游的设计研发、CI/CD 等系统形成一体的开发工程平台，而不只是做 CLI 工具。

2. 工程生态、演变路径和规模化升级

2.1 工程生态

2.1.1 降学习成本

框架约束

根据前文所述，我们一开始要解决的核心问题是学习成本，因此我们会做框架约束。

- 第一是工程目录，我们在真正开发的时候，比如 UI 框架是使用 Vue 还是 React，UI 框架本身的语法成本没有那么高，但是当拿到一个需求时，应该在哪儿改，在哪个位置梳理逻辑，这个部分的成本比较高，因为很多时候不同技术栈的业务组织方式不一样，会带来比较高的理解成本和心智负担。
- 第二个是大团队下工程能力碎片化，团队间水平比较参差，带来的问题是团队之间难以做好、做深工程能力，典型的如 SSR、构建优化等很难跨团队间复用，一直在重复造低门槛的“轮子”。
- 第三是业务工程的配置比较冗长，带来的问题是工程配置比较耗时，很多时候存量系统不敢升级，因为不知道这么长的配置加了一些东西，是否会出现比较严重的后果。
- 最后是公司基建落地，业务对接过程中非标的基建落地实践版本非常多；在公司开发和自己写小型项目有很大的差异，在公司内要解决监控、流量、部署对接等问题，需要做周期升级，而这部分如果不进行整体收敛，整体规模化升级成本就会比较高。

Rome 在这一侧的解法：

- **统一目录认知**：如下图右侧所示，在开发时，对于部门或者团队级别认可的规范，当技术同学不遵守这套规范的时候，在代码提交时会有卡控，当然也可以通过 TL 审批跳过。
- **收敛依赖选型和版本**：在业务开发时，从头进行工程框架选型的上手门槛比较高，因此在公司内将版本和选型进行统一收敛，保证大家可以专注业务。
- **统一跨技术栈工程能力和开闭配置**：保证大家拿到工程后，不管是哪个技术栈都可以以一个相对一致的认知进行开发，无需关注构建、工程能力配置等。

- **收敛基础开发规范**：默认在拿到一个项目时，如果没有特别定制化的团队规范，基本上可以不用关注像 ESLint、Prettier、EditorConfig、部署对接等开发规范。



跨技术栈开发认知一致

在实际迭代的时候，到店内部一开始整体收敛到 Vue 技术栈，后来因团队合并，又有了 React 技术栈。我们采取的第一个策略是把 Vue 和 React 整体目录保持大家认知的一致，如下图所示。而在真正开发的时候，比如配置子目录 SRC 下，当一个同学一开始是偏 Vue，但开发 React 需求的时候，即使团队和业务有一些变动，TA 也可以准确拿到项目快速启动，接下来摆在 TA 面前的可能就是特定技术栈下，对于对应的 API 理解是否深入的问题，这部分同学自己就可以解决。

第二是我们会保证使用 Rome 开发框架的整体开发调试流程和体验一致。如下图，在 Vue 或 React 框架中同学专注业务开发就行，基本上可以不用关注工程框架的相关事情。比如 Vite 在 Vue 或在 React 指令和表现是一致的，同学可以通过一行命令直接开始调试对应项目并进行开发。



第三，我们的整体工程能力会进行高度对齐，如上图所示。我们生态下 80% 的插件在 Vue 和 React 下，用法和表现完全一致，大家基本上不用关心这部分，当真正用到某部分能力，详细看对应的文档就行。

基础基建对接

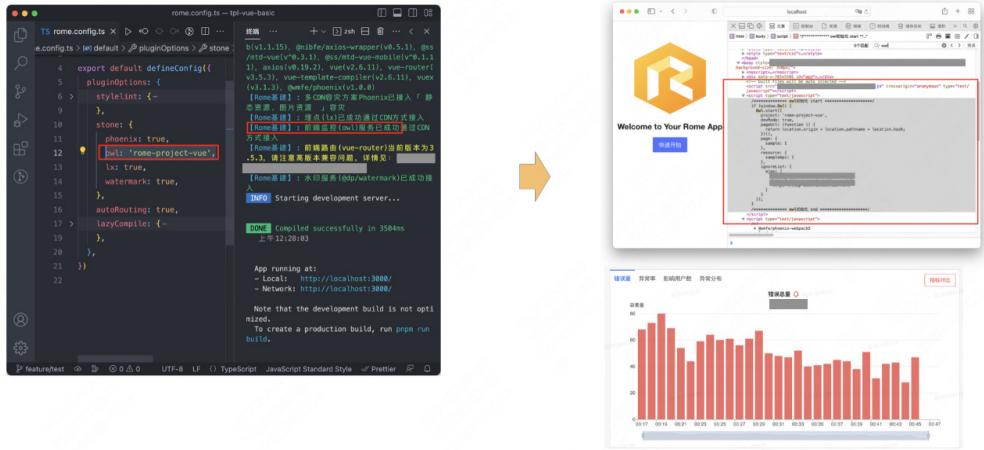
工程能力：一行代码引入公司基建（CDN 容灾）

当前端同学处理如 CDN 厂商故障等问题时，需要自行查找、学习和使用公司 / 开源的基建能力，解决成本较高，但在 Rome 开发过程中可以通过开闭式的配置一键 CDN 容灾生效，如下图，接入后会自动进行静态资源的降级重试，当业务出现故障时（即下图峰值部分），它会有一个明显的资源加载重试，这就是降级 CDN 在生效；对业务同学来说只需配置框架需要哪个能力就行，其详细配置如何注入大家并不需要太关注。



工程能力：一行代码引入公司基建（告警监控）

业务开发以外的像告警监控，对同学来说，可能也要理解很多东西。但是在企业内开发，我们希望同学可以专注业务开发，像线上告警、日志链路等出现问题，可以不用配置对应平台，我们用对应的项目 Key 就可以查到对应的错误总量、错误调用链路等。



工程能力：一行代码引入公司基建（水印）

我们也可以一键完成对应的水印接入，同时开发配置时有对应的配置项智能提示。各种开闭能力的属性命名语义可能没那么清晰，可以通过如 VSCode 中的智能提示一键跳转到对应能力的文档地址，查看它的实现原理以及使用方法。

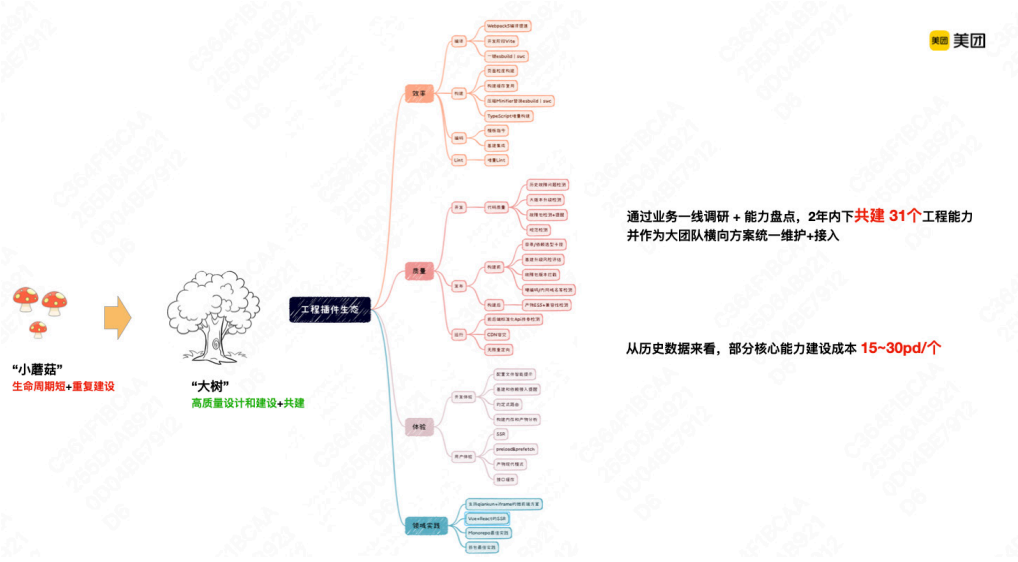


2.1.2 降建设成本

之前组织下建设的很多能力是偏小蘑菇的形式，“小蘑菇”的语义是整体工程能力的生命周期比较短，会进行频繁的重复建设。但我们希望以一个大树的形式建设，“大树”的语义是我们可以通过跨团队、横向方式进行高质量的设计和建设，通过评审委员会把控整体的方案建设，把方案拆分到几个阶段，横向团队共建，共同完成核心能力，比如Serverless、SSR 等。

工程能力生态

如果大家对工程框架的建设能力感兴趣，可以参考我们内部的效率、质量、体验和领域实践进行对应分类的建设。我们通过一线调研、能力盘点的方式建设了 31 个，在企业内进行建设时，有很多能力一开始可能本身成本就很高，比如统一基建接入等，大概需要 30 人力 / 个，而实际业务需求开发一共只有几天时间，很难说服产品运营单独抽出大量人力开发前端基建能力，这部分我们就以**共建 + 横向评审**的方式，保证**业务团队可以摊平整体的建设成本**。



典型共建案例

Serverless SSR 能力，在休娱频道的密室和优选团长端的订单管理流量分别是百万级别和千万级别，整体秒开可以做到 90% 左右。

美团/休娱频道/密室H5

流量：PV百万级

秒开：>90%



优选/团长端/订单管理H5

流量：PV千万级别

秒开：>85%



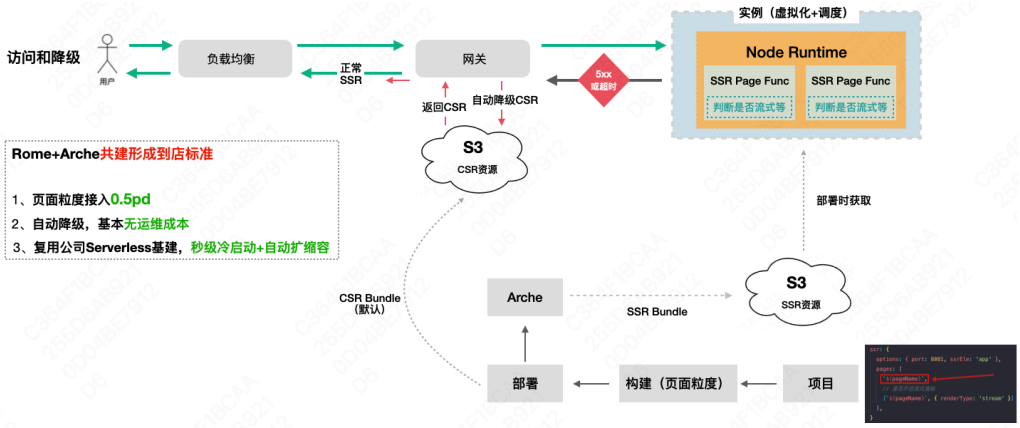
整个过程是 Rome + Arche 团队共建并形成到店标准，业务统一落地，过程中有超过 3 个业务团队参与了部分能力建设。

这里简要介绍下 Serverless 的方案流程：

首先是访问自动降级，用户通过负载均衡命中对应的网关，命中对应的 Serverless 的 Node Runtime，这里会有对应页面粒度的 SSR 渲染函数，当用户访问服务器返回 5.x.x 状态码或超时时会自动降级到 CSR 资源（SPA 页面）；即使出现了极端情况，比如 SSR 服务异常，我们也可以保证整体页面可用，差别主要在 SSR 场景秒开 ~90%，CSR 场景秒开 ~30%，兼顾了性能和稳定性的优势。

其次是支持页面粒度接入，在大型 MPA 项目开发中，可以把某个页面单独设置为 SSR，不必要求整个项目都是 SSR 工程，构建时默认会上传 CSR 端构建资源，而针对 SSR 页面会额外进行另外一份资源的构建，在部署 SSR 端、CSR 端资源时分别部署，用户访问页面链接时，对应命中 Server 端或者 CDN 资源。

支持自动扩缩容和流失渲染，对接公司 Nest 基建，支持业务无感知的峰值流量自动扩缩容；同时长列表页面也可以通过流式渲染来降低 TTFB 时间。



2.1.3 提研发效率

编译提速主要分两个方向进行业务落地，分别是 Webpack 体系和 Vite 体系。

编译提速 – Webpack 体系优化

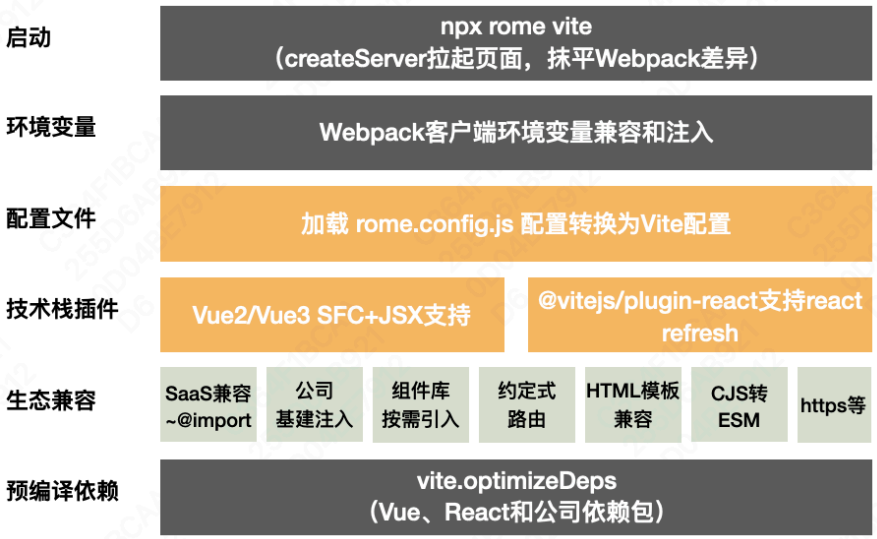
首先是 Webpack 体系，因为我们整体的技术栈分 Vue 和 React 两部分，我们会抽一部分基础的跨技术栈的 Webpack 配置，上层有不同团队分别维护 Vue 技术栈和 React 技术栈的特有配置，中间我们会进行专项优化，包括像 Webpack5 的开发优化、编译压缩器、SWC 和 esbuild 等。

另外，我们做工具时，不仅要考虑我们本身构建工具比如 Webpack，也要考虑如何和我们的 CI/CD 平台结合，如缓存复用结合依赖安装、构建工具做，在很多需求场景下，可以获得耗时的 10 倍提升，CI/CD 侧我们也支持分粒度的构建，比如大型 MPA 做页面粒度的构建等。



编译提速 – 开发时　键 Vite

Vite 主要是我们在开发时使用的，Webpack 是构建时使用，除了少部分增量项目的开发和构建都走 Vite，大多数存量项目都是开发使用 Vite 保证效果，构建阶段使用 Webpack。



整体流程：首先是启动，通过 Vite，比如 createServer 拉起页面，抹平 Webpack 差异；环境变量的抹平，保证客户端 Webpack 存量项目，将 Webpack 客户端的环境变量注入到 Vite 端；对配置文件，大家原本在 rome.config.js 中的 Webpack 配置也可以一键转换为对应的 Vite 配置；在下层我们会做内置的技术栈插件，包括 Vue2/

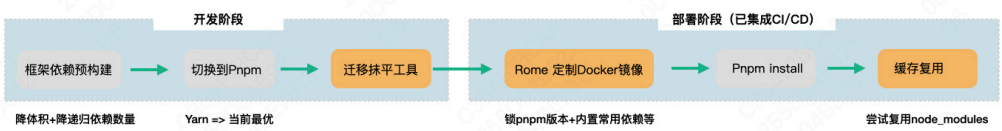
Vue3 SFC+JSX，包括 react 的内置插件等；比较繁琐的是生态兼容，在公司内这部分的工作量比较大，像我们的一些 SaaS 的静态资源的一些路径等，包括之前我们存量 Webpack 这么多年积累下来的公司基建，我们如何保证它能被注入到 Vite 端，包括组件库按需引入，当一些模块比如 common.js 的模块如何指向对应的 esmodule 目录等，最后我们会屏蔽一些认知成本，把 Vue、React 和公司依赖包内置到 Vite 预编译内容中。

依赖提速

主要分两部分，分别是开发阶段和部署阶段。

首先会做工程框架自身 Node.js 依赖的预构建，这部分核心解决的是 Node.js 端公司内外包资源的体积和递归依赖数量庞大的问题。依赖安装的一个核心耗时其实是递归依赖庞杂，需要逐个安装带来的耗时，比如需要安装 A，其实 A 依赖了 B、C、D、E、F、G，整体安装耗时就会非常久，同时研发同学对 Node.js 包体积、依赖数量一般不如面向浏览器端投放的 NPM 包那么敏感，这个时候预编译就很关键。

其次是把整体的依赖环节包管理工具切换到 Pnpm。这部分经过调研（Pnpm、Yarn Berry、Yarn Berry with PNP 等），判断 Pnpm 是未来几年的核心趋势，同时可以稳定发展。内部建设了对应的包管理迁移工具，来服务企业内几百个、上千个项目。



在部署阶段定制了 Rome 依赖安装 Docker 镜像，锁 Pnpm 版本和内置常用依赖、缓存复用，它的整体逻辑是上一次需求迭代和下一次需求迭代可能依赖包的变化没有那么多。我们在第二次发布的时候，如果能够尝试命中上一次的 node_modules 里的一些包，那第二次安装耗时就会有一个比较大的提升。

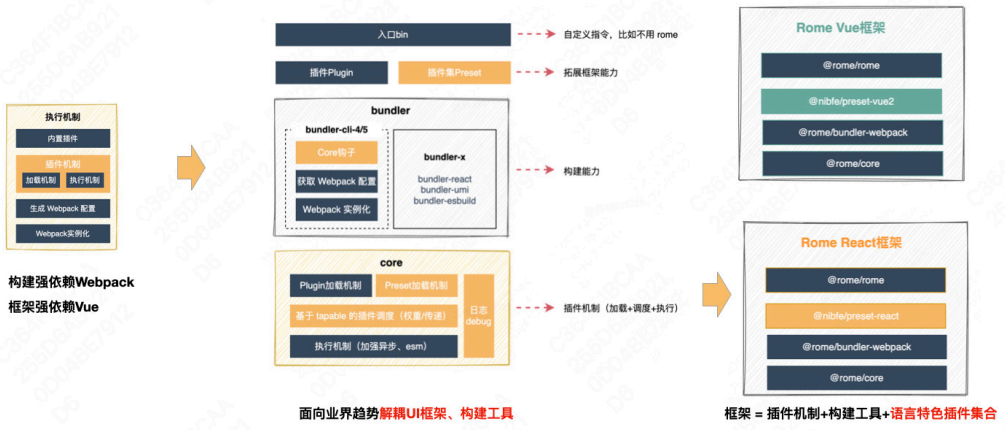
2.2 演变路径

第一阶段的时候是强依赖 Webpack 和 Vue 技术栈的。直到 2020 年 -2021 年，像社区 UI 框架有 Solid.js、Svelte、Vue3，构建工具有 Esbuild、Rspack、Vite 等且发

展势头也比较好，面向这个大趋势，我们希望研发框架能够分层过渡，不再和具体的 UI 框架、构建工具绑定：

1. 首先最底层是差异机制，核心做插件的加载和调度；
2. 中间层是构建工具的 bundler ，这一层主要是分别结合 Webpack、esbuild、Rspack 等构建实现，支持大量项目；
3. 上层 Plugin、Preset 做框架拓展，各个业务能够基于场景定制自己的工程能力；
4. 顶层是品牌指令（bin），这部分主要是企业内合作时业务方希望保持自己的品牌名。

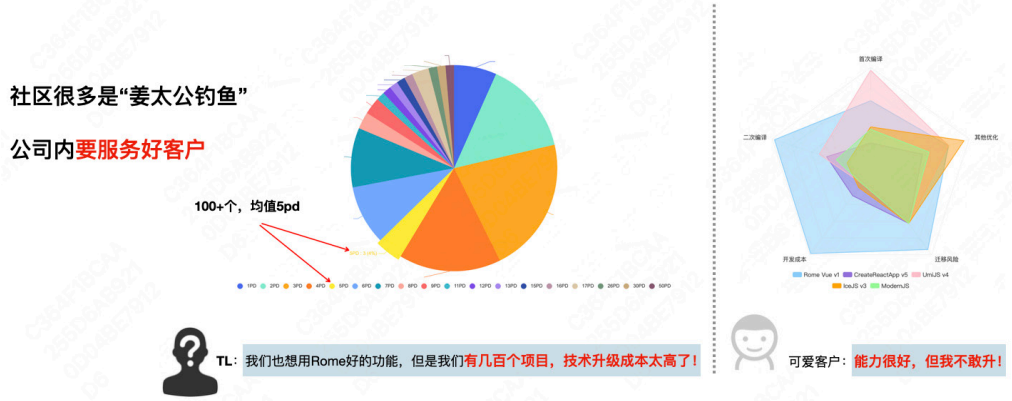
示例如下图右侧，大多数情况下，我们有一个新的研发框架出来，只需要开发我们对应的语言特色的插件集合就行，底层插件机制 + 构建工具大家其实可以保持一致的。



2.3 规模化升级

框架价值 = 覆盖范围 * 功能价值。我们的共识是建设的能力需要落地才能产生价值，核心是帮助大家业务低成本地享受好用的能力。

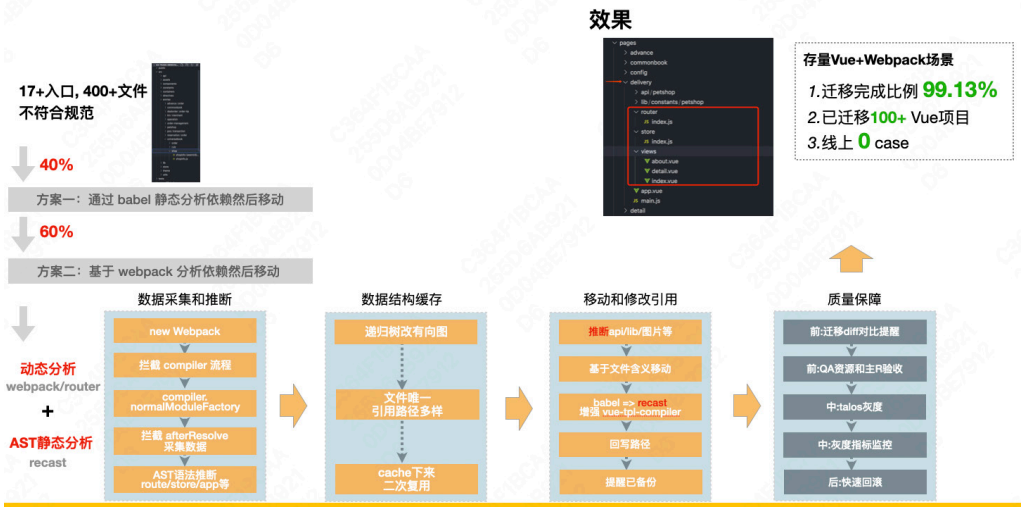
如下图所示，是某个业务的升级成本图，可以看到有 100 多个项目（成本均值 ~5pd/项目），业务面临几百天的成本。对团队 Tech Lead 很难说服业务产品运营业务阶段投入升级。对一线同学来说，虽然能力很多、很棒，但是升级风险怎么处理：编译提升 10s，线上问题定位消耗一周。



这部分的我们给两个解决方案。

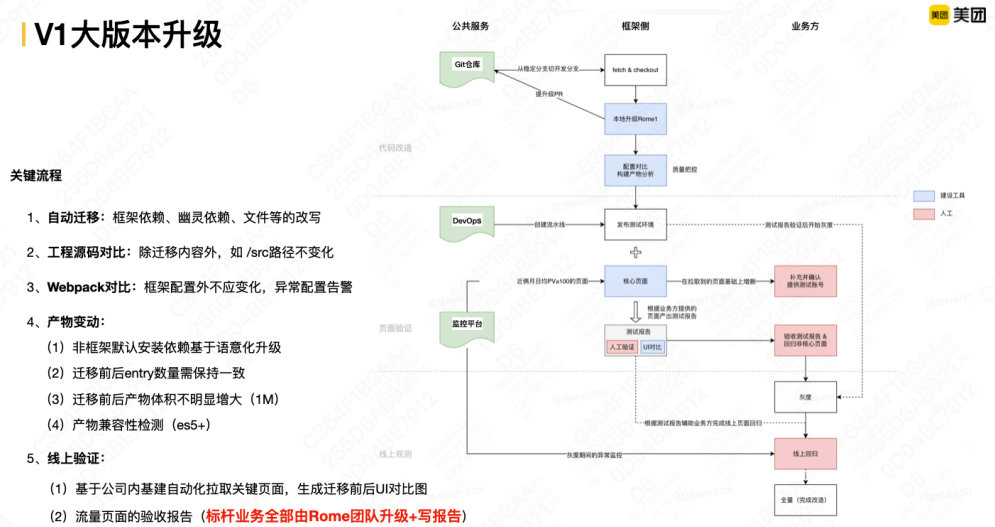
首先是提供迁移工具（非 Rome 到 Rome 项目）：

1. 巨石项目做存量迁移时，一般同学下意识反应是通过 Babel 做静态解析迁移，纯静态解析实测的迁移准确率在 40%，也就说一个项目 40% 的事情通过 Babel 可以搞定。
2. 第二阶段是基于构建工具的依赖引用关系去做迁移，通过构建工具启动时的模块工厂钩子去生成依赖引用树状结构然后做迁移，得到迁移准确率在 60% 左右。
3. 第三阶段是动态分析结合 AST 静态分析一起去做，在数据采集阶段通过构建工具的模块工厂钩子、Vue/React 实例化对象的路由 / 数据流配置去生成依赖图结构，通过 recast 等工具进行静态解析，不用 Babel 是因为 recast 有比较好保持样式的代码回写，文件内容增删改后写回原文件样式不乱，同学在 git diff 时需要 Code Review 代码就可以少一些，便于迁移影响范围判断。
4. 最后配备上基础公司基建做的一些依赖版本、体积、兼容性检测的质量工具，自动迁移了差不多有 100 多个项目，线上零 Case。



另一个是做框架生态的大版本升级，假如一个季度推出来一个大版本，实际业务开发升级难度、成本就会比较大，业务可能也不敢升级，这时大版本升级就要提供关键流程自动迁移支持：

1. 当从 Yarn 到 Pnpm 的时候，会做幽灵依赖检测，自动向 Pnpm/src 的幽灵依赖配置等；
2. 也会做工程源码对比；
3. 构建配置，如 Webpack 前后框架无关的配置对比；
4. 产物变动升级前后 entry 数量是否不一样、产物有没有明显变大；
5. 兼容性检测，升级前是 ES 5+ 的代码，升级后到 ES 6 了，那也会出问题；
6. 线上验证等。



公司内部的一些标杆团队都是由我们 Rome 团队升级 + 写报告，标杆业务由基建团队升级的话，ROI 可能会比较高，我们升级通过工具将我们的二次调整，整体成本可能会非常低，2~4h 就可以完成一个中等存量项目，再由核心业务同学验收、线上灰度跟进等。

3. 框架开发辅助

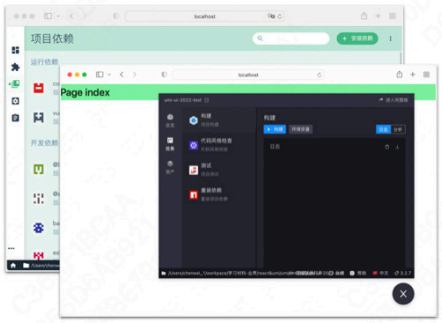
我们 Rome 的工程开发辅助工具是基于 IDE 的。一般业界的开发辅助工具两种形式，下图是 Vue UI 和 VSCode 的拓展图示：

- 一是通过 Web 实现，典型案例有 Vue UI/Umi UI/Angular Console 等；
- 二是 VSCode 拓展，如 Ice Works 等。

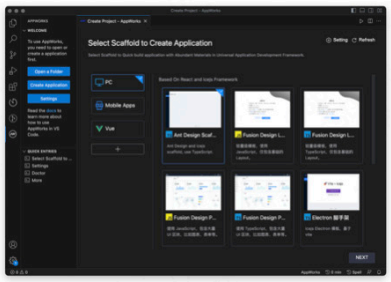
3.1 为什么要基于 IDE？

业界工程框架开发辅助一般两种形式：Web（Vue UI/Umi UI/Angular Console）、VSCode拓展（IceWorks）

Web



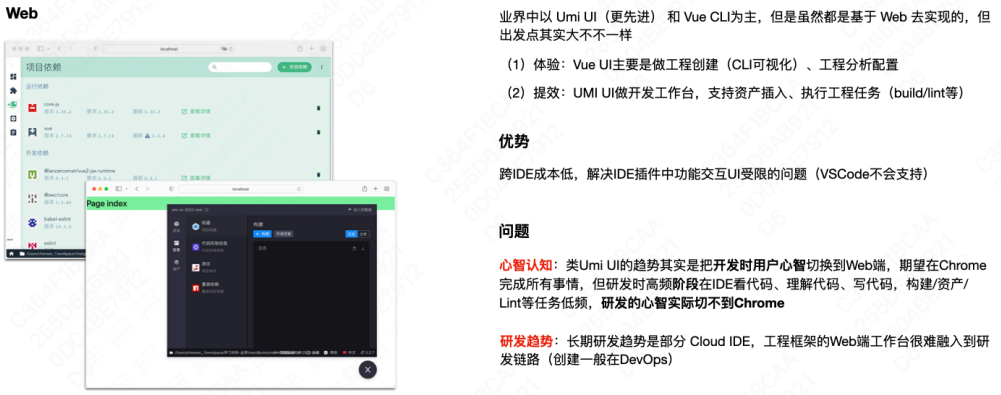
VSCode拓展



理解：对于 Web 形式，业界以 Umi UI（更先进）和 Vue CLI 为主，虽然它们都是基于 Web 实现，但出发点不太一样，Vue UI 核心是做体验提升，支持工程创建 CLI 可视化，工程配置分析等；但是 UMI UI 要服务公司内部，更多的是做提效，支持一些组的资产插入、执行工程任务比如 build/lint 等。

基于 Web 做开发辅助工具

- **好处**：跨 IDE 的成本比较低，前端同学覆盖会比较全，也可以解决 IDE 内功能交互 UI 受限问题。
- **问题**：VSCode 不开放这部分的 UI 侵入能力，一方面心智认知类 Umi UI 的趋势其实把开发时的用户心智切换到 Web 端。但我们研发时，其实要在 IDE 看代码、理解代码、写代码，当我们热更新起来以后，像构建、物料插入、Link 等其实比较低频，就会带来一个问题，我们研发时的心智很难切到对应的浏览器端。另一方面研发链路，长期来看，不管是业界还是美团内部的研发趋势中就一部分场景，后续肯定是用 Cloud IDE，但是我们工程框架的 Web 端这种开发辅助工具，如果通过 Web 形式实现，就很难融入到我们的研发链路中。

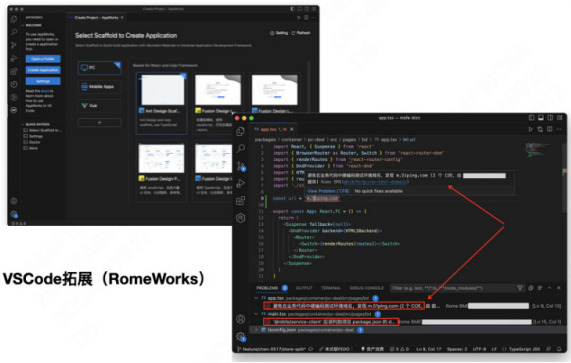


IDE 侧方案确实存在 UI 交互受限、视窗面积、跨 IDE 开发成本等问题，因为同学在开发时，VSCode 的面积就这么大，每弹出一个 Webview 就会侵占用户的开发视窗，会比较难受。

选型原因：

- **用户频次**：我们核心是企业内做提效，那就决定了绕不开 IDE 侧，而且 TA 是用户高频部分，而且我们在代码侧也可以复用 VSCode、IDE 侧的 LS API，交互受限通过用户引导是可以接受的。
- **研发整合**：我们后续可以在 Cloud IDE 整体比较成熟的时候，会把部分轻量迭代切换到 Cloud IDE 中进行研发，同时希望新模式下开发辅助工具可以继续产生价值。第二是可以在 IDE 侧整合上下游链路，保证用户心智可以一直停留在 IDE 内而不用跨平台操作，整体成本可能会比较高。
- **快速落地**：不管是 Vue UI 这种全局 CLI 的形式，还是 Umi UI 这种跟随项目的 NPM 包形式，都有一个比较大的问题是版本碎片化，我们整体 Node 的生态社区，NPM 包版本的不稳定性，对同学来说是认知比较强烈的一个事，功能迭代后，让大家进行 NPM 包版本升级难度会比较大，而且同学会比较慌，而 VSCode IDE 侧的进行发布，它是可以进行动态版本下发的，保证我们大范围落地，这一部分的兼容成本就由我们框架团队内部消化，做一些兼容。

VSCode拓展（IceWorks）



VSCode拓展（RomeWorks）

IDE拓展问题

UI交互受限 + 视窗面积 + 跨IDE开发成本

选型原因

用户频次：

1）看、理解和编码是研发阶段最高频的事情，也是工程中最具提效价值环节，这部分绕不开IDE，且IDE提供大量LS Api

2）交互受限通过用户引导弥补，可接受

研发整合：

1）在研发趋势上可以通过 IDE 拓展形式同步 Cloud IDE

2）可以在研发时通过Webview+Api的正式整合研发系统，用户流程操作心智可以一直保留在IDE内

快速落地：

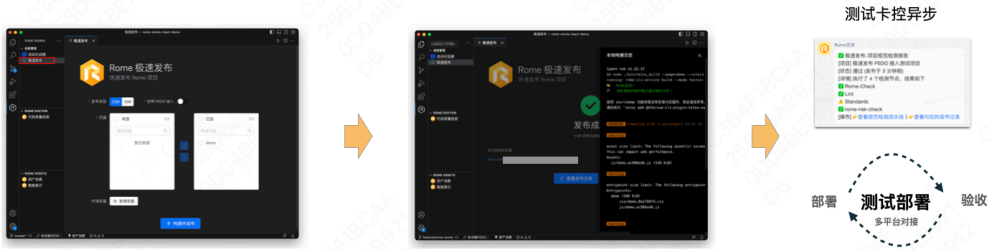
Vue UI、Umi UI采取的NPM包形式都有较严重的版本碎片化问题，难快速推广大范围落地（如中期增加打点和新能力）

3.2 提效率

提效率的一个例子是“一分钟内部署”，这里主要解决我们高频的测试环境部署流程冗长问题。2021 年我们 Rome　线上环境发布　概 10W 次 / 年，核　流程 369s/ 次，如下图所示：

1min部署

主要解高频的测试环境部署流程冗长问题（2021年Rome非线上环境发布大概10W次/年，核心流程369s/次），节省40~75%



页面粒度CSR/SSR发布　　构建环境模拟和实时日志　　同步并流转DevOps节点

通过我们的开发辅助工具，可以做到页面粒度的 CSR/SSR 发布，同学不需要关注跨平台之间有哪些工作量，我们会在背后同步并且流转 DevOps 节点，把原本我们要在研发流程中，企业内要做质量分析、规范卡控等部分做异步，然后异步进行触发，整体的链路就可以保证一分钟内完成部署。

3.3 提质量

IDE 可以做开发时质量检测，开发时会有历史的线上故障提醒。

哪些问题我们之前已经出过很多次线上故障了，在开发时就会实时检测代码并提示这部分应该进行修复，否则后续可能会出现线上故障。

它和 ESLint 的区别：

1. 首先是我们可以做规则的动态下发，研发规则配置，做管理中心化；
2. 内容上我们也会支持故障包版本，比如业界之前出过几次故障包问题，包括 core-js、内部组件库故障版本等；
3. 我们也会把历史业务线上故障分析出来并提示，点击链接就可以跳转详细查看之前具体是出了哪次线上问题，是由什么引起的。

质量检测

开发时激活窗口检测、PR时diff代码，**巨石工程也秒级**

质量检测和ESLint的区别

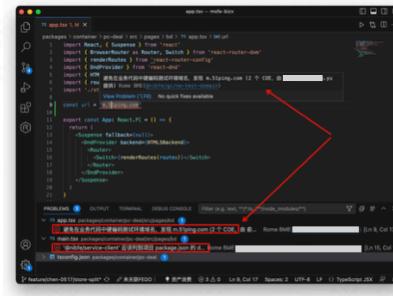
规模治理：规则动态下发 + 研发rule管理中性化

内容不同：包含ESLint，同时增加了故障包版本、兼容性、历史业务线上故障分析链接

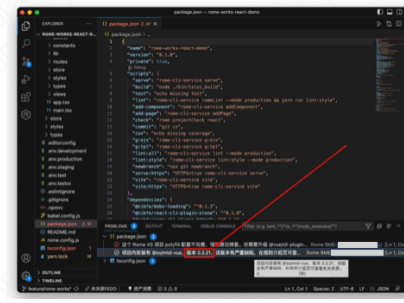
定位不同：ESLint侧重静态语法检测（如未定义就使用），**质量检测则侧重历史故障检测+不重复发生**

时机不同：质量检测包含开发时+PR时，更前置、环节更完善

COE检测



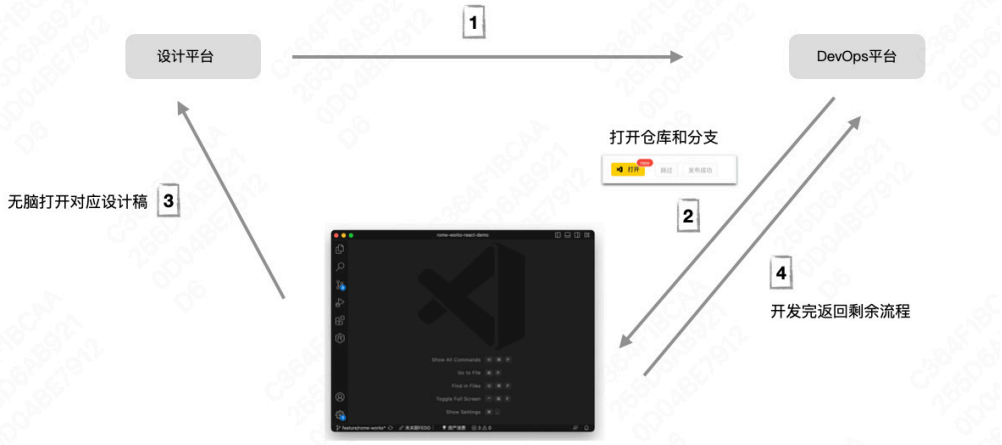
故障包检测



3.4 平台流转

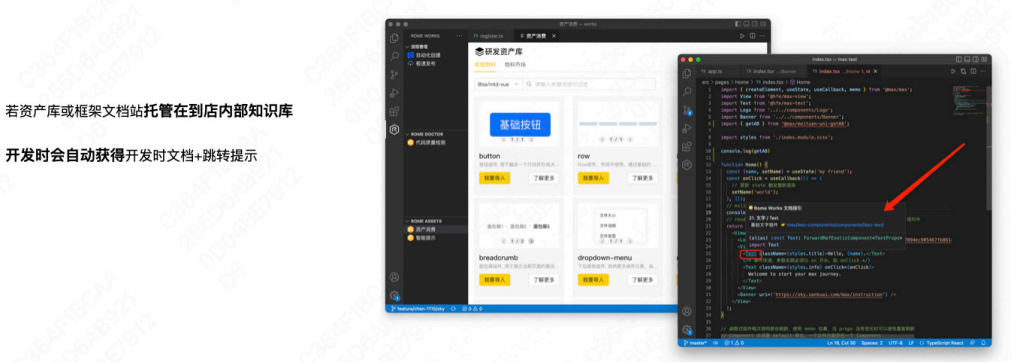
我们内部研发时不仅要开发，还有设计稿，包括 DevOps 平台的流转等，也会上下游联动，保证设计可以快速到 DevOps 平台，DevOps 平台可以一键打开本地 VS Code，记忆历史的仓库和分支并快速打开，也可以快速流转到设计平台，开发完返回

剩余流程。



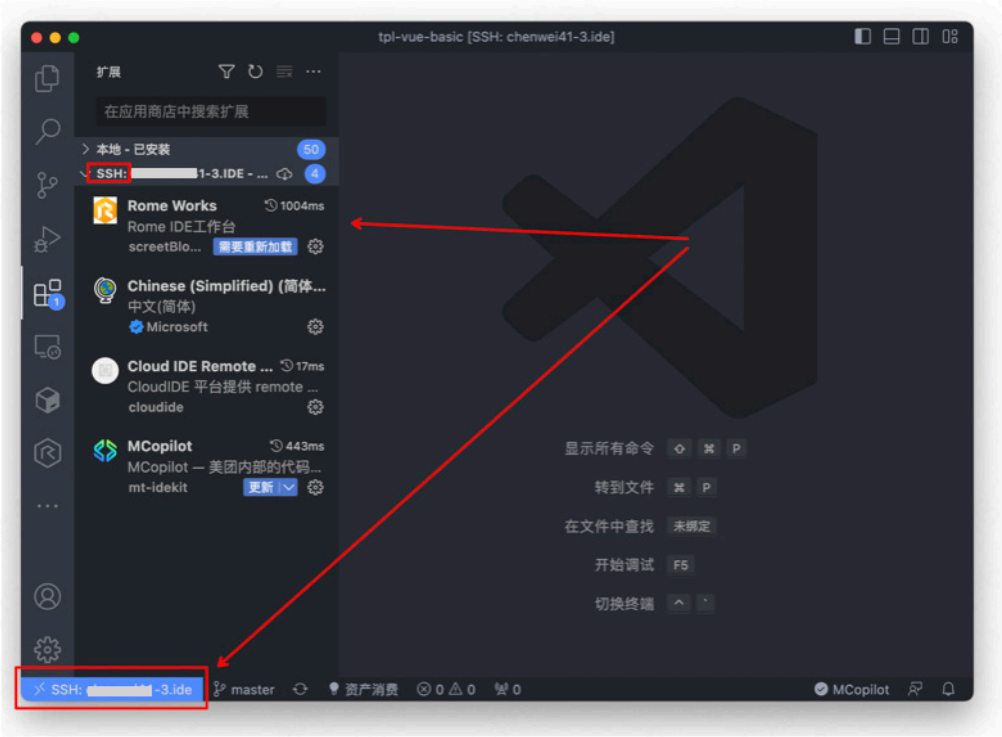
3.5 文档提示

开发时也会给到大家文档提示，只要研发人员托管到到店内部的知识库，就可以在开发时，自动进行相关文档的匹配和跳转提示。



3.6 CloudIDE 对接

得益于业界 IDE 标准的高度统一，一般 CloudIDE 仅需一次开发，即可本地、云端 IDE 自动安装开发辅助，能力完全一致。



4. 框架度量和业务实践

上文中从接入公司基建、编译提速、框架大版本升级等方面提到了很多工程能力，过程中也建设了升级、迁移工具，而作为企业内框架，也有一些问题和指标需要考虑。

4.1 框架度量 – 核心问题

问题和指标

- **标杆业务支撑的怎么样、哪些是我们的优质用户、用户规模是多少?**
- **哪些能力用的多、哪些能力用的少、哪些是安装了但是没用起来的?**
- **做为一个 SaaS产品，如何去量化价值?**
- **下个阶段要核心提升什么，紧跟社区、覆盖率、业务重点场景支撑?**
- **我们的客服成本是多少，是否要降低、如何降低?**

【覆盖率】 需求覆盖、项目覆盖、人员覆盖率

【使用率】 核心能力开启率、能力使用率

【收益点】 人力节省、工程基建复用、风险识别和拦截、降风险运营治理成本

【客服】 客服量、每周客服人力、重复TT重复客服

在业务规模不断扩大、能力落地覆盖范围及用户数量都在不断增加的背景下，框架现状衡量、框架如何发展等问题逐渐显现：

Rome 的用户群体从纯 B 端业务到覆盖 B&C 端业务，业务节点已覆盖 4 个事业群、15 个事业部；对比 2020 年，项目规模已经扩大了 11 倍，公司内共落地 1000+ 项目。

如何判定标杆业务、框架能力支撑效果，如何持续快速统计用户规模并做运营?

框架价值 = 覆盖范围 * 功能价值。能力开发完我们需要尽可能清晰化并提升落地覆盖，在统计运营上可以通过 CI/CD 平台发布过程数据、Git 仓库扫描数据、框架运行时数据打点、IDE 使用数据来做框架能力和用户规模数据采集；同时结合标杆团队、标杆业务以组织维度框架生态能力最大提效数据，并对能力验证和优化有正反馈，促进能力高质量、更大规模落地

哪些生态能力使用频率高，哪些能力使用频率低，哪些是安装在项目中但没有实际使用的?

面对已建设的生态能力，如果纯从能力开发视角出发、缺少有效的衡量体系，会导致有限的人力会被分摊到非常庞大的工程体系内：(1) 已有的好能力因为没有度量数据，感知弱，在人员变动等客观背景下没有得到持续足够的推广导致好能力没有获得足够的落地，导致收益低于预期。(2) 初期能力建设完后产生持续的维护成本，对部分使用率低的能力无法感知，未及时做出关键判断，产生一部分人力浪费。(3) 对存在优化空间的能力，由于未及时拿到使用数据，不了解落地问题，能力未得到持续验证和优化而导致收益不达预期。

框架如何从产品角度量化价值?

从人力节省、质量、缩短交付周期等方面建立有效的产品价值衡量体系，对齐框架年度输出指标并持续运营。

下个阶段框架发展方向和重点是什么?

不变的是要深入结合业务架构重点支撑阶段核心业务场景，紧跟社区做一些先进能力，同时需要通过数据来辅助判断阶段重心是提升框架核心能力覆盖率，还是要建设新能力。

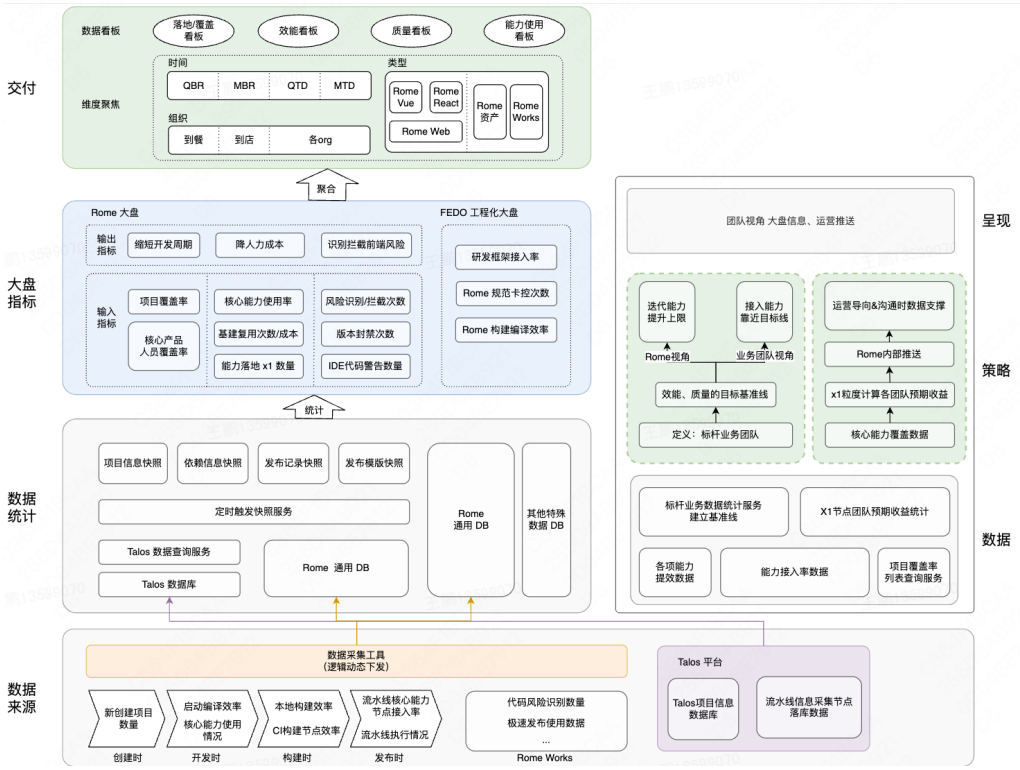
框架面向用户群体的客服成本是多少？如何降低？

收集客服数据，沉淀高质量的 FAQ，保证增量问题不被重复客服，引入 AIGC 智能客服等方式降低增量能力客服成本。

4.2 框架度量 – 全景建设

解决思路

统一定义并梳理指标来源，结合上下游平台提取数据、对缺失数据采集并持久化，通过快照等形式完成数据层聚合和差异抹平；中间层会按照定义的大盘指标对数据进行统计拟合校准，包括项目覆盖率、核心能力使用率、接入率、研发环节效率等等；上层会按不同的业务部门、时间区段等提供可切换的覆盖、效能、质量等数据看板。



数据来源层：

- 研发过程数据持久化：覆盖研发全流程（工程框架内、插件内、CI 平台流水线节点内）

- 生态产品数据埋点：Rome Works 质量检测功能埋点等
- Talos、Fedo 平台数据库

数据统计层：

- 多平台数据 Database
- 关键信息月度快照：支撑多维度看板的数据二次处理

大盘指标层：

- Rome 大盘指标查询服务：核心包含输入输出指标的查询服务
- FEDO 工程化大盘指标查询服务：框架接入率、平均编译构建效率等指标

交付（各模块的数据看板）：

- 月度数据趋势对比
- 可切换到餐 / 到店等组织维度的数据查询展示
- 技术栈等细粒度分类的数据查询展示

运营模块：

- 数据：覆盖率指标的项目列表查询服务；X1 节点粒度的数据统计服务
- 呈现：大盘中增加标杆团队数据图表；增加 X1 节点的团队预期收益统计查询服务

4.3 业务实践

整体 Rome 业务实践如下图所示：截止 2022 年，落地了 1400 多个工程项目，资产库的数量有 100 多个，到店多数 Web 项目是用 Rome 开发，大家打开美团 App，可以看到到店业务 H5 页面都是通过 Rome 进行开发的，有 B 端系统、C 端的 H5，还有我们美团内部的 React Native。



5. 总结及工程框架趋势思考

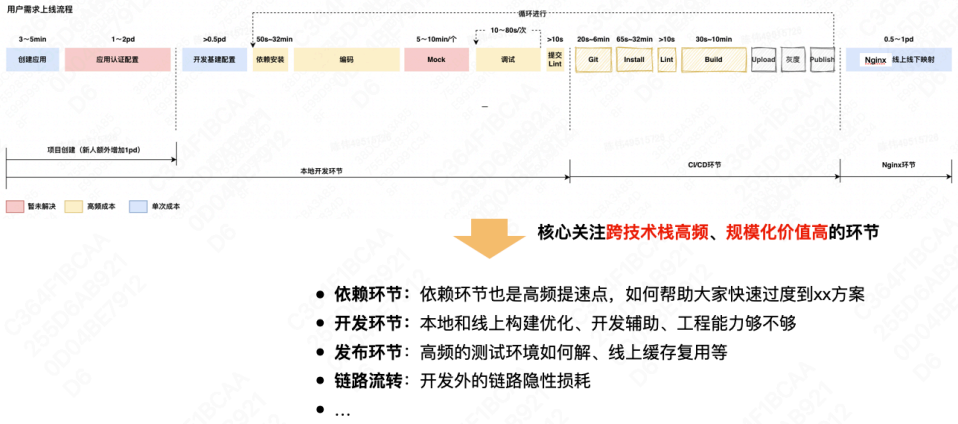
工程框架不止于 CLI。要从需求交付的视角做框架，整个工程链路可能涉及到应用创建、应用认证、开发基建配置、依赖安装、编码、Mock、调试，包括提交 Lint、还有 Git、Install、Int、Build、Upload、灰度、Publish 等等。

我们要核心关注高频的、规模化、价值比较高的环节：

1. 依赖安装环节，因为本地开发和线上部署都会使用到，如果这部分是当前核心的耗时点（发布总 300s，依赖安装 100s），下个阶段构建就可以先放一放，把依赖环节的优化提上日程；
2. 开发编译环节，很多时候开发环节我们提升了 50 秒，看起来好像没有那么高，但我们团队如果是 1000 人有几千个项目，调试编译频次是很高的，全量落地后规模化效果就会非常强；
3. 发布链路层面：交付链路中涉及整个研发的全部上下游，单纯做单点的工程优化（如本地构建优化）效果不一定好，可能到后来只是在卷几秒的差异。部署阶段依赖、构建缓存复用可能可以直接降低 10min/ 次；另外部署时流水线节点异步触发、测试环境聚焦提速快速看到效果，线上再补充管理所需的质量分析、卡控等
4. 链路流转环节：工程框架和上游的设计协作平台怎么流转、下游如何和部署平台联动，他们的耗时可能也比较高，这部分也需要关注，因为对使用我们这套工

具的同学来说，不关注这个能力是谁提供，而是我们在开发需求的时候，他明显会感觉到你们整体提供的这套产品，我用起来很难受，可能不一定是工程框架部分，单体都不错，一起用就是不够顺滑。

不止于CLI（从需求交付视角做框架）



未来趋势（个人阶段视角）：

1. 第一方面是工程框架要做开发链路的深度整合，可以类比业界的 Vercel，我们企业内也可以类比这些成熟的像 SaaS 产品去做；
2. 第二方面是此前提到的 Rust 基建、构建切换 Rspack/Vite、辅助开发工具，保证生态完善，公司内可以平滑升级；
3. AI 时代：编码 Copilot；如何结合公司内 AI 基建把历史业务模板、组件库、资产库做提示生成是一个可以结合的点（框架有相对严格的资产生产结构规范，可以提升下游编码时资产消费准确率）；通过知识库问答降低技术项目的客服成本等。

Robust 2.0：支持 Android R8 的升级版热修复框架

作者：常强

1. 背景

美团 Robust 是基于方法插桩的实时热修复框架，主要优势是实时生效、零 Hook 兼容所有 Android 版本。2016 年，我们在《Android 热更新方案 Robust》一文中对技术原理做了详细介绍，主要通过给每个方法插入 IF 分支来动态控制代码逻辑，进而实现热修复。其核心主要有两部分：一个是代码插桩，一个是自动补丁。

- 代码插桩这部分随着 Javassist、ASM 工具的广泛使用，整体方案比较成熟了，迭代改进主要是针对插桩代码体积和性能的优化；
- 自动补丁这部分在实际使用过程中一直在迭代，跟业界主流热修复方案一样，自动化补丁工具作制作时机是在 Proguard 混淆之后，由于 Proguard 会对代码进行代码优化和混淆处理，在 Proguard 后制作补丁能够降低补丁生成的复杂性。

近年来，Google 推出了新的代码优化混淆工具 R8，用于取代第三方的代码优化混淆工具 Proguard，经过多年功能迭代和缺陷改进，R8 在功能上基本可以替代 Proguard，在结果上更为出色（优化生成的 Android 字节码体积更小）。Google 已经在新版本的构建工具中强制使用 R8 ，国内外已有多个知名 App 完成了 R8 适配并上线，比如微信 Android 在今年正式从 Proguard 切换到了 R8（通过升级 Android 构建工具链）。Android 热修复补丁制作依赖二次构建包和线上包对比，需要对 Proguard 切换到 R8 提前进行适配和改造，本文分享了美团平台技术部 Robust 在适配 R8 以及优化改进中的一些思路和经验。

2. 主要挑战

Android 热修复补丁的大致制作流程：首先基于线上代码进行逻辑修复并二次打包，然后补丁生成工具自动比较修复包和线上包的差异，最后制作出轻量的补丁包。因此在补丁制作的过程中，需要解决两个主要问题：

- 对于没有变动的代码，如何在二次打包时保证和线上包一致；
- 对于本次修复的代码，如何在经过编译、优化、混淆之后准确识别出来并生成补丁代码。

要解决这两个问题，需要对 Android 编译和构建过程有一定了解，弄清楚问题产生的原因。下图 1 是一个 Android 项目从源码到 APK（Android 应用安装包）的构建过程（椭圆形对应构建工具链）：

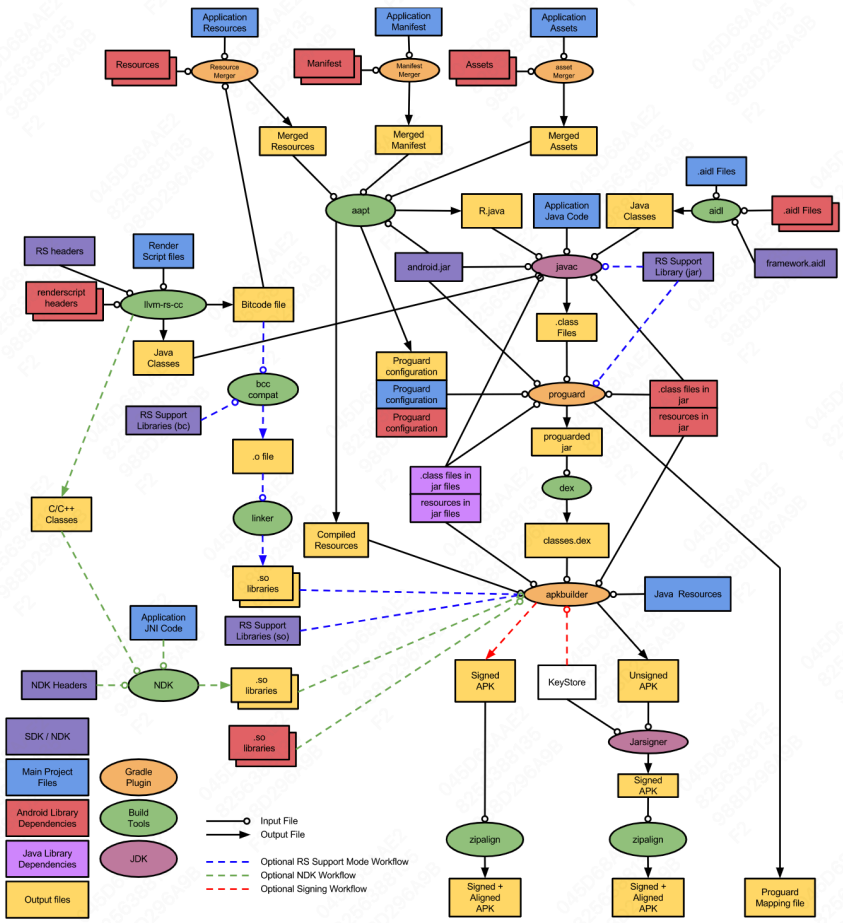


图 1　从源码到 APK 的构建过程

上图有些工具已被新出现的工具所取代，但是整体的流程并没有太大变化。对照这个图，我们分析一下其中对补丁制作 / 二次打包有影响的几个环节：

1. **资源编译器（aapt/aapt2）**：资源编译环节会生成一个 R.java 文件（记录着

资源 id，便于代码中引用），一般为了解决 R field 过多以及减少包大小，大型 Android 项目会在构建过程中会将资源 id 直接内联到调用处（发生在 javac 和 proguard 之间）。如果前后两次打包出现资源 id 不一致，会影响 diff 识别的结果。

2. **代码编译器（javac）**：Java 代码经过 javac 编译成字节码之后，除了有一些简单的优化（如常量表达式折叠、条件编译），还有一些基础的脱糖（Java 8 之前的语法特性）操作会生成一些新的类 / 方法 / 指令，如匿名内部类会被编译成一个名为 `OuterClass$1.class` 的新类，以及命名为 `access$200` 之类的桥方法。如果改动涉及内部类、泛型，二次打包 $ 后面的数字编号可能和线上包出现乱序。
3. **代码优化器（ProGuard/R8）**：目前主要使用第三方开源工具 ProGuard（Google 推出 R8 计划取代 Proguard），通过 30+ 可选优化项，对前面生成的 Java 字节码进一步压缩、优化、混淆，可以使得 Android 安装包更小、更加安全、运行效率更高：
   - **压缩**：通过静态分析并删除未被使用的 class/field/method，即源码中存在的 class/field/method，线上包中不一定存在。
   - **优化**：通过一系列优化算法或者模版，对字节码进行优化，使得构建产物更小、运行更高效 / 安全，优化手段有合并类 / 接口、内联短方法、裁剪方法参数、删除不可达分支、外联代码（R8 新增）、删除无副作用代码（如 Log.d()）、修改方法 / 变量可见性等等。优化后的字节码相比源码，可能出现 class/field/method 数量减少、field/method 访问修饰符发生变化、method 签名发生变化、code 指令变少，另外二次构建优化结果可能和线上包不一致。
   - **混淆**：通过将 class/field/method 的名称重命名为一个无意义的短字符，增加逆向难度，减少包大小。二次打包需要保证和线上包的混淆保持一致，不然补丁加载后因调用异常而发生崩溃。
4. **脱糖工具**（图中未标出，旧版本使用三方插件 Lambda/Desugar，新版本中使用自带的 R8）：由于低版本 Android 设备不支持 Java 8+ 语法特性，这一步需要将 Lambda 表达式、方法引用、默认和静态接口方法等高版本的语法特性转为低版本实现。其中 Lambda 表达式会被编译成一个内部类，会有类似（2）中的问题。

至此，我们对本章开头提到的 2 个问题的产生原因有了一定认识，经过 Android 构建过程生成的字节码相比源码在 class/field/method/code 维度上有了“结构性”的变化，比如修复代码中调用的 class/field/method 在线上包中不存在（被 shrink、被 merge、被 inline），或者源码中可以访问、但在补丁中无法访问的 field/method（修饰符被标记为 private）、method parameter 列表匹配不上（之前没有被用到的 parameter 被裁剪了）等等。

Proguard 提供的这些优化项是可选的，一般情况下大型 Android 项目中会结合实际收益、稳定性以及构建耗时等多方因素综合考量后，会禁用一部分优化项，但并不是完全禁用。因此，二次打包时和线上包会产生一些差异，补丁制作准确性会受此影响。过去 Robust 补丁制作过程经常遇到此类问题，通过特殊字符检测、白名单等方式能够提升识别的准确性，但实现方案不够自动化。Robust 补丁制作流程如下：

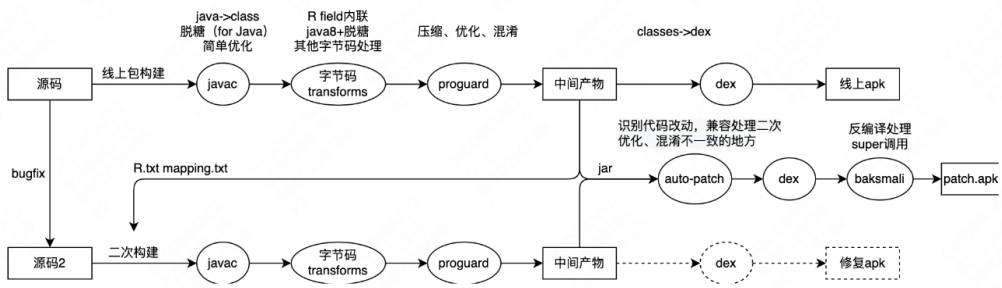


图 2　Robust 补丁制作流程

如果将 Android 项目的构建工具链（Android Gradle Plugin）升级到官方较新版本，上图中的 Proguard（Java 字节码优化和）+ Dex（Android 字节码生成）两个环节将被合并成一个，并被替换成 R8：

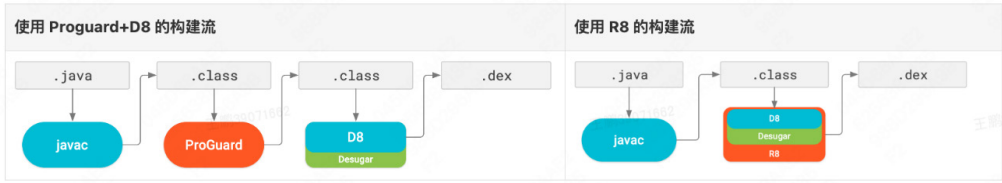


图 3　两种构建流

上述构建工具链的升级变化，给 Robust 补丁制作带来 2 个新的问题：

1. 没有合适时机制作补丁。如果将基于 JAR 的改动识别方案，改成基于 DEX 或者 Smali，等同于更换补丁制作方案，前者需要基于 DEX 文件格式和指令，后者需要处理大量寄存器，更容易出错，兼容性和稳定性不够好。
2. Proguard 可以禁用一部分优化选项，但是 R8 官方文档明确表示不支持禁用部分优化，相比之前会产生更多的差异，对改动识别造成干扰。

三、解决思路

3.1 整体方案介绍

基于 R8 构建的补丁制作思路是将改动识别提到优化混淆之前，对比 Java 字节码，同时结合对线上 APK 结构化解析（class/field/method），校正补丁代码对线上代码的调用，得到 patch.jar，最后借助 R8 对 patch.jar 进行混淆（applymapping）、脱糖、生成 Dex，打包得到 patch.apk，完整流程如下图所示：

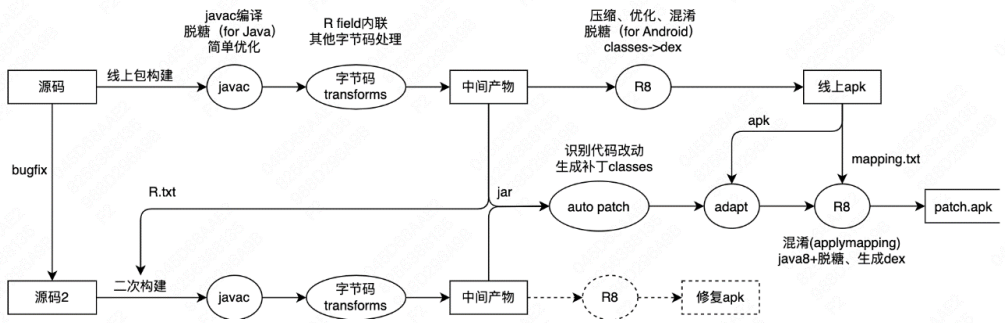


图 4　完整流程

3.2 问题和解决方法

3.2.1 R8 与 Proguard 优化对比

部分 ProGuard 的配置项在切换到 R8 后失效，R8 官方文档对此做出的解释是：随着 R8 的不断改进，维护标准的优化行为有助于 Android Studio 团队轻松排查并解决您可能遇到的任何问题。

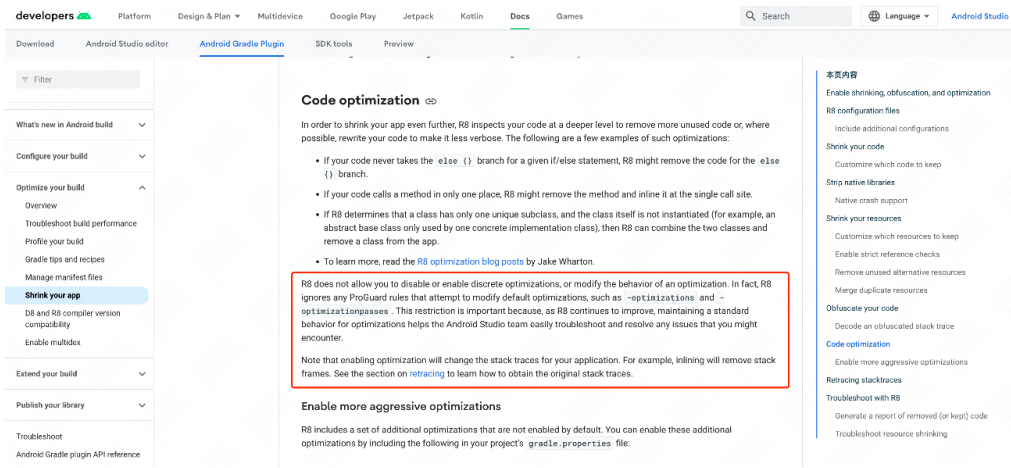


图 5　R8 官方解释

截至目前，仍能在网上搜到不少因 R8 优化带来的问题，没有公开文档介绍优化规则的使用和禁用说明。只能通过阅读 ProGuard 官方文档和 R8 源码，对比分析两者优化规则的相似和差异。通过 R8 源码发现可以通过隐藏的构建参数、反射或者直接修改 R8 源码实现一部分规则禁用，虽然 R8 的优化规则并不是和 Proguard 一一对应，但也基本可以实现和之前使用 Proguard 时相同的优化效果。

```
com.android.tools.r8.utils.InternalOptions.enableEnumUnboxing
com.android.tools.r8.utils.InternalOptions.enableVerticalClassMerging
com.android.tools.r8.utils.InternalOptions.enableClassInlining
com.android.tools.r8.utils.InternalOptions.inlinerOptions().
enableInlining// 方法内联
com.android.tools.r8.utils.InternalOptions.outline.enabled)// 方法外联
com.android.tools.r8.utils.InternalOptions.testing.
disableMarkingMethodsFinal
com.android.tools.r8.utils.InternalOptions.testing.
disableMarkingClassesFinal
```

一些规则可以通过构建参数 -Dcom.android.tools.r8.disableMarkingMethodsFinal 来控制关闭 / 开启，其他不支持的参数也可以参考如下方式简单改造一下：

```
public boolean disableMarkingMethodsFinal =
    System.getProperty("com.android.tools.r8.disableMarkingMethodsFinal") != null;
2 usages
public boolean disableMarkingClassesFinal =
    System.getProperty("com.android.tools.r8.disableMarkingClassesFinal") != null;
```

图 6　改造方式

如果某个项目中不希望禁用这些规则呢？在之前的补丁制作流程中，可能会影响改动识别的准确性。而在新的补丁制作流程中，改动识别不受影响，但在识别之后，还需要结合线上 APK 检查补丁中的外部调用是否合法。进一步仔细分析这些优化规则，可以分为 class、field、method、code 四类，其中对 Robust 补丁制作影响较大的是方法内联、参数移除、被标记为 private，后面的小节里将会介绍相应的处理方法。

3.2.2 “真”“假”改动识别

如果源码中有匿名内部类，javac 会编译生成一个命名为 { 外部类名 }${ 数字编号 } 的类，后面的数字编号是根据该匿名内部类在外部类中出现的先后顺序，依次累加计算出来的。

线上包	二次包	diff
MainActivity$1 { "aaa"}	MainActivity$1 { "aaa"}	完全一样
MainActivity$2 { "bbb"}	MainActivity$2 { "ddd"}	新增了MainActivity$2
MainActivity$3 { "ccc"}	MainActivity$3 { "bbb"}	实际没有变，仅类/方法的编号变化
	MainActivity$4 { "ccc"}	实际没有变，仅类/方法的编号变化

当修复代码中有新增 / 删除匿名内部类时，仅通过类名无法比较（所以在一些以类为最小粒度的热修复框架使用文档里，会看到类似“不支持新增匿名内部类”、“只支持在外部类的末尾增加匿名内部类”之类的描述），这时候 Robust 会模糊处理后面的数字编号，通过字节码对比进一步查找到真实变化的匿名内部类，识别出哪些是真改，哪些是假改。

此外，如果嵌套类之间涉及私有 field/method 访问，javac 编译器会生成命名规则为 `access$100`、`access$200` 的桥接方法，`access$` 后面的数字编号（和出现的先后顺序有关）也会影响改动识别（最终 R8 会将修饰符改成 public 并删除桥接方法），这里的解决办法和上面识别真实内部类改动的方式类似。

还有一种情况值得注意，大一点的 Android 项目通常会采用组件化的方式，每个组件以 AAR 形式参与 App 构建打包，在组件二进制发版（源码 -> AAR）过程中，可以使用 R8 进行脱糖（For Android）得到 Java 7 字节码，典型的例子是 Lambda 表达式，经过脱糖处理生成 { 外部类 }$$ExternalSyntheticLambda{ 数字 }（甚至有多重数字的情况如 $2$1）之类的 class，以及在外部类中生成命名规则为 lambda${ 方法名 }${ 数字 } 的静态方法（不同的脱糖器，命名规则不一样），补丁生成工具处理方法

和上面类似。

最终识别出来的代码改动，不仅包含源码有改动的方法或者新增方法 / 类（如果有），还包括与之有关的、由 javac 编译器脱糖生成的字节码，以及由组件二进制发版过程中经 R8 脱糖生成的字节码。

3.2.3 内联识别与处理

通过第二章节的介绍，可以看到线上代码在经 javac 编译之后还会经过字节码优化、混淆等处理，因此，通过上面字节码比对识别出来的代码变更（class/method 维度），如果涉及对线上代码的调用，还需要确保这些 Field/Method 的调用是“合法”的，避免运行时崩溃。

在众多优化项当中，主要需要关注的是 class/field/method 是否存在、是否可访问。如果线上包中不存在（上次构建过程中被移除或者被内联），补丁生成阶段需要当做新增类 / 方法加进来；如果线上包中不可以被外部访问（上次构建过程中 public 被改为 private），补丁生成阶段需要将直接调用改成反射调用；如果线上包中方法签名发生变化（上次构建过程中参数被裁剪），需要修改调用或者当做新方法加进来。

由于 Dex 文件与标准的 class 文件在结构设计上有着本质的区别（Dex 工具将所有的 class 文件整合到一个或几个 Dex 文件，目的是其中各个类能够共享数据，使得文件结构更加经凑），两者无法直接对比。具体检测方法是先通过 ASM 分析补丁 class 中的外部引用，然后借助 dexlib2 库解析 APK 中的 Dex，提取出 class/field/method 结构化信息（还需反混淆处理），最后再兼容性分析和处理。

R8 外联优化是一种高级优化技术，生效条件非常苛刻，需要在合适的环境下合理使用，R8 的外联优化会将多个方法中的相同代码提取到新方法中，以降低代码体积，但是会增加一次方法调用开销。如果恰好想修复的代码是被外联出去的方法，直接将外联方法当成新增方法来修复即可。

3.2.4 混淆问题与优化

不同于前面对在二次打包过程中对整个项目进行 ApplyMapping，这里只需要对少数发生变更的类进行 ApplyMapping，出现混淆不一致的概率会小很多。Robust 补丁制作过程中，仅将改动的类传递给 Proguard 进行二次混淆，这个过程中会自动应用线上

包的 mapping 文件：

```
-applymapping {线上包的 mapping.txt}
```

但在某些特殊情形下，比如删了一个旧方法、同时又增加了一个新方法，或者是 ApplyMapping 的缺陷，还是会出现补丁中的混淆和线上混淆实际并不一致的情况，因此在生成补丁之后，还需要根据线上 APK 进行对比校验，如果发现错误混淆，进一步反编译成 Smali 之后进行字符替换。

3.2.5 其他方面的优化

(1) super 指令

在 Android 开发中，invoke-super 指令经常被用来重写某个系统方法，同时保留父类方法中的一些逻辑。以 Activity 类的 onCreate 方法为例：

```
public class MyActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); // 调用父类的 onCreate 方法
    }
}
```

其中 super.onCreate(savedInstanceState) 就是一种典型的 super 调用，经过 Dex 编译后，在 Smali 语法层面看到的就是 invoke-super 指令。但在 patch 类中，无法编写类似 myActivity.super.onCreate(savedInstanceState)，因为 super 只能在原类使用；即使采用字节码技术强行编写了，在运行时会提示 java.lang.NoSuchMethodError，因为 patch 不是目标方法的子类。

为了模拟实现 JVM 的 invoke-super 指令，需要为每个 patch 类生成一个继承了被修复类父类的辅助类（解决 super 调用只能在目标子类使用的问题），并且在辅助类里面将 patch.onCreate 转换为原始类的调用 origin.super.onCreate。Robust 早期是在 Smali 层面进行处理的，需要将 Dex 转换为 Smali，处理完以后，再把 Smali 转换为 Dex。用 ASM 字节码直接对 Class 字节码进行处理更方便，不需要再转换为 Smali，针对该辅助类的 ASM 字节码转换关键代码如下：

```
public class SuperMethodVisitor extends MethodVisitor {
    ...
    @Override
```

```
    public void visitMethodInsn(int opcode, String owner, String name,
String desc, boolean itf) {
        if (opcode == Opcodes.INVOKEVIRTUAL) {
            // 将 INVOKEVIRTUAL 指令替换成 INVOKESPECIAL
            super.visitMethodInsn(Opcodes.INVOKESPECIAL, owner, name, desc, itf);
        } else {
            super.visitMethodInsn(opcode, owner, name, desc, itf);
        }
    }

    @Override
    public void visitVarInsn(int opcode, int var) {
        if (opcode == Opcodes.ALOAD && var == 0) {
            // 保证 super 调用在原始类
            mv.visitVarInsn(opcode, 1);
            return;
        }
        mv.visitVarInsn(opcode, var);
    }
    ...
}
```

上述方式是采用了一个辅助类来实现的，下面介绍另一种改进的方法。

在 JNI 层，常见的 CallObjectMethod 函数适用于调用虚方法，即调用方法时依赖于对象的类层次结构，类似于 Java 的 invoke-virtual；与之对应的是 CallNonvirtual-ObjectMethod 函数，它适用于非虚方法调用，即调用的对象为指定的类的对象，无论这个类有没有被继承或覆盖，也就是说可以通过 CallNonvirtualObjectMethod 调用父类 super 方法。

Java 语言中的 invoke-super 指令可以通过 CallNonvirtualObjectMethod、Get-MethodID 组合来实现，关键代码如下：

```
jmethodID methodID = env->GetMethodID(parentClass, "superMethodName", "()V");
jvalue args[] = {};
jobject result = env->CallNonvirtualObjectMethod(parentObj, parentClass,
methodID, args);
```

(2) <init> 函数的插桩与修复

在部分子类 <init> 函数会显式调用父类的构造函数 super()，且 super() 必须是子类 <init> 函数中的第一句语句，否则编译失败。因此对于 <init> 函数，不能在第一行进行 Robust 插桩，需要在父类的构造函数 super() 之后插桩。

那么 <init> 函数如何修复呢？原始类 <init> 函数修改后，在 patch 类也是 <init> 函数，这里需要将该 <init> 函数拷贝成普通函数，并将原始类的 Robust 插桩关联到该普通函数。

复制构造函数并将其转换为方法需要注意：

- 原始类函数名称 <init> 需要改成普通方法名称，避免与 patch 类的 <init> 函数冲突。
- 原始类 <init> 函数如果有方法参数，则需要保留成一致的。
- patch 类新方法的 return type 是 void。
- 原始类 <init> 函数如果有调用 this() 或 super() 构造函数，则需要在 patch 新方法里删除它们。

（3）<clinit> 函数的插桩与修复

<clinit> 函数是由编译器生成的一个特殊的静态构造方法，它被用来初始化类中的静态变量和复杂的静态表达式。如果在一个类中定义了静态变量或代码块，那么编译器会为这些静态变量和代码块生成一个 <clinit> 函数。<clinit> 函数只会被执行一次，虚拟机会保证只有一个线程能够执行 <clinit> 方法，确保对共享的类级别变量的线程安全访问。

因此，对 <clinit> 函数进行插桩和修复时，需要特别注意 <clinit> 方法的执行时机：

- 在类实例化时，如果该类的 <clinit> 方法还没有执行，则会执行该方法，以初始化类的静态变量和复杂的静态表达式。
- 在通过反射获取该类的某个静态成员时，如果该类的 <clinit> 方法还没有执行，则会执行该方法，以初始化类的静态变量和复杂的静态表达式。
- 如果该类被子类继承，而子类中也定义了 <clinit> 方法，则在创建子类实例时，会先执行父类的 <clinit> 方法，然后再执行子类的 <clinit> 方法。

根据上述 <clinit> 函数执行时机分析，插桩时不能访问类的静态成员变量（访问静态变量时 clinit 函数就已经执行了，无法被有效修复），因此无法借助于 Robust 常规插桩方法（给 Class 插入一个静态接口 Field），需要借助一个辅助类 ClintPatchProxy 来实现插桩逻辑。

```
/**
 * 线上 MainActiviy clinit 插桩
 */
public class MainActivity {
    static {
        String classLongName = "com.app.MainActivity";
        if (ClintPatchProxy.isSupport(classLongName)) {
            ClintPatchProxy.accessDispatch(classLongName);
        } else {
            // MainActitiy Clinit origin code
        }
```

clinit 函数修复时，在补丁入口类的静态代码块里面设置好 ClintPatchProxy 的跳转接口实现即可，原 MainActivity 的 clinit 代码将不再执行，转而执行 MainActivityPatch 的 clinit 代码（对应 MainActivity 的新 clinit 代码）。

（4）修复新增类 / 新增成员变量 / 新增方法

基于方法插桩的方法，天然支持新增类；对于新增 Field 和新增 Method，分两种情况：静态的 Field 和 Method 可以用一个新增类来包裹；新增非静态 Field 可以使用一个辅助类来维持 this 对象与该 Field 的映射关系，补丁里面原本使用 `this.newFieldName` 的代码，通过字节码工具转换为 `FieldHelper.get(this).getNewFieldName()` 即可。

4. 总结

回顾 Robust 热修复补丁制作过程，主要是对构建编译过程和字节码编辑技术的巧妙结合。通过分析 Android 应用打包过程、Java 语言编译和优化过程，补丁制作过程中可能会遇到的各种问题就有了答案，再借助字节码工具分析、处理就能够生成一个热修复补丁。当然，这其中涉及大量的细节处理，仅通过一篇文章不足以涵盖各种细节，还需要结合实际项目才能有更全面的了解。

本文作者

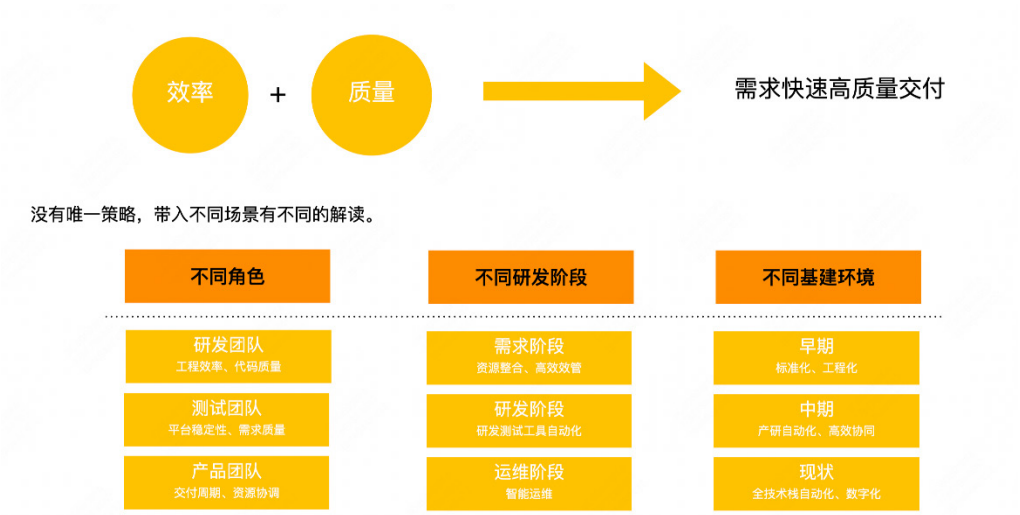
常强，美团平台 –App 技术部工程师。

美团到店终端从标准化到数字化的演进之路

作者：彬彬 自强

1. 背景

在我们日常工作中，要实现需求的快速和高质量交付，关键在于提升效率和质量。然而，在实践中很难找到一个通用策略，能同时针对效率和质量进行优化。这是因为在不同的场景下，对效率和质量会产生不同的解读。例如，在需求研发阶段，研发团队可能更注重提升工程效率和代码质量；而测试团队可能会更关注平台的稳定性和需求的质量；产品团队则可能聚焦于缩短交付周期和提高资源的有效使用。



在深入探讨具体的研发场景时，我们遇到了若干挑战。首先，一个完整的交付流程涵盖了从需求开发到产品上线的各个阶段，涉及产品、研发、测试等关键角色，每个人的职责各有侧重，且整个过程依赖于持续的沟通与协作。在这个环节中，我们常会遇到这样的问题：尽管每个人都忙碌，但整体效率却并没有明显的提升；即便对单一环节进行了优化，但优化效果并没有直接让整个需求的时间缩短，有时甚至会导致资源等待时间变长。

对于第一个问题，这通常不是个人能力的问题，而是因为我们在产研协同流程或资源管

理工具上还有改进的空间。而第二个问题，可以举个例子这样理解：假设一个研发任务预计需要 3 天完成，一个高效的研发人员可能在 1 天内就完成了开发工作。然而，在实际的工作流程中，这样的个人优化并不会直接把 5 天的需求变为 3 天。因为开发后还需要进行多个步骤，如前 / 后端联调、需求验收、测试用例编写等，因此个人的单点优化很难对整个项目产生显著的影响。

因此，面向全流程优化的关键在于采取一个全面的视角，关注整个团队和协作流程的改进。只有当我们从整体出发，全面优化各个团队的工作流程和协同机制，我们才能真正实现缩短交付周期、提高效率和质量的目标。



产研阶段不同应对的问题不同

在进行分阶段优化时，各个企业在需求、研发和运维这些关键阶段通常都已经拥有一些成熟的解决方案或产品。例如，在需求阶段，企业通常会利用需求管理和排期管理工具；而在研发阶段，则广泛采用各种成熟的 CI/CD 工具（这个阶段进一步细分为多个子阶段，每个子阶段都有其特定的目标和潜在的提升点）。为了有效地进行优化，首先需要建立一个清晰的流程规范，明确各个阶段的具体任务和目标。随后，将这些明确的规范转化为线上化的操作，通过人力资源的有效配置，以自动化的方式实现效率和质量的双重提升。



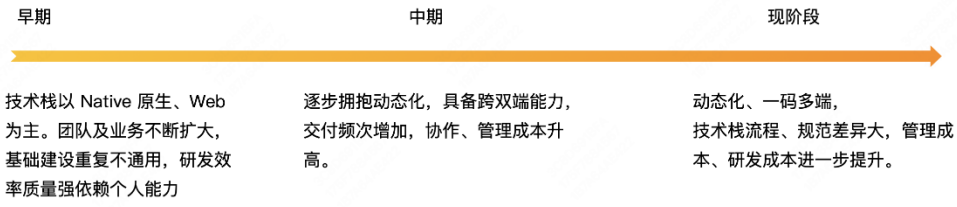
不同业务形态和基础环境面对的问题不同

总结起来，优化方向主要有两个：首先是全面提升产研协同的效率，其次是通过建立标准化、线上化，进而实现自动化的过程进行优化。然而，根据不同公司和业务形态的具体情况，这两方面的具体感受和需求也会有所不同。

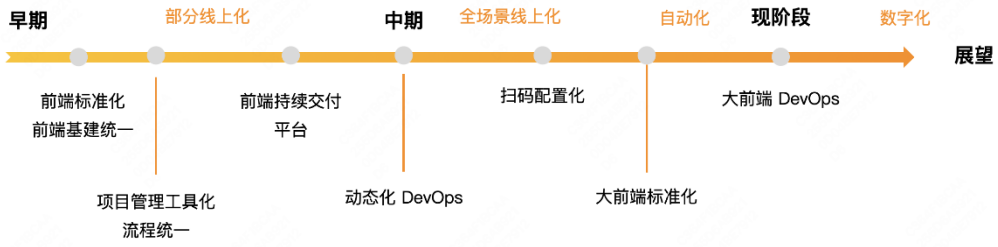
以美团的业务技术形态为例，早期时，技术栈主要是 Native 和 Web 独立运营，各个团队“各自为政”，效率和质量的提升很大程度上取决于团队的规范执行情况和个人的编码习惯。因此，这一阶段最主要的痛点在于缺乏统一的标准和最佳实践。

进入中期，开始逐渐采用动态化技术，一些需求也开始具备跨 iOS 和 Android 的能力。发版模式从原先的 Native“火车发版”转变为动态发版，产研交付的频次大幅增加，从两周一版变成了一周甚至一天一版。这一变化极大地考验了团队的协同能力和快速发版的能力。因此，这个阶段的挑战在于如何通过工具化的方式有效整合资源，以及如何适应快速变化和发展的节奏。

目前阶段，美团的业务已全面拥抱动态化，并在多个场景下实现了一码多端的能力。从之前的 iOS、Android 到现在的多端对齐，研发的差异性和管理成本、质量风险都有所增加。因此，当前阶段的核心问题是如何通过一套统一且通用的自动化方案，来实现整体效率的提升和质量的保障。



演进策略概览



2. 标准化

什么是标准化？为什么要做标准化？为什么大家都是共同实践出来的经验，A 说的就是标准，B 说的就不标准？带着这些问题我们这里先展开标准化的建设过程。

标准化背景

如图中所示，前端标准化实施之前，从业务规范到技术选型，差异显著，且难以判断各种方案的优劣，每个团队均根据自身的经验实施各异的实践。



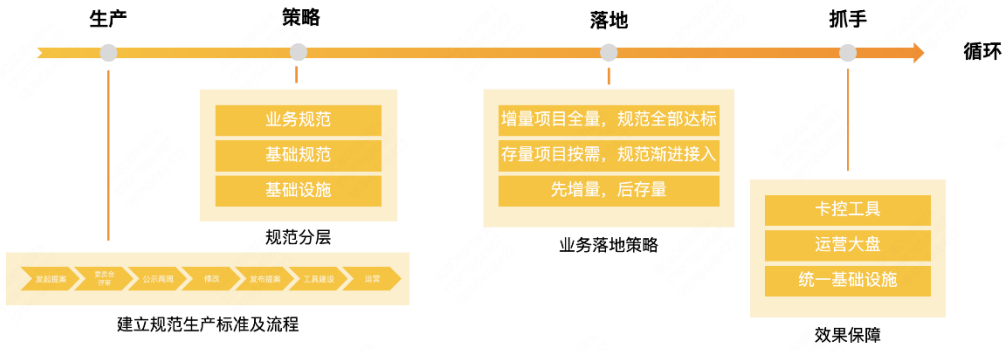
在此背景下，我们面临了所谓的“三高”问题：

- **操作成本高**：缺乏统一流程标准，相同的工作流在不同团队中执行方式各异。例如，在处理故障时，快速实现跨团队的止损方式和优先级判定共识极为困难。
- **学习成本高**：由于工程结构、技术选型和周边能力的差异，新成员或跨团队协作时的学习成本高，适配难度大。
- **维护成本高**：工具的重复建设，同一环节可能存在多套保障工具，质量参差不齐，且问题碎片化严重，导致维护成本激增。

面对这些挑战，我们亟需制定一套全面的标准化落地流程。

标准化生产过程

首先，从制定标准的生产过程入手，建立规范的生产标准和讨论流程，明确谁将负责制定和决策标准。接着，通过实施分层次的规范，从基础设施和基本规范到业务层规范，逐步实施。最终，当规范在业务侧得以实施后，应通过监控工具和运营大盘指标来确保这些标准得到有效执行。



基础设施支撑规范落地

基础设施、研发规范和研发流程三方面的标准化统一。



研发规范落地

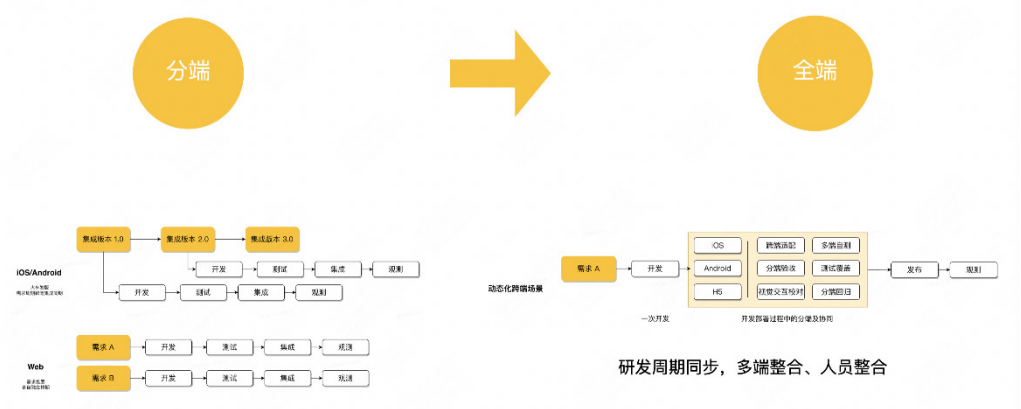
业务规范和技术选型的统一。



3. 产研协同

协同成本增加

在动态化改造的背景下，我们从之前各技术栈独立运维转向了合并至一个统一的开发和发布流程。这一转变带来了人员和团队角色的显著变化。随着多端技术的整合，原先专注于单一平台的研发和测试人员现在需要同时关注多个平台，人和团队的职责都被放大，同时复杂度大幅增加了。



终端场景能力增加

随着终端的不断建设，能力逐渐丰富，场景也越来越复杂。这些终端能力在不同 App、iOS、Android 上的实现也都各异。高门槛及一系列复杂操作让终端研发和测试头痛不已。

举几个研发中常遇到的情况：1. 前端问后端的接口怎么出问题了，是不是你们又变更了部署？后端又反问前端，你切环境了吗，泳道是不是正确？ 2. 测试问研发这个 Bug 怎么没有修？研发开始反确认，容器锁包锁了没有，环境对不对，AB 策略切了没有，灰度链路是否正确？ 3. 团队来了新同学，这些问题差异又就会被无限放大，在某 APP/某端上，到底怎么切灰度链路，怎么设置环境，怎么锁包这一系列的问题。

这些问题十分阻碍整个需求的研发，但却频繁的在日常研发中遇到。

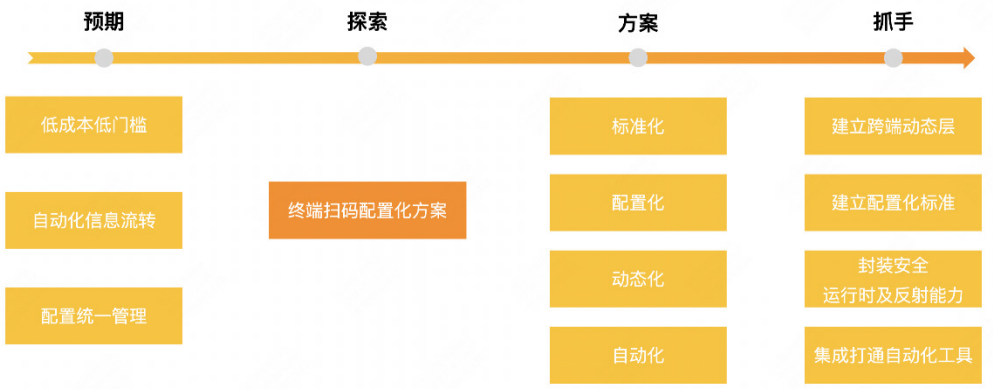


如何基于终端特点解决这些问题?

好的功能往往需要用最简单的方式呈现，比如移动支付仅是通过“扫一扫”的方式（几乎没有操作成本，而且零学习成本）解决复杂场景，“扫一扫”的背后可能会解决像入网支付、支付安全性等一系列复杂问题，但呈现给用户是简单的“扫一扫”操作。

因为终端本身和服务端、Web 不一样，它有一些原生的能力，那我们是否也可以把这种复杂的方式，用最简单的方式呈现?

用简单无门槛的操作解决全流程痛点



终端扫码配置方案挑战点

在方案设计过程中，我们面临了几项技术挑战。首先，推动基础设施的统一是一项艰巨

任务。鉴于公司内有众多基础设施团队，每个团队负责独立的应用、平台或功能，要统一这些分散的基建到一个共同标准，并推动大规模的改造，既困难且成本高昂。其次，即使推动了基础设施能力的标准化，由于终端技术不断进化且新功能持续增加，如果依赖单一团队来维护所有扩展，不仅成本高昂，还容易导致团队过度劳累。

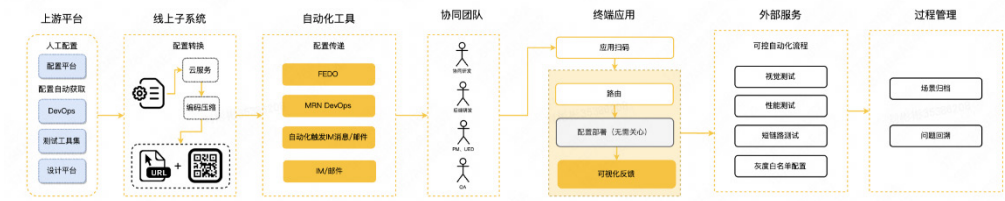
针对这些技术挑战，我们采用了结合原生特性的方法。例如，引入了动态化能力来解决跨端一致性问题，使用原生技术解除了双向依赖，并通过按需加载机制来避免对业务逻辑的影响。最终，实现了两大插件化能力：一是框架插件化，允许无依赖地集成至各个应用中，提供运行时检查和挂载能力。二是功能插件化，允许业务团队在自己的业务组件和仓库中，按照标准接口实现和快速低成本地新增功能插件。



工具化解决流程管理难点

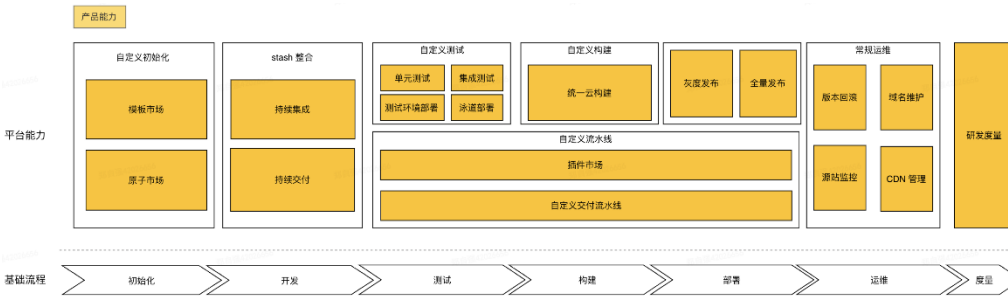
最终，我们落地了这个低成本、高易用性的方案，即一码全流程方案，有效解决了过去依赖文档或大量口头沟通才能实现的产研协同问题。现在，通过一个二维码就可以串联整个终端研发流程，无需反复确认操作步骤，也不用关心具体哪个端、哪个 App 需要怎样操作，所有复杂的操作都隐含在“扫一扫”的背后。

同时，由于配置的自动化和可控性，方案还顺利整合了美团的自动化测试工具体系，促进了自动化的精准控制。结合正在建设的 DevOps 平台，我们实现了从产研信息的自动化采集到二维码的自动生成；此外，还可以通过二维码反解出当时各个验收环境的具



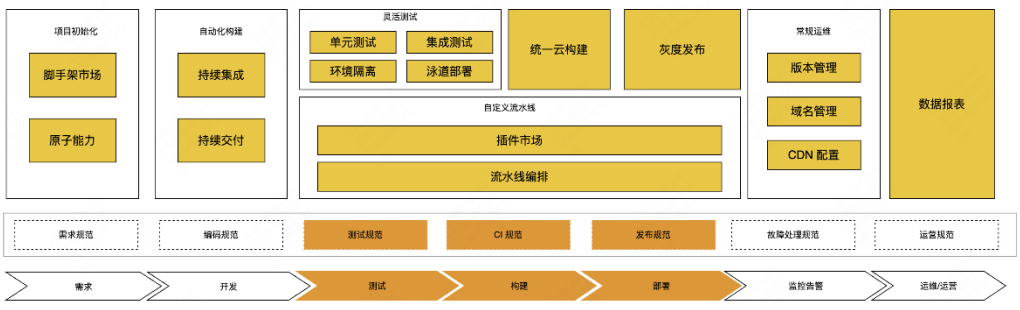
4. 持续交付体系

在流程和规范之上是团队基础设施，有句老话是“基础设施承载团队的流程规范”，所以不同阶段、不同规范需要依托不同基础设施来落地。持续交付基础设施作为我们需求交付过程中的核心环节或者硬卡控平台（硬卡控是我们在发布过程中，如果某一个环节不符合规范，那就作为卡控形式拦截线上风险），持续交付平台也成为整个流程和规范落地过程中的核心抓手。接下来，会主要分享持续交付基础设施从线上化到数字化的演进过程。

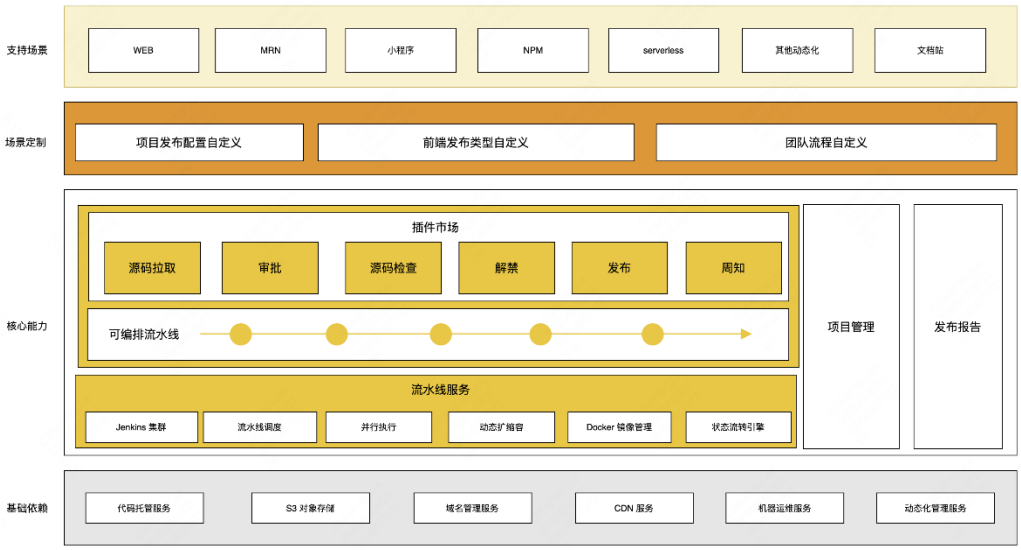


早期：发布流程线上化

前文提到，在早期阶段，团队完成了对前端发布流程的标准化，但团队前端项目数量不断扩张，多样性的工程化与规范定制诉求导致了我们脚手架的碎片化程度高、重复工程化建设、上线质量不可控、约束流程不健全等问题。所以我们在这个阶段做了线上化、可定制的交付流程解决这个问题，也进一步成为了团队前端的持续交付平台。



如下图可以看到，除了底层的基础依赖，核心支撑团队规范的是一个可编排流水线，在它基础上提供了不同场景下的卡控能力，比如源码拉取、审批、代码检查、解禁、发布、周知等，承载了团队标准化所定义的一些基础规范。此外，加上项目管理和发布报告，组成了团队早期持续交付的基础能力。同时可以对团队不同场景提供能力定制，比如有不同类型项目的发布配置、在前端有不同发布类型自定义、团队流程的自定义。在这些能力之上，支撑了早期团队不同类型的交付场景，比如 Web、动态化、小程序、NPM 等。



用团队实际场景来举例，从团队标准流水线来看，我们将团队的标准化规范内置到默认流水线中，例如在默认流水线里提供了 Lint 规范检查、单测检查、工程依赖收集、标准化检查等。

在这个基础上，团队可以定义扩展自己的流程规范，比如在线上发布的过程提供域名安全检查 / 是否包含不合规域名、ES 高级语法检查以及上线后的数据监控，既满足了标准规范的卡控，又满足了团队能力定制扩展的诉求。



除了存量项目通过标准规范进行流程卡控，对于增量项目我们也期望同样保持标准不劣化，所以在持续交付的解决方案里，我们也内置了一键初始化基础能力，做到项目创建即标准，除了 10 分钟完成项目从创建到发布外，其本身也默认符合团队的标准化规范。

10 分钟完成标准项目从创建到发布



项目创建



工作台消息推送



项目发布

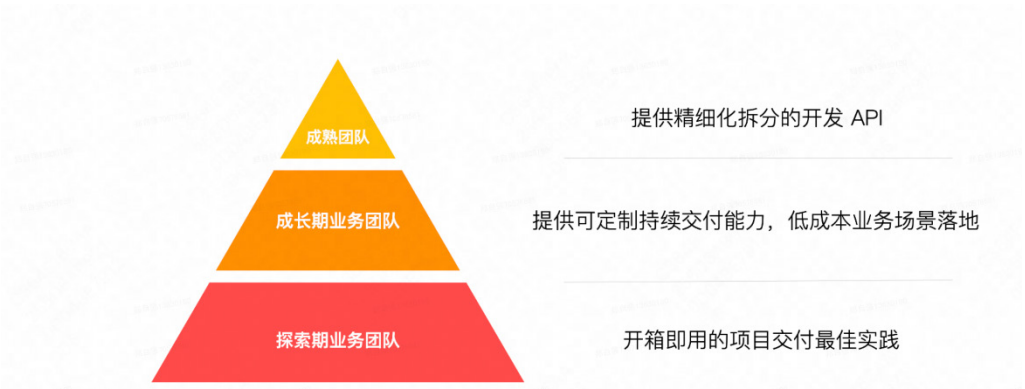
为了把这套能力更好的提供给业务团队，平台在基础能力之上提供了不同场景差异性的解决方案，为不同场景、不同阶段的团队对于规范的诉求差异性提供不同的基础能力。

探索期团队的诉求是想快速验证业务可行性，团队内的规范定义相对比较少，我们会提

供开箱即用的最佳实践给团队，使用默认的这套规范快速交付。

成长期的团队逐渐有了对于业务场景的定义或特定场景下的规范，我们会提供可定制的持续交付能力，例如“期望在特定阶段完成自己业务项的检查”或者“要求团队所有发布需 TL 审批”等这种定制性的规范，可以低成本扩展在持续交付过程中。

成熟的团队一般有自己完整的标准化规范，我们就需要将持续交付基础能力拆解为相对细粒度的 API，给成熟团队拼装自己完整的定制化交付流程。



第一个阶段，我们将标准化结合持续交付平台完成发布流程的线上化落地，通过灵活的定制化能力支撑不同业务场景，累计支撑了美团上千个前端团队的项目发布流程升级。

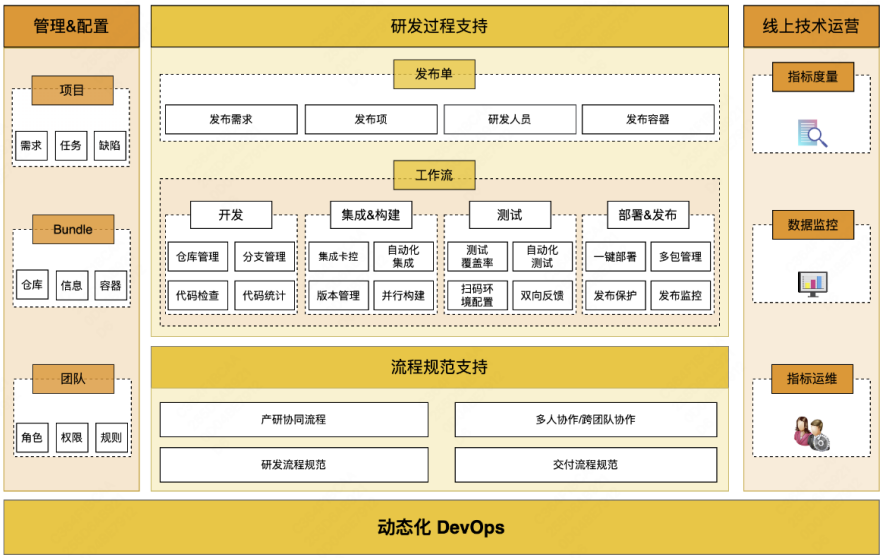
中期：动态化交付复杂度增加

动态化场景的引入是一把双刃剑，其本身会提升业务交付价值效率，但也也增加了研发流程复杂度。无论在开发、测试还是发布阶段，我们从原来的单一关注 App 以及 App 测试和发布过程，变成了需要同时关注 App 和多个动态化 Bundle 的交付全流程，增加了研发交付的复杂度。比如原来开发只需关注某个 App 单仓库单分支流程，但如果在动态化场景，除了要关注 App，也需要关注 Bundle 可能分散在不同的仓库，研发的流程可能就是多仓库多分支、对应不同 Bundle 的测试线上灰度流程和以及线上指标。



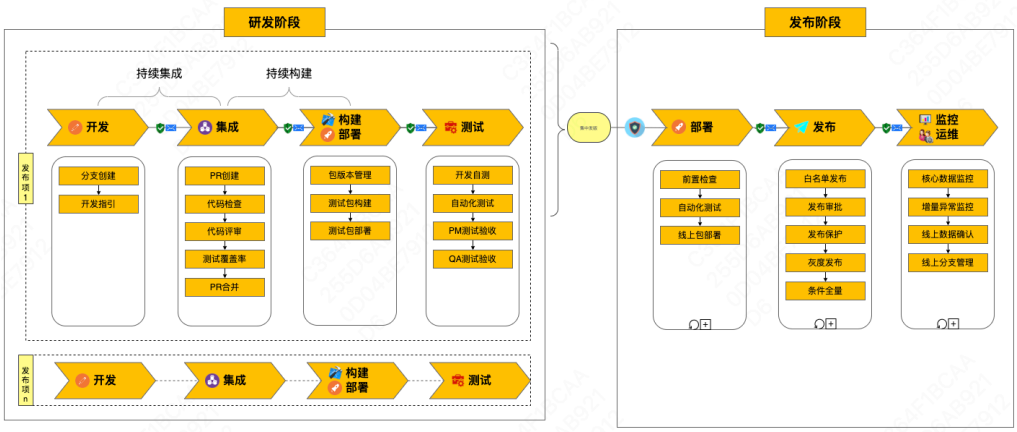
面对这种复杂度增加的情况，如何支撑团队定义的产研流程 / 流程规范的落地？我们的解决方案是动态化 DevOps，即将全流程通过线上托管的形式降低研发同学在交付过程中的复杂度。

如图是在动态化 DevOps 所提供的解决方案，除了两侧 DevOps 的常规能力即项目管理 & 配置和线上技术运营基础能力外，平台对团队流程规范内置支持，例如“产研协同流程”、“多人协作场景下的规范”、“研发流程规范”、“交付流程规范”等，平台的核心能力是工作流，通过平台定义的工作流将交付过程中的开发集成、构建测试和部署发布全流程托管起来，辅助研发提效。



还是以团队实际场景举例，平台工作流覆盖了研发和发布过程，同时定义了研发和发布的具体阶段。研发可以按流程或规范分为开发、集成、构建 / 部署、测试这 4 个阶段，在不同阶段，平台以工作流形式将标准动作或流程规范内置在工作流节点中，比如可以在平台上创建符合规范的分支，开发完成后，自动进行测试包构建、部署以及自动化测试和 PM、QA 的验收。研发过程中，除了将标准流程内置，也将产研协同的流程内置到这套工作流里，辅助规范落地的同时支撑产研协同效率。

同样，在发布阶段，我们将发布流程进行标准化定义并线上化，让工作流自动执行，辅助交付流程符合标准 / 研发协作规范。



总之，动态化 DevOps 完成了动态化场景全流程线上化，通过平台能力降低了交付的复杂度，同时我们将协同流程和研发规范进行平台内置，守住了质量底线。

大前端背景下的新问题

随着我们业务的演进，越来越多业务交付到多种前端场景，比如某些业务既涉及到 Web/H5 交付也需要动态化交付，甚至 C 端业务可能既有 H5 又有小程序同时又有动态化，业务形态决定了需要交付的场景，以交付流程来审视，它是多个维度的复杂问题。

除了持续交付领域流程，即从需求到发布再到度量，还有大前端不同交付端的维度，比如一个项目既要交付 H5，又要有小程序、动态化，同时需要 Native 的配合。还有第三个维度就是不同团队对交付的技术栈以及不同阶段的定义规范不一样。

虽然之前解决了部分场景下比如解决了 H5/ 小程序的发布流程、线上化的问题，还解决了动态化场景下的定制和线上化诉求，但是大前端、多团队视角下还是存在流程 / 规范落地的瓶颈。



那这些瓶颈都有哪些？我们总结了一下，主要包括以下三点：

研发周期涉及平台多，老人靠经验、新人靠询问。

各技术栈研发流程平台的差异性，导致团队学习成本和能力重复建设问题。

各团队基于标准规范扩展业务规范，规范落地分散，落地效果参差不齐。

那么在这种背景下，我们的解决方案是大前端 DevOps，提供给美团内部前端团队的研发工作台，分为四个能力：

第一，它涵盖了产研全流程，即从开始到需求、关联设计稿、自动创建分支，到最后上线、发布、周知等全流程所涉及到的能力；

第二，它支持团队定制，即允许在标准流程基础之上，各个团队有自己的场景定制；

第三，我们需要在这个流程中整合公司内部的基建，辅助在这个过程中因为平台多或者各技术栈场景不一样导致学习成本高问题；

第四，它一定要支持团队不同的大前端技术场景。



下图是我们对大前端 DevOps 平台的架构分层。



在底层，我们对于公司的前端基础设施进行了深度整合，并对大前端标准研发流程内置。

核心能力层通过工作流的编排以及定制化节点市场提供前端不同团队的场景定制，通过团队自研的工作流引擎支撑多样化的工作流执行。

工作流本身也是研发同学在日常工作中大绝大多数的工作场景，工作流可以内置团队研发规范，比如分支规范、卡控规范和部署规范，这些规范节点组成了工作流最小原子，不同技术栈、不同团队共享一个规范配置池，这样就解决了对于不同团队、技术栈对于同质化能力的重复建设问题。除此之外，工作流本身基于产研全流程，我们在全流程埋点提供精确的研发过程数字化能力，输出团队研发度量数据。

如图流程所示，通过标准工作流承载了的团队规范和产研流程，比如在流程里提供了标准分支创建、Lint 检查、产研协作动作、安全检查、灰度规范以及上线后的标准动作。



同时我们会在工作流内置研发前需求方案的一键创建、智能评估能力，研发中灰度异常指标自动采集、自动监控告警，研发后需求数据整理归档的一整套辅助能力，做到团队新人“开箱即用”。



在标准流程中可以增加团队的自定义规范，在节点市场选择的基础能力，无缝整合公司

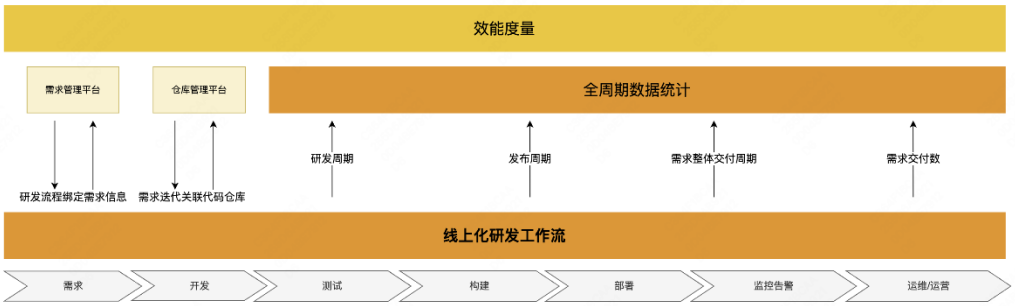
基础能力，例如可以在特定阶段创建分支、在某个阶段查看代码信息、在上线阶段触发某个端的侧 / 正式包构建，通过节点能力，整合基础设施，一站式闭环研发同学工作，避免这些研发过程来回跳转造成的效率损耗。



将不同场景、不同阶段的流程，定义成团队标准工作流模板，支撑不同场景下的标准交付动作，当然也可以针对一些新场景或者不同阶段定义自己的研发工作流。



大前端 DevOps 做到全流程线上化，所以可以在需求过程中关联需求信息、仓库信息，在不同阶段提供准确的研发数据，进行全周期数据统计，在全流程数字化基础之上，实现团队的效能度量，效能度量从宏观角度看到研发过程中所存在的共性问题，再反过来推动研发流程修正可能存在的问题，形成一个正向循环过程。



在这个阶段，通过大前端 DevOps 支撑了团队所有前端场景的研发交付，承载团队规范、产研协作流程，是研发开箱即用的一站式工作台。

团队的流程规范的演进从原来的不规范到研发规范标准化，再到产研协同过程的升级（即从产研协同配置化），再到产研流程自动化，为了支撑规范的落地，在持续交付基础设施上，需要完成交付线上化和全流程线上化，最终在线上化基础上，支撑全场景线上化、自动化和数字化能力。未来整合人工智能相关能力，完成流程规范上信息触达智能化、规范执行智能化，在这个过程中，团队流程规范和基础设施相辅相成、互相落地。



5. Q&A

Q1：多人、多分支的代码管理是否有好的经验可以分享？

A：分支管理有两个维度，一是质量相关，二是高频操作和效率相关。根据维度不同，有不同的推进策略。

首先质量相关方面，由于是多人协同，首先要解决的问题是大家在流程上很难达成一致，比如开发完成后，自己拉出一个分支，直接在 Master 分支上稳定分支上合，其他同学不知道 TA 正在 Release 分支上发布，这样的冲突很难通过线下方式主动发现，所以这种质量相关的东西，我们基本上会通过流程达成统一和增加卡控这两个方面做成，比如所有分支通过自动创建，人工不用关心应该从哪里检查出什么问题，它会直接告诉我们结果，即应该在哪里进行操作，一方面通过自动化方式解决流程不统一，一方面通过代码检查强卡控漏代码等场景。

其次效率相关方面，通过高频操作，分析哪些操作频繁，并可以提供像自动 TPR 或者智能推荐分析在哪个分支进行什么操作，智能推荐 GIT 命令、快捷命令等这方式。

Q2：团队代码审查有什么最佳实践吗？怎么平衡开发和审查任务之间的平衡？

A：这是我们持续交付过程中的标准环节，首先是定义规范，比如在这个环节有团队的共识规范，根据单个 PR 大小、Approve 的数量以及每个同学所提建议的数量定义规范，这个规范和不同团队 / 项目有关系，通过不同场景定义规范，再落到交付卡控过程中，比如我们会检查它是否达到阈值，如果没有达到，就不允许合并，或者检查不通过，没有办法进行下一步，以这种方式实现了代码审查规范的落地。

6. 本文作者

彬彬，自强，均来自部门美团到店事业群 / 平台技术部。

测试

如何提供一个可信的 AB 测试解决方案

作者：王鹏 永斌 中锋

1. 背景

虽然 AB 测试（AB 实验）的统计基础已经有一个世纪的历史了，但大规模地构建一个正确可靠的 A/B 测试平台仍然是一个巨大的挑战：不仅要在实验设计环节应对溢出效应和小样本的双重挑战，平衡好实验偏差与方差以确定合适的实验单元、分组方法和分析方法，给出合理的实验设计，而且要在分析环节应对方差计算、P 值计算、多重比较、混淆因素、假阴性（实际策略有效果，但是检测显示无效果）等多种统计陷阱。因此，要获得高质量的结果需要对实验和统计有专家级的理解，这无疑增加了实验门槛，难以达成任何人进行实验都可得出可信结论的目标。

本文将从实验方法和平台建设的两个视角，分别介绍如何正确地使用统计方法避免统计陷阱，以及输出什么样的平台能力，从而确保任何人使用该平台时都可得出可信结论。同时，我们也积累了如何进行更好的实验，以及如何利用实验来做出更好的决策，希望能给从事相关工作的同学有所帮助，也真诚地希望欢迎大家给出反馈或者建议，不断优化我们的工作。

2. 走进 AB 测试

哪个线上选项会更好？我们经常需要做出这样的选择。当我们想要在两个策略之间做出决定时，理想的方案是面向同一拨用户，在两个平行时空，平行时空 1 体验原策略 A，

平行时空 2 体验新策略 B，然后根据观测到的事实进行比较，以决定哪个策略胜出。然而在现实世界中，不存在两个平行时空，针对同一用户，我们只能观察到其接受策略 A 或策略 B 的一种效果，即反事实结果是观测不到的。

因此，在现实世界中，我们通常采用实验的方法做出决策。它将用户分配到不同的组，同一组内的用户在实验期间使用相同的策略，不同组的用户使用不同的策略。同时，日志系统根据实验系统为用户打标记，用于记录用户的行为，然后根据带有标记的日志计算度量差异，并进行统计分析以排除由于噪声导致的任何差异。实验者通过这些指标去理解和分析不同的策略对用户起了什么样的作用，是否符合实验预先假设。

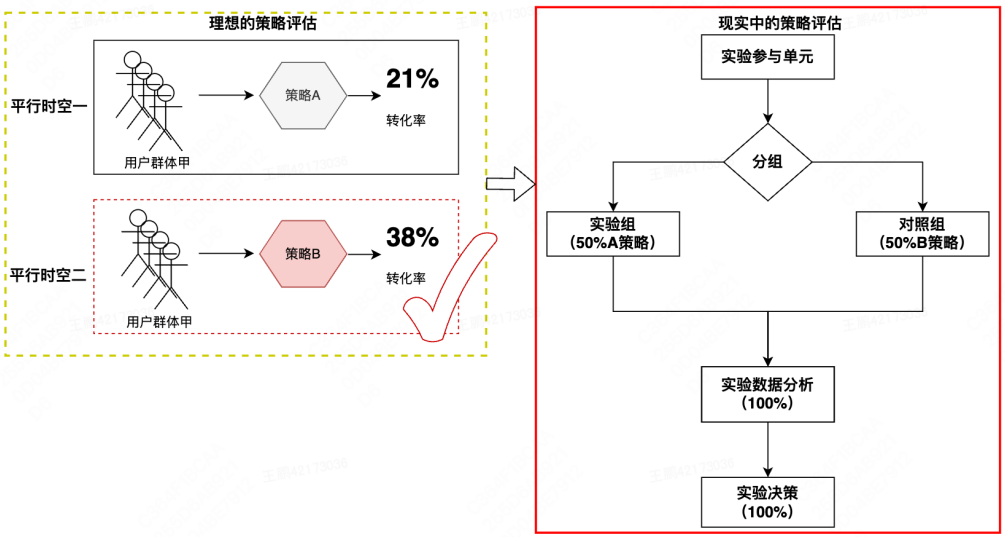


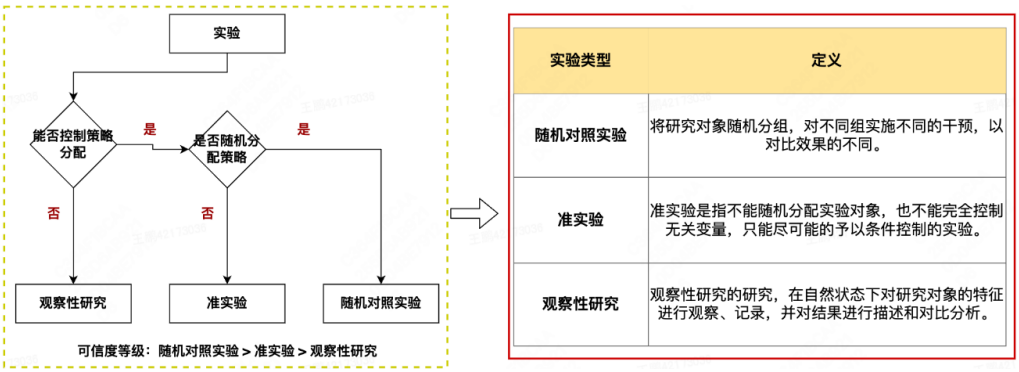
图 1　理想和现实中的策略评估

2.1 AB 测试概述

实证中由于不可能同时观测到同一群体在不同策略下的两种潜在结果，无法决定哪个策略胜出，需要构建一个反事实（Counterfactual）用来代表接受策略 B 的群体在接受 A 策略时的潜在结果。

具体来讲，构建一个与实验组群体特征均值无差异的对照组，用其观测结果代表实验组群体在施加 A 策略时的潜在结果，此时两种结果的均值差便是策略效应大小。由于是基于样本的观测数据得出的结论，需要通过显著性分析（Significance Test），以证明结论具有统计意义，这便是策略评估的完整路径。

根据能否在实验前控制策略的分配，我们将实验分为 AB 实验和观察性研究（Observational Studies），在 AB 实验分支下，根据能否控制策略的随机分配，又将 AB 实验分为随机对照实验（Randomized Experiments）和准实验（Quasi Experiments）。不同的实验类型使用不同的分组方法，在一定程度上影响着实验后分析数据的表现形式，实验后选择与实验类型匹配的分析方法尤为重要，直接制约着我们能否统计意义上的科学结论。具体分类如下：



实验类型	定义
随机对照实验	将研究对象随机分组，对不同组实施不同的干预，以对比效果的不同。
准实验	准实验是指不能随机分配实验对象，也不能完全控制无关变量，只能尽可能的予以条件控制的实验。
观察性研究	观察性研究的研究，在自然状态下对研究对象的特征进行观察、记录，并对结果进行描述和对比分析。

图 2　履约业务下的三种实验类型

对于大部分的实验场景，我们可以在实验前控制对不同的实验对象分配不同的策略，然而在有些场景下，如：①测试线上演唱会活动对短视频平台的影响，考虑到用户公平，需要给全部用户施加演唱会活动策略；②在测试不同的营销邮件策略对用户影响的场景中，我们无法控制哪些用户会最终接受策略。我们要么不能控制策略分配，要么不能控制策略在对应的人群生效，只能采用观察性研究，即在自然状态下对研究对象的特征进行观察、记录，并对结果进行描述和分析。

在我们可以控制对实验对象施加策略的场景，如①测试不同的产品 UI 对用户的影响，进而决定使用哪种 UI；②快速验证首页商品列表图素材对转化率的影响。这些典型的 C 端实验场景，不仅有海量用户且用户在实验组、对照组间的行为不会相互影响，可以通过随机分组的方式找到同质且独立的实验组和对照组，这类实验称之为随机对照实验，是业界衡量策略效应的黄金标准。

然而在美团履约业务场景中，如调度场景，要测试不同的调度策略对区域内用户体验的影响，策略施加单位是区域，由于区域数量少，同时区域之间各项指标（商家、运力、

消费者）差异较大，采用随机分组难以得出同质的实验组、对照组，而且由于区域之间可以共享运力，施加不同策略的实验组、对照组区域之间相互影响，不满足实验单位独立的条件。在这种场景下，我们不能对实验对象进行随机分配，只能有选择的进行实验组和对照组的分配，这种虽然能够控制策略分配但不能控制策略随机分配的实验，我们称之为准实验，常用的准实验方法如双重差分。

随机对照实验，因为其能够保证实验组、对照组两组的特征均值相同，不会因为分组差异干扰对真实效应的衡量，是业界衡量策略效应的黄金标准。在不满足随机对照实验约束的业务场景下衡量策略效应，我们采用准实验的方法，通过改进分组方法消除实验组、对照组可观测特征的差异或使其保持恒定差异，分析环节采用适配准实验场景的分析方法。

如果由于场景约束，只能基于实验后得到的数据来进行实验的话，就只能采用适用于观察性研究的方法。准实验和观察性研究虽然不是衡量策略效应的金标准，但是如果使用得当，也可以得出相对科学可信的分析结论。在学界，三种不同实验类型的可信度等级如下：

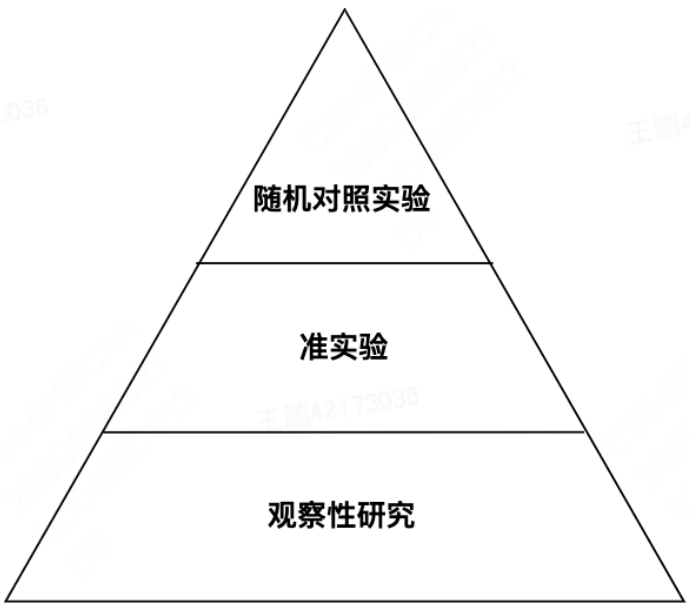


图 3　用于评估 AB 测试质量的可信度等级

2.2 AB 测试的关键问题

不管何种类型的 AB 实验，都符合分流 –> 实验 –> 数据分析 –> 决策的基本流程，以

及需要满足 AB 实验的 3 个基本要素。分流是实验平台的顶层设计，它规范和约束了不同实验者如何在平台上独立运行各自实验而不相互影响，运行实验，看似简单，但是成功运行不同类型实验的前提是实验场景要满足其理论假设。

AB 实验主要是通过观察抽样的样本来推断总体的行为，属于预测型结论，数据分析涉及大量的统计学理论，稍有不慎，容易掉入统计陷阱。上述流程，任一环节出错，都可能导致错误的结论，因此，AB 实验统计一个数字容易，得到可靠可信的统计结论并不容易。

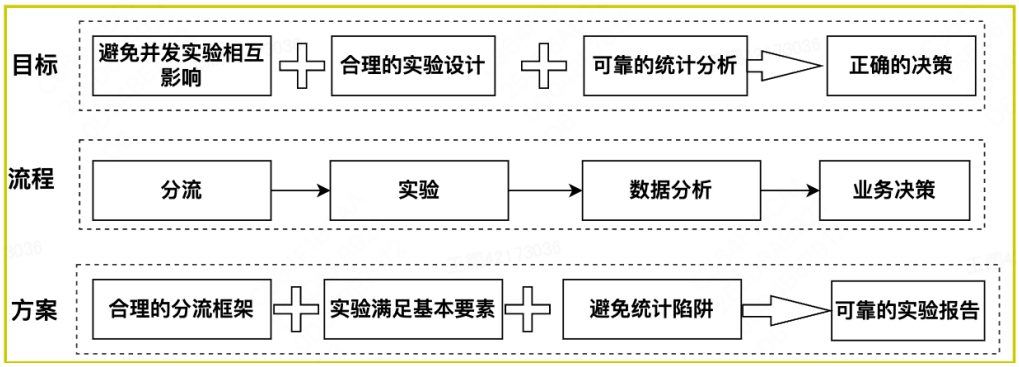


图 4　构建可信 AB 测试的关键要素

2.2.1 AB 测试的分流框架

在履约技术平台，我们通过实验衡量真实的用户反应，以确定新产品功能的效果，如果无法同时运行多个并行实验，将会大大减慢迭代速度。扩大同时运行实验的数量对于实现更快的迭代是必不可少的。为了增加可以同时运行的实验数量，提高并行性，并允许同时运行多个互斥实验，业界出现了两种分流框架，一种是像谷歌、微软、脸书这种单边业务形态的公司，采用层、域嵌套的分流框架；另一种是像 Uber、DoorDash 这种多边业务形态的公司，采用基于约束的分流框架。具体如下图所示：



图 5　业界流行的两种分流框架

基于层、域嵌套的重叠分流框架：该分流框架的特点是事先将流量随机打散做桶号标识并提前规划流量的用途，如上图所示，提前将全国流量划分为 10 等份并用 1 到 10 桶号来标识流量，1–6 号桶的流量用于短期策略验证，7–10 号桶用于长期性策略验证。为了支持同时运行多个互斥实验，提升迭代效率，分别在 1–6 号桶和 7–10 号桶中，进一步区分了正交桶和互斥桶，落在正交桶中的流量可以同时进入多个实验，在每进入一个实验前，重新打散一次流量，避免上一实验的残留效应做下一实验造成影响，实现了多个互斥实验的并行运行，落在互斥桶的流量，每次只能进入一个实验，用于运行不满足随机打散条件的实验。基于桶号划分的用于特定用途的一组流量集合，我们称为域；同一流量进入的不同类型的实验，我们称为层。

该分流框架的优点是不仅可以实现流量复用，扩大实验并行度，而且较容易避免具有潜在交互作用的实验可能给用户带来的糟糕体验。引入层的概念，将系统参数划分为多个层，约束让组合在一起可能产生较差用户体验的实验必须位于同一层，同一用户可进入不同层多个实验以实验流量复用，并防止其进入同一层多个实验，避免糟糕的用户体验。

不足之处在于：首先，这种分流框架的一大前提是提前将流量打散，这种在大流量的单边场景下尚可，在小流量的多边场景下，难以行得通。多边场景下，考虑到溢出效应，无法直接采用单边实体进行分流，而是采用聚类的方式将有相互作用的多边聚合成一个大的实体，基于大的实体进行分流，考虑到有限的实体数量，这种提前打散的方式难以获得均匀的流量；其次，域提前规定了流量用途，这种提前隔离的方式，降低了流量利用率，无法满足小流量场下实验功效要求，如即使在互斥域中没有实验，也无法采用该流量进行其它的正交实验；第三，这种提前预规划流量用途的分流框架灵活性不足，如果后期发现域的设置不合理，要改变域的配置将付出较大的代价。

基于冲突检测的分流框架：该分流框架的特点是由实验者制定约束，平台根据实验者制定的约束，确保无法避免潜在交互影响的实验没有同时曝光给用户。如微软、Uber 等公司，实验平台都集成了检测交互作用的自动化系统，以避免实验间潜在交互影响。以 Uber 为例，将策略看作是一组独立参数的集合，并提前声明对应策略涉及的专用参数和与其它策略共享的参数，配置实验时检测是否有任何影响相同参数的两个实验重叠，只要它们没有重叠，便允许实验创建或更新操作。

该分流框架的优点是灵活且能最大限度的复用流量，相较于重叠流量框架，不受提前划分的域的约束只能在特定域中进行实验，即使对应的域中此时并无其它实验。只要满足进行并行实验的条件，便可以任意圈定流量进行实验；不足之处在于：实验平台需要构建自动检测交互作用的能力。

2.2.2 AB 测试满足的基本要素

在运行 AB 实验时，要满足三个基本要素：①施加不同策略的实验组、对照组可比较，即实验组、对照组的特征均值在实验前相同或在实验前具有固定差异，便于实验后计算哪些差异是由于策略不同导致的；②策略之间没有干扰，实验群体相互独立，即当我们比较策略 A 和策略 B 时，接受策略 A 的用户行为不会受接受策略 B 的用户行为影响；③实验群体的数量要足够，以满足功效要求，避免实验结果假阴性，即实际策略有效果但是由于样本量不足没有检测出来。

不满足要素一，实验后难以确定实验组对照组的差异是由策略导致的还是由于分组导致的，难以准确衡量策略的真实效应；不满足条件二，可能会高估策略效应。举个例子，在履约配送范围实验中，橘色是实验组，商户 A 范围的扩大，会使用户的需求从商户 B 转移到商户 A，如果评估的是商户集合的单量，会造成实验组单量相对对照组多，实验环节，得出扩大配送范围，会提高整体单量，但是当策略在全国应用后，发现单量并没有明显增加，因为实验期间观察的增加仅仅是单量转移，实验组单量转移到了对照组。不满足条件三，难以确定策略无效果是真正的无效果还是由于样本量不足，没有检测到策略效果。

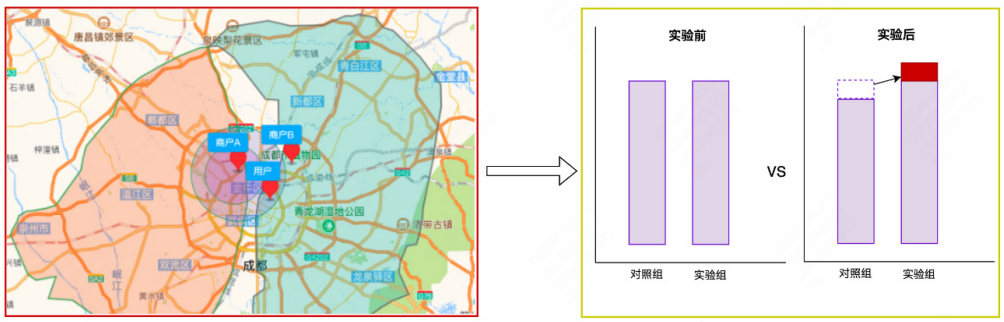


图 6　溢出效应导致的估计偏差举例

2.2.3 不可忽视的统计陷阱

AB 实验主要是通过观察抽样的样本来推断总体的行为，属于预测型结论，涉及大量的统计学理论，稍有不慎，容易掉入统计陷阱，难以得出可靠的统计结论。

实验组和对照组之间的差异是真实的还是噪音通过显著性检验来辅助判断，要得出结论涉及方差、检验方式和 P 值计算，这些环节充斥着统计陷阱，稍有不慎便会导致我们通过假设检验得到错误的结论。样本的抽样方式、分布特点以及样本量大小决定了我们的检验方式和采用的具体 P 值计算方法，实验单元、分析单元和实验组、对照组差值类型，决定了方差计算，方差作为 P 值计算的一个输入，直接影响着 P 值结果。上述环节，忽略任一因素，会导致 P 值计算错误，使我们通过假设检验得到错误的结论。

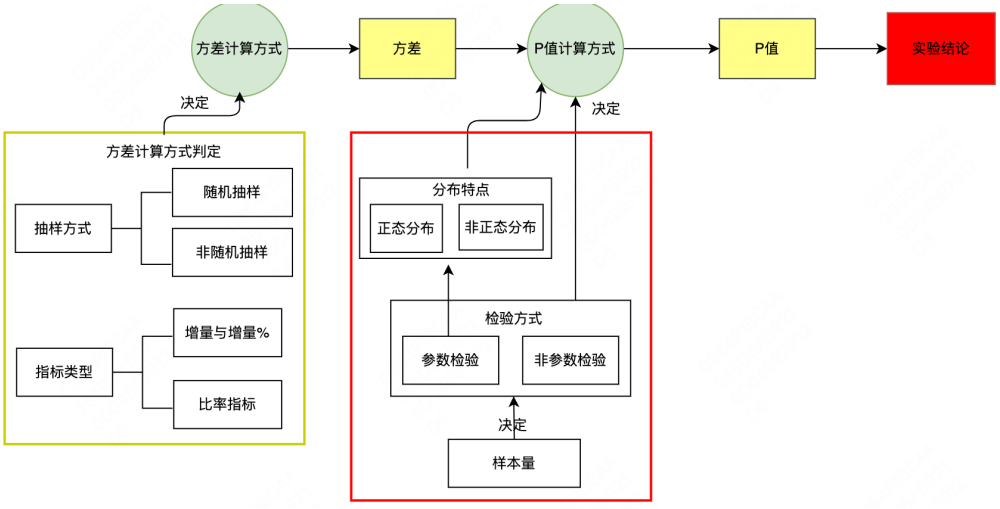


图 7　分析环节影响实验结论的因素

容易忽视的方差计算陷阱：如果不能正确的估计方差，那么 P 值和置信区间都将是错的，这些错误会导致我们通过假设检验得到错误的结论。高估的方差会导致假阴性，而低估的方差会导致假阳性。下面是几个估计方差时的常见错误。

样本在实验组或对照组的分配机制影响着其方差的计算，避免用随机抽样（样本满足独立同分布）下的样本方差计算方法计算非随机抽样下的样本方差。该陷阱主要出现在用随机对照实验下的方差计算方法计算准实验下的方差，导致方差计算错误。计算 Y_t-Y_c 的方差，按照方差性质，非随机抽样下，若 Y_t 和 Y_c 不独立：

$$D(Y_t-Y_c)=D(Y_t)+D(Y_c)-2Cov(Y_t,Y_c)$$

在随机抽样下，Y_t 和 Y_c 独立：

$$D(Y_t-Y_c)=D(Y_t)+D(Yc)$$

两者不相等。例如，在按天交替轮转的轮转实验中，一旦确定了实验开始第一天在实验组还是对照组，后续其它天在实验组和对照组相继确定。此时，实验组、对照组两组样本不独立，如果按照独立的方法计算方差，将错误估计方差。实际上，样本的分配机制影响着其方差计算，在 AB 测试中，我们将流量划分为实验组和对照组并在实验组施加策略，然后计算实验组相对对照组的某个度量指标的绝对提升值或者相对提升率，并检验该差异是否存在统计上的显著性，进而判断实验策略是否真实有效。

例如对于骑手运单量指标Y可通过计算实验组与对照组的样本均值差$\hat{\Delta}=\overline{Y_t}-\overline{Y_c}$来估计实验效应，并构造两样本t-统计量$\Delta/\sqrt{var(\Delta)}$对原假设$H_0$:实验组对照组无差别进行双边假设检验。为方便讨论，先不考虑Y随机性，仅考虑分组W随机性， 并且假设实验组样本量N_t，对照组样本量N_c固定。

$$\hat{\Delta}=\frac{1}{N}\sum_{i=1}^{N}(\frac{N}{N_t}*w_i*Y_i(1)-\frac{N}{N_c}*(1-w_i)*Y_i(0))$$

样本在实验组$w_i=1$，样本在对照组$w_i=0$，令$D_i=w_i-\frac{N_t}{N}$，与分配机制有关，若：

$$Y_i^+=(\frac{N}{N_t})Y_i(1)+(\frac{N}{N_c})Y_i(0)$$

则：

$$\hat{\Delta}=\frac{1}{N}\sum_{i=1}^{N}(Y_i(1)-Y_i(0))+\frac{1}{N}\sum_{i=1}^{N}D_i*Y_i^+=\Delta+\frac{1}{N}\sum_{i=1}^{N}D_i*Y_i^+$$

若想保证$\hat{\Delta}$为真实效果Δ的无偏估计，实际上分配机制要保证如下公式成立：

$$E_w(\frac{1}{N}\sum_{i=1}^{N} D_i * Y_i^+) = 0$$

实际中，如果分配机制w保证$\hat{\Delta}$相和性，在原假设下$E(\hat{\Delta}) = 0$，$\hat{\Delta}$方差计算为：

$$E_w[(\frac{1}{N}\sum_{i=1}^{N} D_i * Y_i^+)^2] = \frac{1}{N^2}\sum_{i=1}^{N}(Y_i^+)^2 * E_w[D_i^2] + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j\neq i}(Y_i^+ * Y_j^+) * E_w[D_i D_j]$$

从上述公式可以看出，方差计算与分配机制有关，如果忽略分配机制将导致错误的方差计算。

评估相对提升或实验单元与分析单元不一致时，错误的方差计算方式容易低估实际方差，导致假阳性。在计算指标的相对提升率，如下公式所示：

$$\Delta\% = \frac{\overline{Y}_t - \overline{Y}_c}{\overline{Y}_c} = \frac{\Delta}{\overline{Y}_c}$$

相对提升率的方差往往使用错误的公式如下所示来估计方差：

$$\frac{var(\Delta)}{\overline{Y}_c^2} = \frac{var(\overline{Y}_t) + var(\overline{Y}_c)}{\overline{Y}_c^2}$$

这是错误的，因为分母 $\overline{Y}_c$ 本身也是一个随机变量。另外当分析单元比实验单元更细时，例如，骑手有单均时长指标，在骑手活动的实验中，实验单元为骑手而骑手单均时长指标（所有运单时长的平均值）的分析单元为运单，分析单元为比实验单元更细的方差时，Deng等(2011)与Deng等(2016)指出此时分析单元样本间的独立性不再成立，因为运单1、2、3来自同一个骑手k的相关行为时，运单时长指标Y_{k1}、Y_{k2}、Y_{k3}应该是相关的。此时应重写样本均值为：

$$\overline{Y} = \frac{\sum_k(\sum_i Y_{ki})}{\sum_k N_k} = \frac{\sum_k S_k/K}{\sum_k N_k/K}$$

此时的方差应为：

$$\frac{1}{K\mu_N^2}(\sigma_S^2 - 2\frac{\mu_S}{\mu_N}\sigma_{SN} + \frac{\mu_S^2}{\mu_N^2}\sigma_N^2)$$

其中S_k为第k个骑手的运单时长Y_{ki}之和，N_k为第k个骑手的运单量，K为骑手个数。另外对于比率型指标等更复杂的指标，正确的方差估计公式往往需要统计学中的Delta方法等推导得到。

容易忽视的检验方式导致的 P 值计算陷阱：统计学对于多大样本量即可认为中心极限定理成立并没有完全的定论，并非所有大样本场景下的样本分布都满足正态性假设，避免

有偏样本采用默认正态分布下的检验方法。Weich t 假设检验是参数检验常用的一种检验方法，其本质上假定实验组、对照组样本均值等的渐近正态性成立，该理论实际上是建立在大样本情形下的中心极限定理基础上。统计学对于多大样本量即可认为中心极限定理成立并没有完全的定论，这实际上也取决于原始分布本身偏离正态分布的程度。

从经验来看，若样本仅稍微偏离正态总体，大于 30 的样本量或许就足够了。然而对于有偏样本，Ron Kohavi 等 (2014) 指出当样本偏度大于等于 1 时，一个经验准则便是只有计算样本均值的观测样本量大于 *355s2355s2* 时才可认为中心极限定律成立。实际抽取了一个样本量为 13832 的活动实验，其实验组、对照组差值的抽样分布呈现右偏，不符合正态分布，如下图所示：

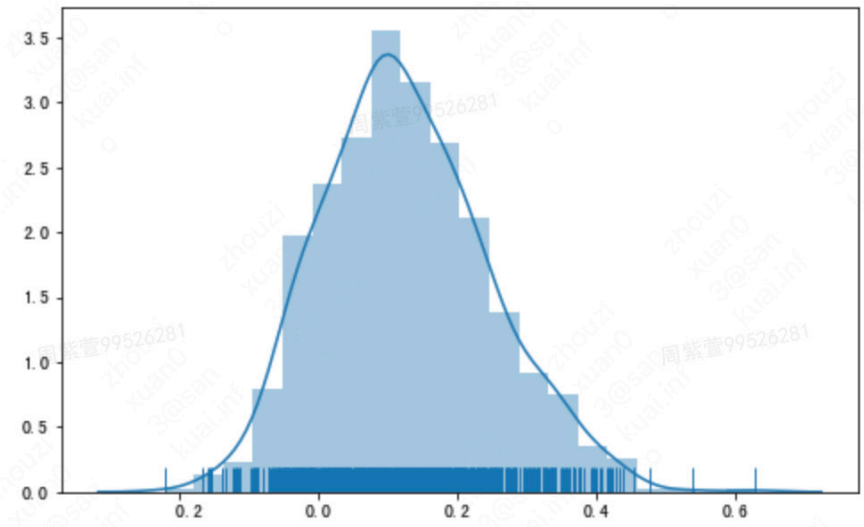


图 8　数据分布的偏态举例

如果所有场景下默认采用正态分布情形下的检验方式计算 P 值，容易导致错误的 P 值计算。

2.3 基于一组核心抽象的平台建设难以适配所有业务场景

整个 AB 实验的过程涉及大量统计学知识，正确运用书本理论知识的前提是实际业务场景满足理论假设，实际情况是很多场景不满足理论假设。在这种情况下，获得高质量的结果需要对实验和统计有专家级的理解，以及大量的工作包括：实验设计、配置、指标加工、自定义分析等流水线工作，任一环节出错，将会导致大量的工作浪费。

通过核心抽象为实验者输出不同方法能力的平台建设思路，难以避免方法使用不当导致的实验置信问题，在此过程中任一设计的微小偏差，会导致无法比较的实验组、对照组，从而影响实验结果。例如，事例一方差估计错误：在实验分析时，经常犯的一个错误，不管分组方式是不是随机分组，在实际分析时，仍然按照样本满足独立同分布的条件计算方差，造成我们对估计值的准确性过度自信，低估了方差，容易犯假阳性的错误。

一个极端的例子，随机抽取 100 个学生用于估计该需要的平均成绩，如果被抽查的 100 人都是同一个学生，他们的成绩只反映了一个学生的成绩，对于估计所有学生平均成绩的信息含量等同于一个学生提供的信息。

如果我们把它们当成独立的，所得的样本平均值的标准误差显然是不对的，其结果是造成我们对估计值的准确性过度自信，即估计值的标准误差估计过小。

事例二业务场景不满足理论约束：双重差分是我们准实验中常用的分析模型，它的计算过程很简单，即实验组干预前后的均值的差减去对照组干预前后均值的差，根据业务场景，可以选择传统 DID 模型或固定效应的 DID 模型，但具体哪种模型合适，需要进一步看，在当下的业务场景下，哪种模型满足平行趋势假设，即在没有干预的情况下，实验组和对照组指标的均值差异在不同时间内保持一致，在都满足平行趋势假设下，哪种模型更优？如果不进行严格的检验，将会导致有偏估计。如下是我们具体场景下的案例：

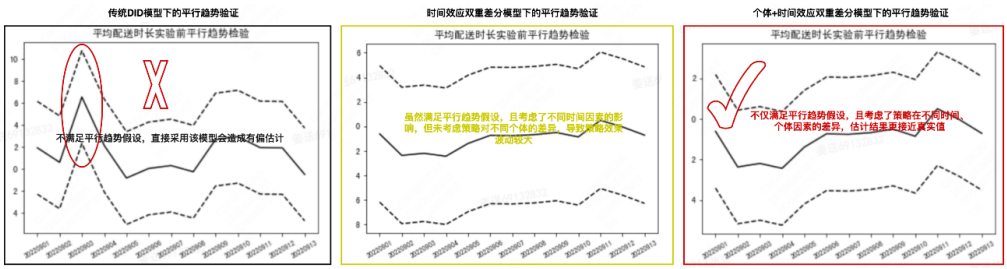


图 9　准实验场景下选择双重差分模型需要考虑的因素

虽然根据大致的场景特点，可以判断出采用双重差分模型，但是到底采用双重差分的哪个模型，还需要基于实际数据，做进一步验证和选择。根据上图所示，在当前的业务场景下，并不满足传统 DID 模型的平行趋势假设，如果贸然使用，会造成估计偏差，时间效应的双重差分模型和个体 + 时间效应的双重差分模型，虽然都满足平行趋势假设，但从实际置信区间看，后者因为考虑了策略对不同个体的差异，波动较小，估计结果更

加接近实际值，所以应采用后者。

3. 在履约我们如何进行 AB 测试

3.1 多边业务模式下面临的 AB 测试难题

溢出效应和小样本是当前业务场景下实验面临的最大挑战，其次策略施加的公平性约束了实验分组也是我们不得不面对的挑战。每个因素的单独制约，对得出置信实验结论将是不小挑战，而履约场景下，这些因素综合的叠加在一起，加剧了挑战。

我们的即时配送物流系统在多边市场中扮演着交易中间人的角色，它通过平台匹配用户、骑手、商家三方的需求，平台通过产品策略优化这一匹配过程，每一次匹配都会对同一时刻及后续一段时间内其它的匹配产生影响，具有较强的溢出效应。受溢出效应影响，实验单元的实验结果不仅取决于个体本身，还会受其他实验单元的影响。网络效应的存在违反了实验单元独立的原则，导致有偏的实验结果。

如履约业务下的商家配送范围实验，实验 Treatment（配送范围划分）直接决定了用户能否在某家商家下单，且同一时空下不同商家共享相同的用户，某一策略的施加会导致原本只能在商家 B 下单的用户可以在商家 A 下单，导致原本在对照组中的单转移到了实验组，虽然从实验效果上看，策略提升了单量规模，策略推全后，效果并未到达实验的先验预期甚至无效果。因为这种提升可能仅仅是单量转移，并非策略真正带来了提升。

履约 LBS 的业务形态决定了其大部分策略都是地域（主要是配送区域）展开的，受限于配送区域数量及本身的地域差异，难以获得足够的样本检测出策略的小提升。如调度实验，受限于自身业务形态和空间维度限制，调度算法的最小作用单元为区域或区域组，实验必须考虑区域或者更粗粒度的分流，然而大部分城市区域和区域组很少，并且城市各地域间的差异往往比较显著，这在数据上体现为区域间指标波动剧烈。

该场景下严峻的小样本与地域间差异显著的问题导致统计功效低，从而很难有效地检测出策略小的提升效果。其也会导致随机分流下与响应变量相关的协变量在实验组、对照组的分布差距较大，放大业务上实验组对照组不同质问题的同时给实验结果带来质疑。

更致命的是，该场景下的混合调度模式，不同运力类型的重叠区域可以共享运力互派单，区域可以召回其附近的其它区域运力并派单的特点带来的溢出效应，会导致实验效

果估计不够精确甚至带来显著的估计偏差。类似调度实验同时要克服小样本和溢出效应的双重约束，是不小的挑战。

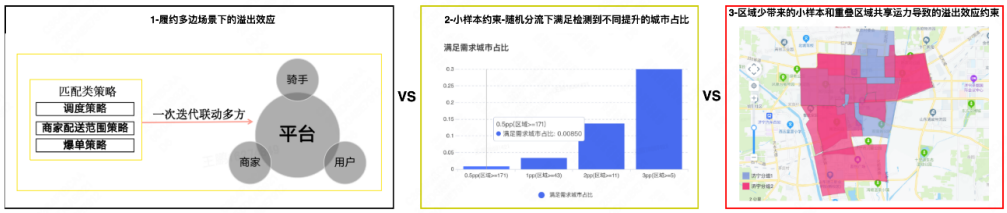


图 10　履约业务模式下面临的 AB 测试难题

3.2 AB 测试的组织和流程

随着履约业务的发展，我们越来越依赖于良好的策略驱动业务规模快速发展，以及效率、体验和成本的持续优化。A/B 测试提供了最科学的方法来评估策略变化的影响，并绘制出清晰的因果关系。通过 A/B 测试量化影响，最终辅助团队做出决策。我们将人员、流程和平台更紧密地结合在一起——这是成功实验生态系统的基本要素。

人员方面，我们将算法（实验用户）、算法工程、数据科学（下称数科）有机组合成了一个虚拟团队，数科同学在策略迭代之初，就参与到算法年度目标的讨论中，辅助算法一起制定量化策略好坏的综合评估指标，并基于场景特点选择合适的实验方法，完成对应场景下的实验设计，算法工程同学，负责将新方法集成到实验平台，作为公共能力为用户提供服务。

人员	角色	职责
算法	用户	构建策略想法、运行实验、发起实验决策
工程	平台建设	负责实验平台建设，为实验提效率
数科	实验指导	通过数据测算为算法构建策略想法提供基础输入、在实验环节，完成实验假设、实验设计并提供分析方法

有了组织、平台后，构建高效、基于 AB 实验的数据驱动工作流是通过 AB 实验帮助我们达成产品目标获得成功的关键，我们将整个流程分为三个阶段：构建想法，通过 AB 实验验证想法、沉淀知识库形成实验记忆。

构建想法是实验的输入阶段，构建想法的质量直接决定了实验的效果，如果这个阶段构

建的想法不够好，那么 AB 实验阶段只能起到验证错误的作用，降低犯错误的概率，无法带来增长。

验证想法就是实践 AB 实验的过程，可以分为实验假设、实验设计、实验运行、实验分析和实验决策五个关节，实验假设环节，即形成实验目标，构建综合评估指标，实验设计，基于场景约束，选择合适的实验方法。

最后，通过 Launch Review 发起实验决策；将成功和失败的案例沉淀下来，形成实验记忆，不仅可以帮助我们发现策略的通用性，而且有助于帮助我们从失败中寻找机会。

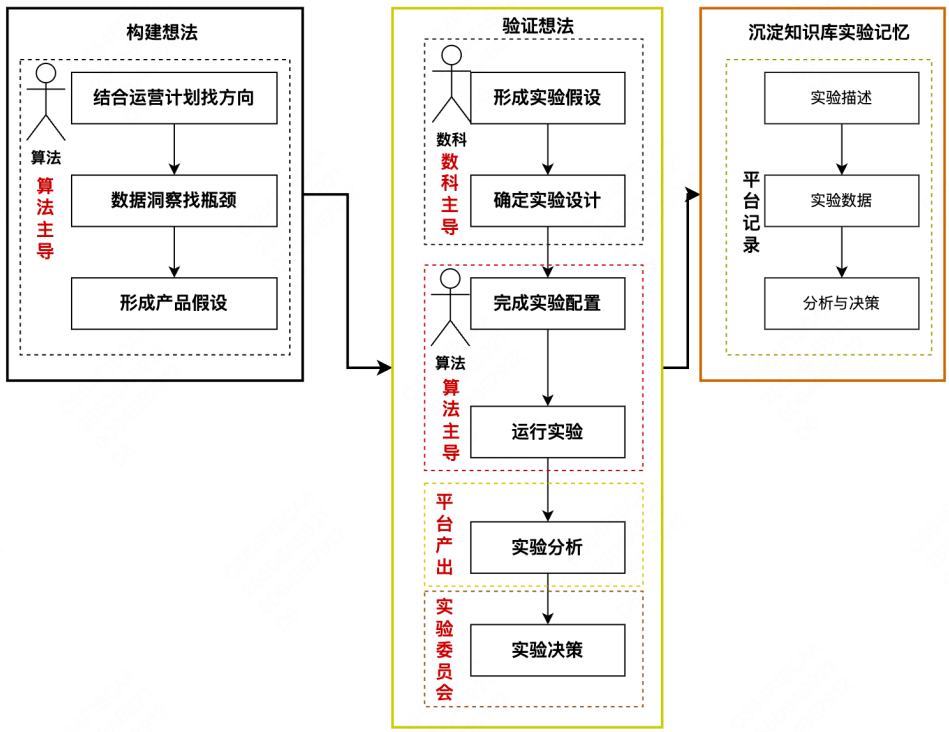


图 11　履约 AB 测试流程

3.3 AB 测试平台简介

3.3.1 平台概述

AB 实验得以在工程中广泛应用和推广，与 AB 实验的并行性（多个实验可并行开展）和先验性（通过小流量预先获得效果评估）密不可分，分流框架直接决定了实验的并行度，与场景匹配的实验解决方案直接影响着实验先验结论的可信度。

为提高实验并行度并让我们同时运行多个互斥的实验，我们构建了基于约束的分流量框架，以规范和约束不同实验如何共享和使用流量。为确保平台提供可靠的实验结果，平台针对实验设计直接输出解决方案而非能力，实验分析完全自动化，即基于实验设计和数据特点自适应选择与之匹配的方法。以达成不管是实验或统计领域的专家进行实验，还是无统计和实验知识的普通用户进行实验，任何人都能够相信实验的结果。

为提高实验并行度，业界有层、域嵌套的重叠分流框架和基于约束的分流框架，前者以谷歌为代表，后者以 Uber、微软为代表。层、域嵌套的重叠分流框架要求提前将流量均匀打散并规划好用途，这不仅需要大流量以确保能够被均匀打散，而且对业务未来演进需要有精准预判确保合理划分流量用途。小流量无法实现均匀打散，流量用途划分不合理，不仅会导致分配流量大的域实验数量少而浪费流量，分配流量小的域实验数量多导致流量不够用而排队；而且重新划分流量用途会导致线上实验失效、新策略无法正确推全、无法进行长期实验。

在履约技术平台，分流单位经常是区域、区域组甚至城市，样本量有限，不满足均匀打散的样本量要求，其次，履约业务持续演进变动，难以基于业务预判提前规划好流量用途，基于以上两点，履约采用了基于约束的分流框架。

在实验领域使用统计方法统计一个数字容易，但是确保统计方法合理适配得出可靠的实验结论并不容易，特别是在履约这种连接用户、骑手、商家三边的平台型经济业务模式下，不同的实验需要在降低网络效应和提高实验功效两个目标之间权衡，制定与场景匹配的实验方法并得出置信实验结论。

做到这一点，需要实验者对实验和统计有专家级的理解。为降低实验门槛，确保实验置信，平台建设提出了，针对实验设计直接输出解决方案而非能力。针对实验分析，实现完全自动化，避免实验者将大量精力放在方案的论证上和人为因素导致的实验置信问题。

3.3.2 基于约束的分流框架，以适配履约业务场景

分流框架像法律法规规范着规范着大家的日常行为，使大家在社会大家庭中有序生活一样，它规范和约束着不同的实验在不相互影响的前提下如何共享和使用流量，它是实验平台的顶层设计。基于约束的分流框架让实验者指定约束，平台冲突检测根据实验者指定的约束，进一步判断是否允许实验。在展开之前，先引入三个概念：算法 Key、场景和实验模版。

算法 Key 代表一组可独立测试的功能，在技术层面可以表示为一组独立参数集合，场景代表对应算法 Key（对于联合实验而言，是多个算法 Key；非联合实验，是一个算法 Key）下具有相同实验模版的实验集合，实验模版为一组相同实验类型、实验单位、分组方法、评估方法的配置。

考虑到：①同一算法 Key，不同实验是针对同一功能不同版本的测试，实验间要互斥；②不同算法 Key 之间，只要其对应的功能之间没有潜在的交互作用，其对应的实验间天然正交可以放心的复用流量，如存在潜在交互作用，只要确保流量能被随机打散，便可消除策略间的潜在相互作用对实验结论的影响。

因此，针对并行实验，初步的约束如下：①同一算法 Key 下的任意两个实验不能复用流量，冲突；②不同算法 Key 下存在潜在交互作用的两个实验，只要有一个实验类型是随机对照实验，皆可复用流量。约束②不仅避免了全因子流量框架不同策略实验间潜在相互影响的风险，而且避免了重叠流量框架因不同域流量隔离导致的流量复用率低的问题，特别是在准实验、观察性研究比随机对照实验多的情形下，由于准实验、观察性研究分处不同的域，无法实现随机对照实验和准实验、观察性研究之间的流量复用。

考虑到同一算法 Key 下不同实验因目标流量或迭代验证的功能不同，同一个算法 Key 下的不同实验与另一算法 Key 下的不同实验间是否冲突取决于其对应的测试功能或实验方法，我们引入场景来描述不同算法 Key 的功能描述和其对应的实验方法，并根据业务经验构造不同场景间的业务影响矩阵。基于不同场景的业务影响矩阵、场景实验方法和并行实验约束，生成场景实验冲突矩阵，基于此矩阵完成不同算法 Key 实验间的冲突检测。

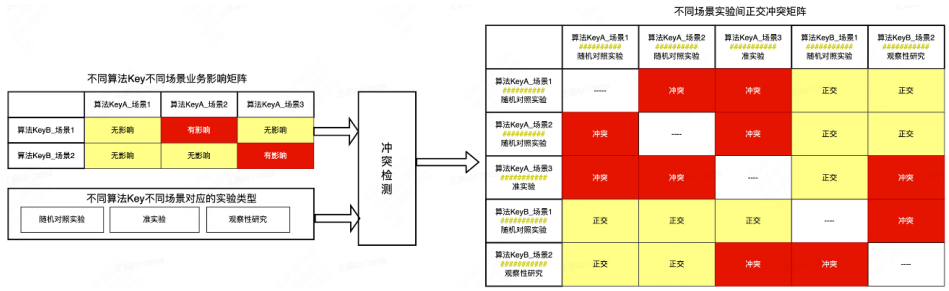


图 12　履约实验平台分流框架

不同场景下的联合实验，与其对应算法 Key 下所有场景实验冲突，与其它算法 Key 场

景实验，根据约束 2 进一步判定；为避免冲突实验间的流量重叠，提供了基于表达式定义流量范围的能力，通过检测表达式流量覆盖范围避免冲突实验间的流量重叠。基于约束的分流框架，不预先规划流量用途，也没有层、域复杂概念，实验时按需选择流量，只要通过冲突检测，就可以上线实验，不仅降低了用户使用门槛，而且提高了平台灵活性，以适应履约业务场景。

3.3.3 打包输出实验设计，降低实验门槛，确保实验质量

受履约溢出效应、小样本和公平性因素制约，实验设计是降低溢出效应、提高实验功效、关注实验公平等多种目标进行方差和偏差平衡的过程。

实验者虽然可以根据对应场景所允许的分组方式（是否允许分组以及在允许分组的前提下是否允许随机分组），初步判断可以考虑应采用如下“普通随机对照实验”、“随机轮转实验”、“准实验”、“观察性研究”中的哪一种实验方式，但是具体应该采用哪种实验方式以及对应实验方式下应该采用什么分流单元，是综合考虑溢出效应、实验功效、公平性等因素多方平衡的结果。

例如：在运单实验中，实验组、对照组运单可以来源于同一区域，由于同一区域的运单可以共享骑手，运单间不独立，导致实验组、对照组存在溢出效应。轮转实验是解决该问题的一个可选项，前提是我们需要在如下两个相互冲突的目标之间做平衡。

- 我们希望可以划分更多的实验单元来增加样本量，这就需要我们将实验单元划的足够小以得出更多的实验单元来保障我们有更多的样本量来满足实验灵敏度要求。
- 我们希望实验单元划的足够大确保将相互影响的个体包含在一个独立单元中，以消除溢出效应对实验结果的影响。

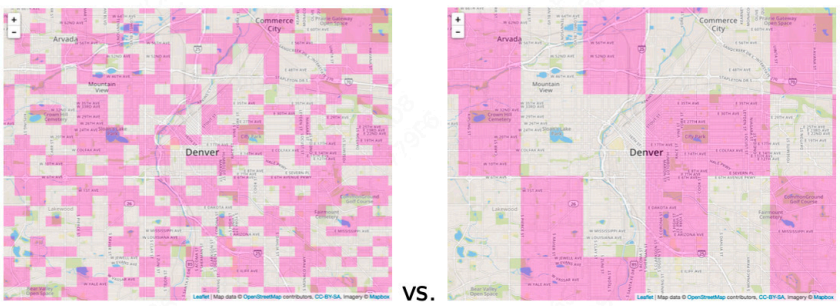


图 13　轮转实验 – 在样本量和溢出效应之间平衡

在有限的样本下，如果只是进行简单的随机分组，不仅会导致实验组对照组的一些指标在实验前存在偏差，而且会由于样本量不足导致无法检测出策略的微小提升，我们到底是通过控制影响指标差异的协变量和改进分组方式来达成偏差和方差的平衡，还是实验前允许偏差存在，通过实验后纠偏的方法进行补充，这些都需要在实验设计时基于算力，以及基于分组方式和分析方法组合方案得出的数据表现综合判断，来制定合理的实验方案。

为数科同学提供一系列实验方案的设计工具，辅助其完成实验设计中关键的方差和偏差平衡，输出与场景匹配的解决方案。由于用户、商家、骑手通过履约平台形成的复杂交互关系，在履约进行 AB 实验，需要去权衡溢出效应和实验功效，给出合理的实验设计，这不仅需要从事该工作的人对实验和统计有专家级的理解以及足够的时间投入，而且需要平台提供实验设计的能力，可按需选择实验类型、实验单位、分组方式、分析方法、评估指标，完成实验设计并验证该设计的可行性。

在履约技术部，由数科同学承担实验设计的职责，由其为对应场景制定与之匹配的实验方案，释放算法同学精力，让其有更多的时间思考如何进行策略迭代。为满足履约各种场景实验诉求，平台提供了如下几种类型实验模版：

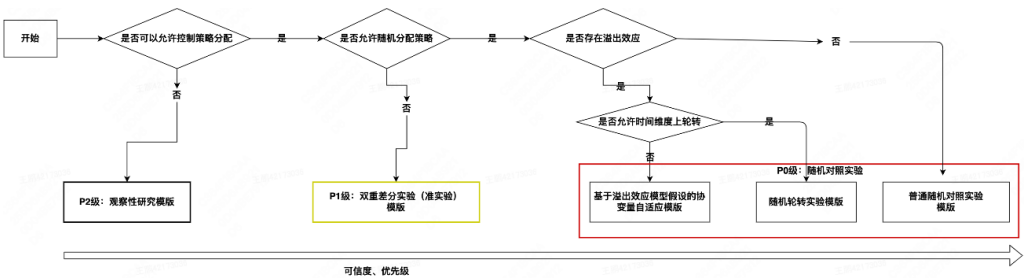


图 14　实验平台为数科同学推荐的实验设计模版

为了便于数科同学基于要检测的指标、对应指标的预计提升量、可用样本等约束进一步确定对应实验模版下的实验单位、具体的分组方法和分析方法，制定出与场景匹配的实验方案，平台为其提供了实验设计工具包，包括：分组工具包、降方差工具包、显著性分析工具包，实验设计报告工具包，包括：MDE 分析、同质性检验、样本量预估等工具包。

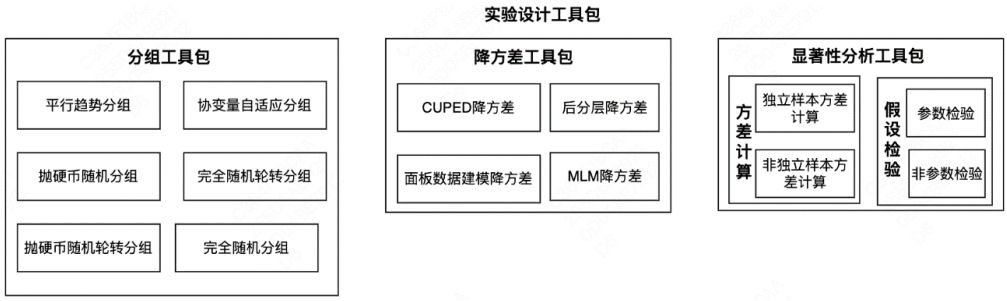


图 15　实验设计工具包

平台为实验者直接输出与其场景匹配的实验设计方案，实验者无需担心实验方法使用不当导致的置信问题。在平台的场景管理模块，配置着该场景下具体的实验方案：具体的实验类型、实验单位、分组方式、评估指标（包括：目标指标、护栏指标、驱动指标）、降方差方法，实验者进行实验配置时，根据平台为其提供样本量预估和“MDE”分析工具，完成流量圈选和实验周期确定，之后平台输出实验设计报告，输出分组结果、“MDE”分析和同质性检验报告，检验通过后，实验者可进一步配置实验组、对照组对应的策略参数，完成最终的实验配置，进而发布实验。

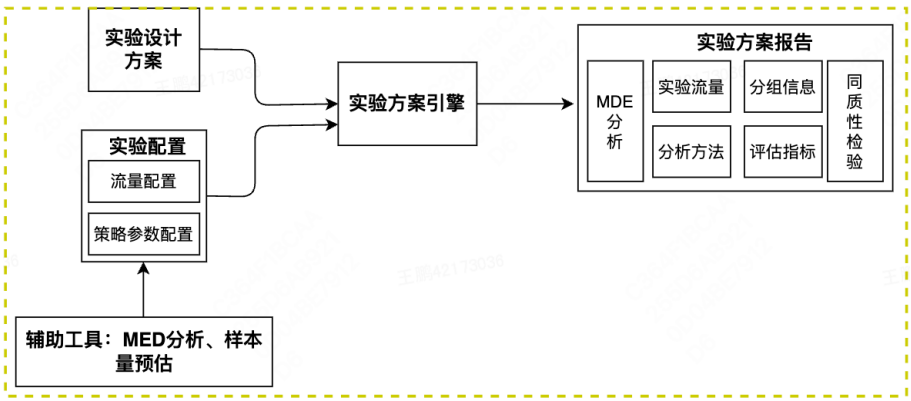


图 16　平台基于实验设计方案和实验者的实验配置，为具体实验确定实验方案

3.3.4 构建适用于不同实验方法的分析引擎，标准化实验分析

可靠的数据和科学的分析方法是得出可信分析的关键。实验分析涉及大量的统计学理论，稍有不慎，容易掉入统计陷阱，考虑到大多数实验者缺乏统计知识且自助分析带来的时间消耗，我们构建了统一的分析引擎，标准化实验分析过程，为不同的实验设计提供与之匹配的分析方法，通过验证相关指标的统计显著性和估计策略效应，帮助我们根

据分析结果做出数据驱动的决策。分析流程大致包括如下过程：

1. 通过数据诊断，确保分析数据的可靠性；
2. 基于分组方式、分析方法，进行实验效应估计，得出实验效应估计；
3. 基于分组方式、数据类型、实验单元与分析单元的关系、分析方法，选择合适的方差计算方式进行降方差以提高实验灵敏度，避免假阴性；
4. 基于分组方式、数据分布特点，选择合适的检验方式计算方差计算和 P 值，验证相关指标的统计显著性给出统计结论；
5. 基于诊断和分析，输出实验报告。

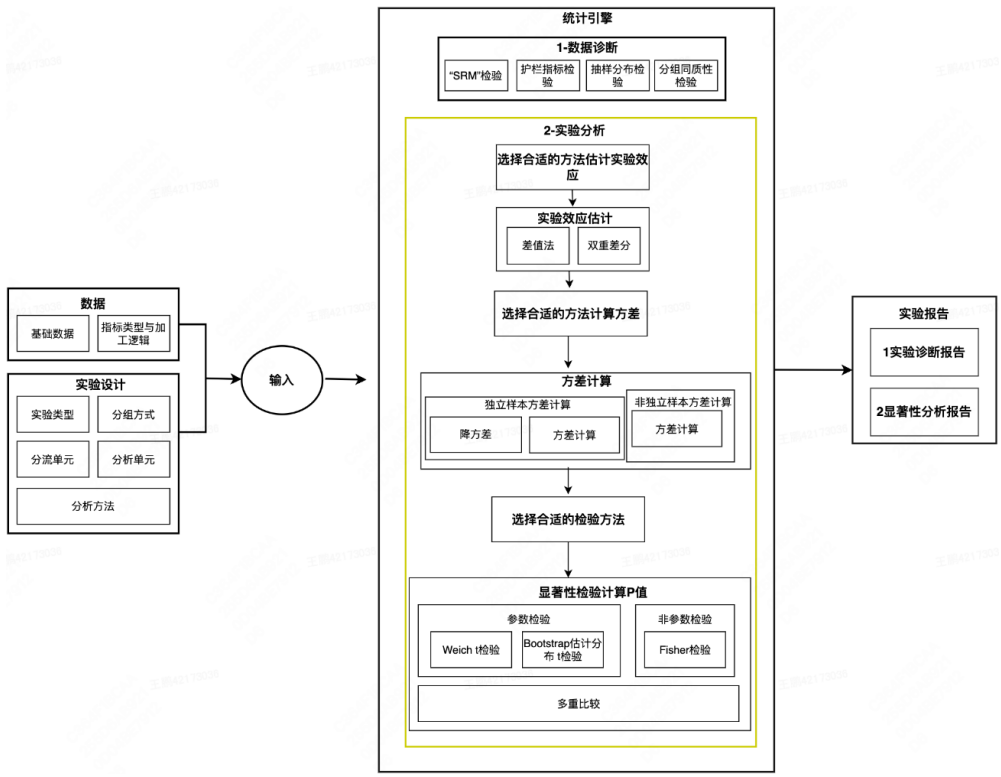


图 17　履约实验平台统计引擎

分析环节的数据诊断，旨在提醒实验者注意可能违反实验假设的情况。很多人认为实验一定按照设计运行，实际上这一假设失败的概率远高于人们的预期。失败实验的分析结论通常是有严重偏颇的，甚至一些结论是完全错误的。在输出显著性分析报告之前，通过护栏指标检验，确保业务不会因策略的迭代受到伤害，通过分组同质性检验、“SRM”检验查看实验执行是否符合预期，确保实验本身的可信度，抽样分布检验，为

后续选择合适的显著性检验方法提供依据。

分析环节的实验分析，自动选择与数据和实验设计匹配的分析方法，避免统计陷阱。根据分组方式，提供了差值法和双重差分两种效应估计方法，方差计算和 P 值计算是统计陷阱集中发生的两个环节，分别提供了判别方差和 P 值计算方式的判别引擎。

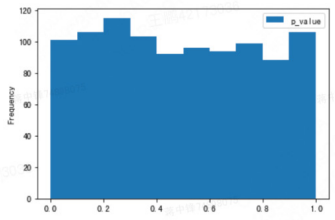
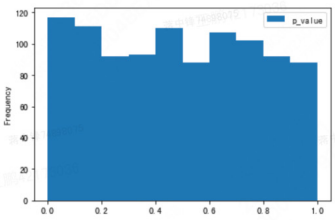
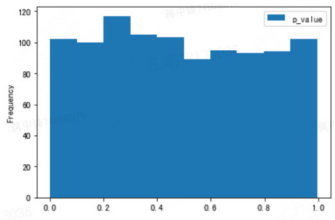
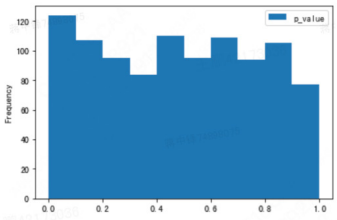
首先，方差计算根据是否随机抽样，将其分为独立样本方差计算和非独立样本方差计算，独立样本方差计算，根据指标是增量提升还是相对提升和分流单位与分析单位是否一致综合因素，提供了直接计算和 Delta 方法计算，避免方差计算陷阱，非独立样本，通过模拟数据的实际分布，给出方差的准确计算。

其次，在样本量小于 30 的超小样本下，采用非参的 Fisher 检验以满足功效要求，在样本量大约 1 万的超大样本下，接受检验统计量的渐近正态性成立并采用 Weich t 假设检验；在样本量大于 30 小于 1 万下，进一步样本实际分布情况，如果统计量渐近正态性成立，则采用 Weich t 检验，如果不成立，采用 Bootstrap 估分布进行统计推断。

3.3.5 平台建设不仅要根植于业务场景，而且要做到严格的质量控制，确保结果可信

履约实验平台建设拒绝功能的简单堆砌，而是设计了可灵活扩展的实验 Pipeline 工程架构，然后将更多的精力放在业务场景的实验设计和分析方案上，严格控制质量，确保实验结果可信。在此过程中，数科同学扮演了双重角色，作为平台建设的一员，他们永远比工程建设快半步，深入业务，定义新问题并找到答案，协同工程同学一起完成新能力建设。

同时，数科同学又作为一批特殊的用户，承担者平台质量把控和产品易用性职责，特别是质量把控，在平台对外发布新实验方法前，都需要通过他们的 AA 模拟，通过模拟几百次 AA 实验，查看关注指标的 P 值是否在 0 到 1 之间均匀分布，通过验证后如果符合新能力发布条件，进行新能力发布，否则，继续分析找出问题所在。下表是在随机轮转实验中，引入入 Fisher 和 Neyman 检验时的模拟验证。

	Fisher检验	Neyman检验
连续指标	 P值分布直方图	 P值分布直方图
比率指标	 P值分布直方图	 P值分布直方图

4. 总结与展望

在履约算法和业务同学每年分别运行数以万计的实验，测试内容涵盖履约业务的各方面，我们已经积累了如何进行更好的实验以及如何利用实验来做出更好的决策的知识。本文从实验通识的视角，介绍了履约在建设可信实验的实践，希望能对实验者有所帮助。由于履约的规模、影响力和多边业务模式的特色，履约的问题空间带来了独特挑战。溢出效应、小样本、策略公平性等综合因素制约着我们运行可信实验，我们在解决上述问题时，也沉淀了一系列实践，后续也会陆续推出相关实践文章。

基于不同的因素，如指标类型、样本量、样本分布特点等，可以应用不同的方法如线性模型、Delta 方法、Bootstrap 方法来计算 P 值和标准误差，并且在以城市、区域、站点为实验单位的实验中，可以自动选择不同的方法来调整标准误差，避免了由于数据聚类而导致的误报。这种灵活分析能力对策略的快速迭代非常重要，并且深受数据工程师的欢迎，这样他们就可以把精力和时间集中在实验的其他关键方面。

基于此，我们构建了统一的分析引擎，它标准化了实验核心框架，如普通随机对照实验、随机轮转实验、协变量自适应实验、准实验和观察性研究，以及一些其它的在产业界和学术界前沿的实验评估技术：如样本量预估、降方差、MED 分析、数据纠偏、轮转实验的携带效应估计等，以减少大家花在分析上的时间。未来，我们进一步开放该能

力，以服务于更多的用户。

5. 本文作者

王鹏、永斌、中锋等，均来自美团到家研发平台 – 履约平台技术部。

招聘信息

履约平台技术部 – 数据科学工程师

我们期望候选人可以建立统计模型、应用机器学习技术、分析配送业务数据，并使用这些建模技术构建相关指标来协助关键的业务决策。

1. 与算法或业务协作负责各类实验，确保实验的科学性与高效率；负责复杂实验的设计和评估，通过实验分析给业务决策提供推荐方案；负责科学实验评估平台的规划和演进。
2. 和业务紧密协同，深入理解业务和产品，将业务和产品的问题转化为数据和技术的问题，并且设计合理的解决方案，如主动通过数据探索和挖掘，帮助业务自动识别虚假申诉并驱动其完善管控规则；通过体系化的因果推断解释日常业务核心指标的波动变化，及时发现问题等。
3. 把握技术趋势加强行业对标调研适用于配送各业务的因果推断、统计推断，异常检测以及其它数据科学的方法，并把这些方法应用到实际业务问题。
4. 指导团队成员和数据分析师，帮助他们快速成长，为团队培养数据分析人才。

欢迎加入我们，简历请投递至：wangpeng47@meituan.com

基于 UI 交互意图理解的异常检测方法

作者：诗雨 张雨 永祥

1. 背景

近年来，随着美团多种业务线的扩充和迭代，UI 测试的任务愈发繁重。针对 UI 测试中人工成本过高的问题，美团到店测试团队开发了视觉自动化工具以进行 UI 界面的**静态回归**检查。然而，对于 **UI 交互功能逻辑**的检验仍强依赖于脚本测试，其无法满足对于进一步效率、覆盖面提升的强烈需求。主要难点体现在两方面：

- 前端技术栈多样，不同页面的实现方式各异，这导致不同页面中功能相似的 UI 模块的组件树差异很多，基于规则的测试脚本也就很难具备泛化能力，生产、维护的成本非常高。
- UI 样式繁多，同样的功能模块可能在视觉上有很大差异，这为基于 CV 方法实现自动化驱动带来了困难。

考虑上述两个难点，美团到店平台技术部 / 质量工程部与复旦大学计算机科学技术学院周扬帆教授团队展开了“基于 UI 交互理解的智能化异常检测方法”的科研合作，利用多模态模型对用户可见文本、视觉图像内容和 UI 组件树中的属性进行融合，实现了对于 UI 交互意图 [1] 的准确识别。该工作对于大前端 UI 的质量保障等多个领域都具有可借鉴的意义，介绍该工作的学术论文 [2] 已经被 ESEC/FSE 2023（软件领域 CCF A 类推荐会议）接收，并将于 12 月 6 日在其工业届轨（Industry track）公开发布、推介。

2. UI 交互介绍

2.1 UI 模块与交互意图

移动应用由“页面”组成，不同页面中的不同“模块”为用户提供着不同的功能。用户在浏览页面时，根据以往使用经验以及当前页面中的图像、文字、页面结构等信息，可快速理解页面当中不同【模块】所想要提供的【功能】，以及通过该功能用户能够达到的【目的】。这些被用户认为能够提供特定功能并达到预期目的的页面模块，我们将其命名为一个【交互意图簇】。

以下图中的页面为例，不同模块通常对应不同的交互意图类型划分。比如商品详情区域，我们可以得知此模块主要是向我们展示当前商品最主要的信息，起**展示**作用；而顾客信息区域，需要用户进行**点击或输入**个人信息，用以补全整个订单所需要的信息；同时页面当中也会存在各类功能按钮，通过按钮的 **位置** 、**文本信息、图标**等信息，用户也可以大致推断出操作后会得到怎样的结果。由此，我们可以将 UI 交互意图定义为「**用户通过当前 UI 展示推断出来的不同模块的概念及交互功能**」。

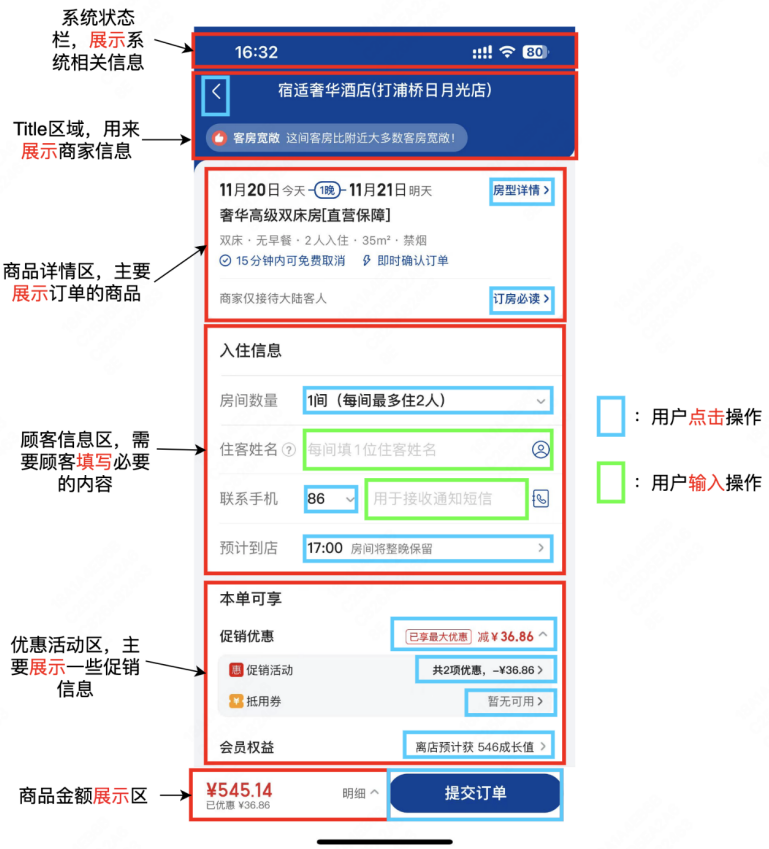


图 1　模块的概念及交互功能样例

2.2 当下痛点与启示

对于复杂的 UI 交互场景，如提交订单页，测试人员需要对不同模块制定较复杂的测试流程、测试规则，同时编写及维护复杂的自动化测试逻辑。以美团内的 App 测试场景为例，许多不同的页面有着相似的功能模块，这些功能模块尽管表象不同，但对于一般

用户来说，交互意图明确且相似，没有理解困难。如：



图 2　相似的功能模块

2.3 研究目标

本课题期望**结合多种机器学习方法，通过机器获取与人工认知一致的“交互意图”，从而利用该信息模拟测试人员对客户端产品进行“理解 – 操作 – 检查”的测试验证流程**。如人工操作一般，我们希望该能力能够以一个与一般用户类似的逻辑来操作、检查相似的功能，同时兼容不同的技术栈、App、业务领域，无需特定适配。就像一个用户在美团上能够订酒店，在没有使用过的点评或者美团小程序上也同样能完成预订酒店的流程。

从能力目标视角来看，UI 交互意图识别的定位是完成一般用户的交互概念到页面实体的映射。由人直接进该映射的准确率最高、泛化性最好，典型的场景就是手工测试，即人观测页面后操作、检查。人能在不同的设计、程序实现形式下实现下找到目标操作实体（例如各种各样的提交按钮、商品卡片）。当前的自动化脚本测试提高了效率，但由于映射的泛化性较差，往往需为每个页面做单独的适配。

此外，业内尝试了诸如 CV 页面目标检测等方法，但在鲁棒性、泛化性、使用成本等方面上仍不太令人满意。本研究旨在利用深度学习和多模态信息，通过少量标注数据，尽可能提升交互意图识别的映射能力，使其接近人的识别、认知水平。

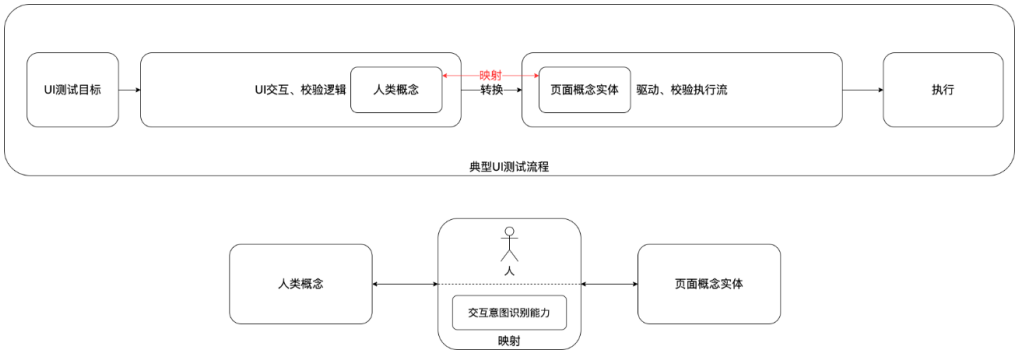


图 3　UI 交互意图理解的能力目标

2.4 效果预期

本研究**提供一种 UI 交互意图理解的通用能力，能够在测试核心流程“理解 – 操作 – 检查”各个环节应用**。

- 识别页面模块交互意图：通过页面 UI 交互意图识别来模拟测试人员的认知
- 测试行为的注入：利用 UI 交互意图识别结果信息，将操作逻辑程序化
- 测试结果检查：利用 UI 交互意图识别结果信息进行页面状态通用化校验



图 4　核心流程“理解 – 操作 – 检查”



图 5　预期效果示例

3. 架构设计

3.1 技术思路

考虑到 UI 交互意图理解是一种页面理解的通用能力，需要结合业务场景产生实际效果，我们决定首先将其应用于**智能化 UI 交互**，探索交互意图理解的能力范畴以及落地效果验证。后续会将该能力扩展到智能化测试逻辑检查、智能化遍历测试、测试知识标准化管理以及推荐等其它大前端测试领域应用。

为了验证技术方向的可行性，本课题先限定在某个垂直业务领域（订单 / 表单）进行探索，确认实际使用效果，再将方法推广泛化到其他领域。

具体来说，本项目的技术方案分为两个部分：

- **UI 交互意图理解：**基于深度学习方法对交易流程中表单 / 订单场景进行目标 UI 交互意图簇识别划分。
- **智能化测试用例驱动：**定义测试用例目标，基于表单 / 订单等场景中的 UI 交互意图簇编写交互逻辑，在跨 App、跨技术栈、跨业务的场景下尝试复用执行。

3.2 当前进展与效果 Demo

本项目目前已经实现了一套通用 UI 交互意图理解方法，利用 UI 交互意图在一些场景

下编写的智能化测试用例可以在不同 UI 页面、不同技术栈，甚至不同 App 之间复用。下面是一个使用 UI 交互意图编写的“**下单首个商品**”测试用例的交互意图和其泛化能力的效果展示：

交互流程：识别第一个商品、点击购买进入提交订单页、填写顾客信息、提交订单。

App 效果展示：



【视频】美团 App 下单购买列表内首个商品

3.3 实现难点



图 6　测试过程中存在的挑战

如何让机器学习到一般用户的认知概念，自动分析获取到预先定义的 UI 交互意图是本课题中最大的难点。

4. 实现方式探索

针对上述难点，本课题从真实的业务需求出发，首先梳理了需要识别的交互意图类别。随后，对交互意图类别进行了分析，先后进行了多种方法的尝试，通过定量实验对不同设计进行了效果比对，最终选取了**先分类，再聚类**的落地方案：先以渲染树元素为最小单位进行交互意图类型的分类，然后在不同的交互意图维度进行元素聚类，生成对应的交互意图簇。

4.1 交互意图识别需求

深入挖掘业务需求后，我们发现：UI 交互意图并不是一维的，在不同场景、不同需求下会有不同的分类标准。具体来说，如果关注所属业务类别，可以将交互意图簇分为：商品信息、评价和发票等；当关注用户的操作方式时，又需要将可交互的组件分为：点击、键入、长按三类。例如，对于“点击进入第一个商品详情页”这样一个交互意图：模型需要从**业务层**找到“商品信息”，在商品信息簇中找到**操作层**的可“点击”的 UI 组件（“商品信息”和“点击”的交集），然后执行点击操作。

此外，由于本工作的初步实验场景为具有大量计算逻辑和信息输入的表单页，因此我们又增加了**计算层**和**表单层**两个特有的维度。例如，对于“购买最便宜的商品”这样的交互意图，其细分为“找出最便宜的商品”和“订单填写”两个串行子意图。具体来说，模型需要首先在商品列表页找出业务层“商品信息”和计算层“金额统计信息”的交集并排序，随后点击最便宜的商品进入提交订单页。在提交订单页中，模型需要在业务层的“顾客信息”和表单层的“信息输入”中找出共有的元素，并根据这些元素生成对应的文本输入信息，从而完成“订单填写”的子意图。

由此，我们利用上述四个维度将新的分类标准确定为 16 个不互斥的类别。

维度	类别
1. 业务层	顾客信息、商品信息、优惠、评价、备注、发票、提交按钮
2. 表单层	功能按钮、信息输入、标题、说明
3. 计算层	数量、金额统计信息
4. 操作层	点击、键入、长按

四个维度的预期分类结果分别如下图所示：



图 7　多层次预期识别结果

4.2 模型的输入

为了实现 UI 交互意图理解这个目标，我们推测，与一般用户的理解方式类似，基于多种信息的综合考虑能够提升整体效果，因此选择了三种模态的页面信息：**图像信息**（来源于页面截图）、**渲染树信息**和**文本信息**。

例如，针对元素【“普通支付”按钮】，可获得的关键信息有以下三种：

	图像信息	渲染树信息	文本信息
【“普通支付”按钮】			text - “普通支付”

图 8　多模态信息示例

4.3 双阶段 UI 交互意图理解

分析输入数据可知：三种信息输入中只有“渲染树”带有明确的边界，但其与“交互意图”概念在粒度上有显著差异。因此，本研究考虑采用**先分类，再聚类**的思路：先以渲染树元素为最小单位进行交互意图类型的分类，然后在不同的交互意图维度进行元素聚类，生成对应的交互意图簇。

具体来说：

- 在分类时，利用自注意力机制进行特征提取，实现在判断当前元素类别时会参考其它元素信息的目标。
- 在元素聚类的过程中，利用有监督的聚类方法将分类后的渲染树元素在不同交互意图维度进行聚合，得到簇划分结果。

4.3.1 UI 组件分类模型

由于渲染树反映的是最细粒度的 UI 组件，因此对渲染树中组件进行分类的最大难点是信息缺失：订单页中的数字有可能是金额、商品数量、顾客人数，此类情况仅依据当前渲染树节点无法区分。因此，本研究借鉴了 NLP 领域的经验，利用自注意力机制进行特征提取，实现在判断当前元素类别时会参考其它元素的信息的目标。

分类模型结构如下图所示，我们利用 Vision Transformer 预训练模型提取图像特征，

使用中文 bert 预训练模型提取文本特征，同时把渲染树元素属性进行特征提取后输入模型，综合判断元素类别：

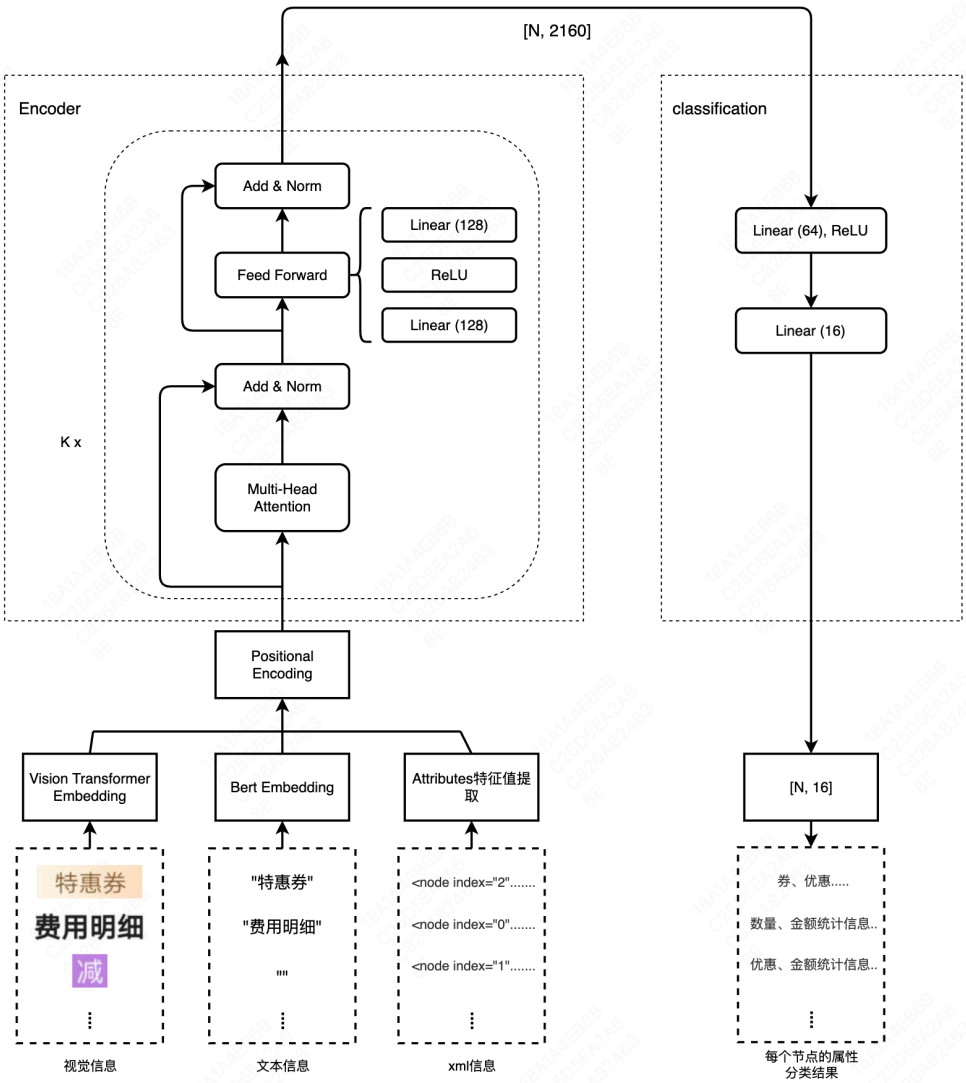


图 9　模型结构

为了探究三种关键信息（渲染树、视觉图像信息、用户可见文本）的有效性以及三者之间的关系（是否相互补充），我们将不同关键信息作为模型的输入类型进行了消融实验，训练了 7 种不同的自注意力分类模型。此外，考虑到在 UI 领域很多实践使用 CV 目标检测能力实现类似工作，为了对比此类目标检测模型和自注意力模型在当前问题上的效果差异，本研究以 YOLOv7 模型为代表，定量评估了其在 UI 组件分类上的效果。

实验时，本研究从美团 App 上的四种业务线（酒店、KTV、密室、门票）中随机截取了 158 个提交订单页面，进行人工标注后，将其中的 123 个页面作为训练集，其余为测试集。在测试集上，各个维度的 F1 Score[3] 如下：

模型	平均F1 Score	顾客信息	商品信息	优惠	提交	信息输入	数量	金额统计
YOLOv7	0.551	0.477	0.329	0.570	0.614	0.526	0.853	0.547
仅渲染树	0.779	0.816	0.794	0.700	0.863	0.784	0.785	0.686
仅图像	0.720	0.729	0.738	0.649	0.900	0.696	0.681	0.646
仅文本	0.780	0.905	0.798	0.733	0.734	0.789	0.737	0.823
图像 + 渲染树	0.813	0.833	0.783	0.740	0.926	0.795	0.839	0.754
文本 + 渲染树	0.837	0.920	0.813	0.758	0.865	0.797	0.922	0.815
图像 + 文本	0.830	0.916	0.788	0.788	0.924	0.776	0.826	0.829
多模态	0.861	0.894	0.823	0.742	0.996	0.803	0.869	0.850

从上表可以看出，**多模态自注意力深度学习的 UI 交互意图理解方案在相同数据集下具有最好的表现**。分析原因主要有两个：首先，随着模态的增多模型的效果会变好，可见三种模态的信息互为补充，让模型能够通过多个维度更准确的进行拟合；此外，自注意力机制的引入使得节点的分类能够考虑到其周围的相关信息，提升了特征提取的效果，让 UI 组件分类更准确。因此，我们的后续研究基于此**多模态自注意力模型**展开。

该多模态模型的 UI 组件的多维度分类结果示例如下：



图 10　不同页面下的分类效果（不同颜色框代表不同类别）

4.3.2 交互意图簇生成：UI 组件聚类

当前多模态多分类模型针对的识别对象是一个个渲染树节点。一般来说多个渲染树节点才能组合成一个完整的交互意图簇，所以我们考虑将属于同一个意图的节点聚类在一起，这样就能够给下游任务提供更多可用信息。

我们首先尝试了基于规则的无监督聚类方式：将一个表单页上被分类模型判为同类型的连续节点聚为一个交互意图簇。但由于其在处理连续但独立的同类交互意图簇时效果很差，并不适用于当下复杂场景。

深入分析可知，聚类任务有两个难点：

- 情形 1：如果渲染树节点不连续但是属于同一个簇，仍希望对其成功聚类。
- 情形 2：连续的渲染树节点可能被分类模型判定为同一个交互意图类别，但希望对齐一般用户的理解将其聚类为多个独立交互意图簇。（如：连续的几个表单填写框我们希望能够将其分割开，如下图中出现的三个连续的【信息输入】交互意图簇）。



图 11　三个连续的信息输入交互意图簇

在聚类的实现方式上，我们考察了多种常用聚类手段：

1. 最简单的规则聚类（连续同标签的渲染树节点为同簇）并不能处理前述连续多个同类簇情形。
2. 以 k-means[4] 为代表的无监督聚类方法在本研究所涉及的多维度聚类问题上也表现不佳，我们对其超参数进行了广范尝试也无法得到理想的聚类效果。

经过广泛尝试，我们最终选取了一种有监督聚类方案：**每个节点依次与其它节点计算是否属于同一个簇，将被判断属于同一个簇的聚合起来**。通过分类模型判断是否属于同一个簇，而模型的 ground truth（真值）是我们人工标注出的每个类别页面中的所有渲染树元素簇。聚类模型结构与训练流程如下：

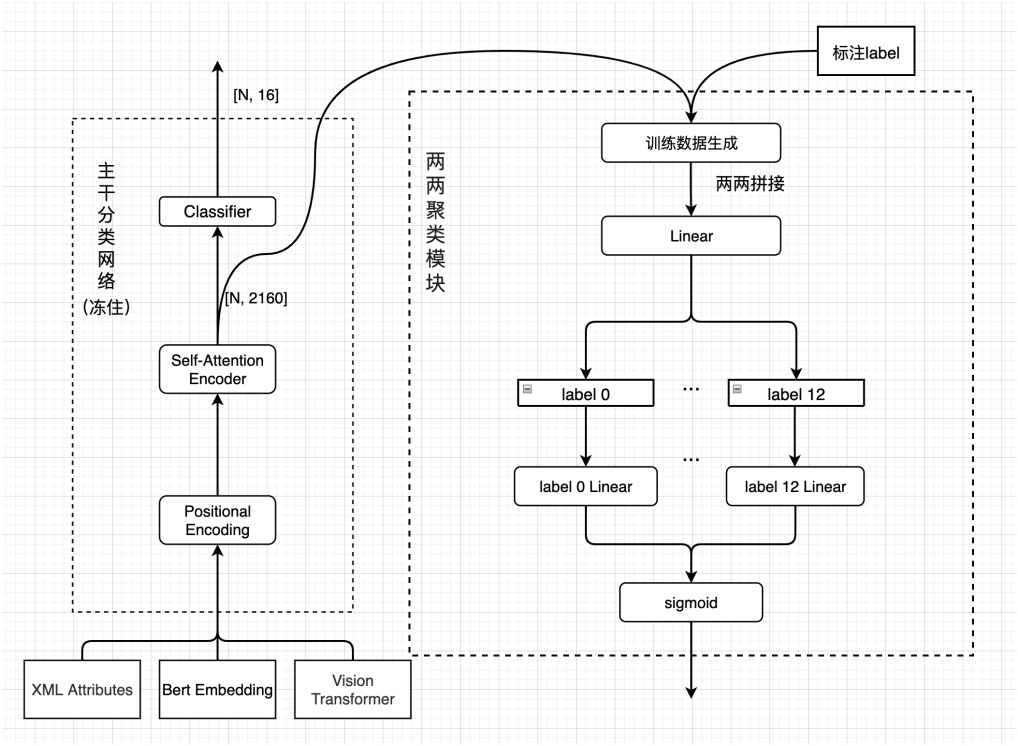


图 12　聚类模型结构

我们采用与分类模型相同的页面标注数据生成训练集和测试集。训练时，我们首先为每个 UI 界面上生成所有可能的两两组合，其中在任意类别属于同一簇的组合是模型输入正例，其余为反例。预测时，我们将分类模型的结果送入聚类模型，由聚类模型输出最终的交互意图簇。

在 UI 组件聚类的评测指标上，我们采用标准聚类量化评估参数兰德系数 [5] 衡量聚类模型的效果。William M. Rand 通过将聚类问题转换为任意两个元素的组合（N(N−1)个）是否在同一簇的决策问题，由此定义了在聚类问题上的混淆矩阵（TP，TN，FP，FN）。因此本研究用 Precision 来衡量聚类结果的准确性，用 Recall 表示聚类结果的全面性，而两者的 F1 Score 就是兰德系数。

由于本研究采用的是多层次平行聚类算法，所以兰德系数的数值整体偏低（如下表所示）。但从聚类效果图（图 13）可知当前有监督聚类模型对交互簇有良好的聚类效果。

	顾客信息	商品信息	优惠	备注	提交	功能按钮	信息输入	标题	说明	数量	金额统计
Precision	1.000	0.964	0.954	1.000	1.000	1.000	0.997	1.000	0.924	1.000	0.866
Recall	0.871	0.965	0.402	1.000	0.711	0.873	0.517	0.983	0.624	0.734	0.296
F1	0.931	0.964	0.565	1.000	0.831	0.932	0.681	0.991	0.745	0.846	0.441



图 13　选用的有监督聚类效果示例（不同颜色框代表不同类别）

4.4 实验结论

综上，我们得到了以下结论：

本研究提出的基于多模态自注意力深度学习的 UI 交互意图理解方案在准确性、泛化性上具有一定优势，且其对数据标注和训练轻量化的需求贴合业界的真实测试场景。

5. 实际落地探索

基于 UI 交互意图理解的智能化测试用例驱动

UI 交互意图识别模型在订单页领域已经具备了一定的交互意图簇识别能力，我们期望利用 UI 交互意图识别模型进行智能化测试用例驱动：在交互意图层面进行大前端测试用例的编写，希望测试用例在不需要任何适配的情况下实现跨端、跨 App、跨技术栈的执行。 我们在美团 App 的酒店详情页场景的安卓端利用交互意图簇识别能力完成以下测试用例的驱动编写：

1. **下单首个产品**
2. **下单最便宜的产品**

下面是测试用例伪代码以及部分输入集合定义：

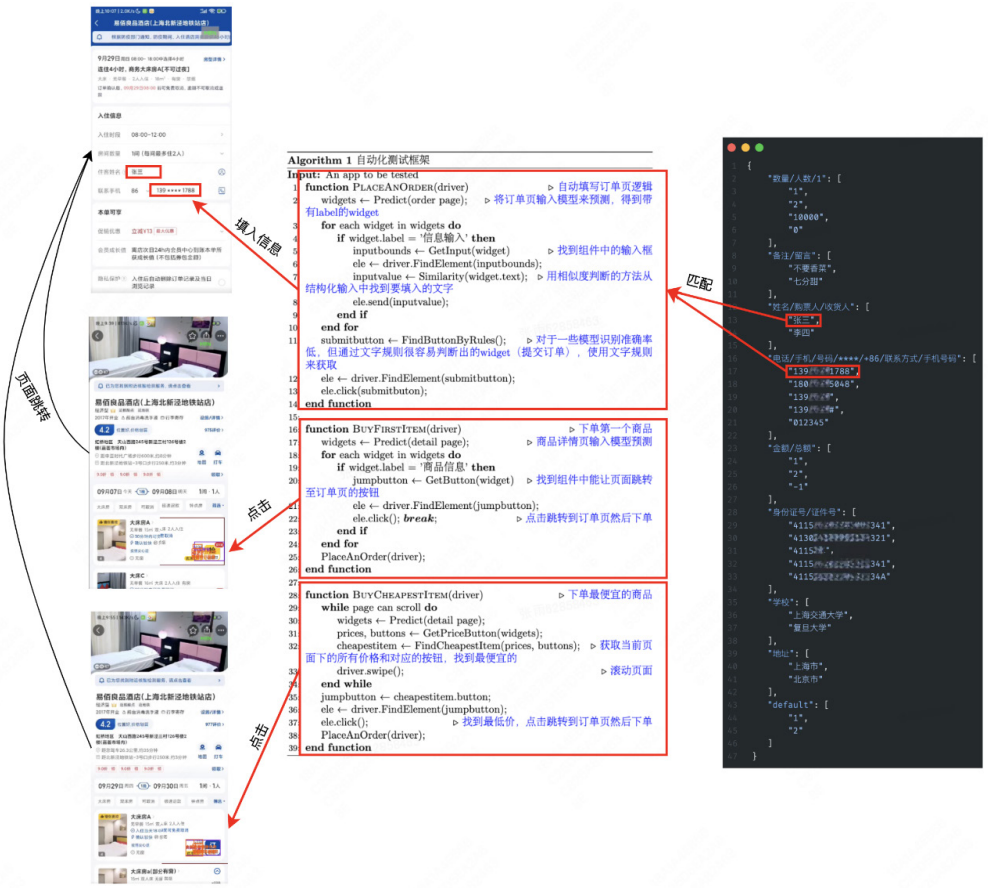


图 14　测试用例伪代码与执行流程

图中显示了选择“首个酒店”与“最便宜酒店”的伪代码流程。

- 首先在酒店详情页，我们会在 BuyFirstItem 与 BuyCheapestItem 这两个函数中实现主要逻辑 。其中在 BuyFirstItem 会寻找到首个被模型识别为“商品信息”的交互意图，并从中找到在这个交互意图中的“购买按钮”意图，进行点击后跳转到填单页。在 BuyCheapestItem 中我们会获取页面上所有的“商品信息”的交互意图，并从每个商品信息意图中识别“价格信息”交互意图，得到每个商品的价格进行比较，找到最便宜的商品后点击其“购买按钮”意图进入填单页。
- 进入填单页后，通过模型识别“信息填写”交互意图。如图中所示，首先识别到两个“信息填写”意图，通过其提示文字【 住客姓名、联系手机 】匹配输入集合中的 姓名、电话，从而选择出信息填写到对应的输入框中。在此之后，利用模型识别页面中的“提交订单”交互意图，点击进行订单提交从而完成整个下单流程。



【视频】美团 App 下单购买列表内首个商品



【视频】美团 App 下单购买列表内最便宜的商品

此外，我们在训练集以外的五种 App 上定量研究了智能化测试用例的可用性和泛化性。100 个不同的页面中，基于 UI 交互意图理解的智能化测试用例在 89 个页面正确执行通过。该实验证明：基于 UI 交互意图理解的智能化测试用例具备良好的鲁棒性和泛化性。

目前，我们正在推进 UI 交互意图在实际自动化测试用例编写中的落地工作，即用 UI 交互意图代替基于规则的测试驱动脚本。由于业内的测试场景往往涉及不同技术栈、不同业务之间的大量相似的页面，这种泛化能力强的测试用例可以在相似页面复用，因此可以减少开发成本。此外，与现有的基于规则的测试脚本不同，该方法对 UI 页面的小规模变化不敏感，不会出现需要频繁维护 Selector[6] 的情况，可一定程度上减少自动化 Case 维护所耗费的精力。

未来，我们将通过收集更为广泛的 UI 数据来训练一个通用的 UI 交互意图理解模型以覆盖常见页面中的 UI 交互意图识别，业务质量保障人员可以直接利用这种通用的识别能力开发泛化性、鲁棒性更好的智能化测试用例。对于那些暂时处于模型能力范围之外的页面或者新上线的业务，我们将提供模型的微调接口，经少量的标注数据微调即可使其在相关页面展现出识别效果。

6. 总结

本文介绍了利用页面多模态信息在 UI 测试领域的探索与实践经验。针对意图信息识别问题，我们利用图像 + 文本 + 渲染布局属性信息探索出了一种交互意图簇识别模型，验证了基于自注意力的多模态方向可行性。此模型可以识别出渲染树元素多维度的意图属性信息，同时利用聚类算法将节点聚成交互意图簇，可以为后续的任务提供结构化决策信息。在标注数据较少的情况下仍体现了较好的准确率以及泛化能力。后续计划通过扩大数据集、加强预训练等方式继续提升模型识别的精度。

回顾整个 UI 交互意图理解探索历程，先后经历了“无监督 / 无类别的区域划分”，“有监督针对 UI 节点分类”，“分类后聚类”，“利用识别结果进行测试用例编写、执行”四个阶段。目前在 UI 交互意图提取上我们已探索出了较为合适的方案，正进行实际业务落地，让 UI 交互意图识别能力融入当前大前端测试能力，在智能测试用例驱动、智能检查等方向上取得实际应用收益。

7. 展望

下面是几个基于 UI 交互意图理解能力开展的业务落地工作。

1. 智能探索性测试

当前 App 功能复杂，有大量可以操作的组件，无意识的探索效率太低，期望利用意图识别结果，对当前测试场景的一些通用可操作组件执行有意义操作的自动化测试，并进行逻辑问题校验。

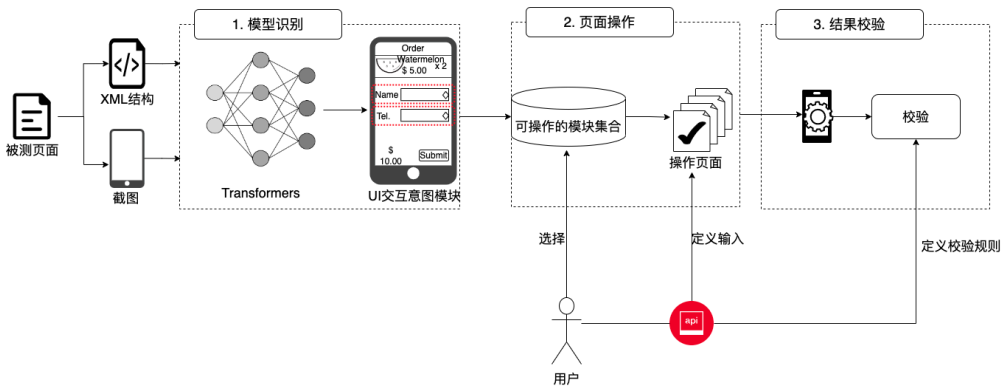


图 15　探索性测试大体流程

2. 跨分辨率 UI Diff 及归因

不同分辨率 / 设备下布局存在差异，像素级比对无法识别不同分辨率下的 UI Diff。使用交互意图簇 Diff 可以大大削弱像素位置的差异造成的干扰，支持跨分辨率的比较，凸显 Diff 所需要关注的文本 / 图像变化，并可利用意图信息对结果进行结构化归因。

3. 节点匹配选择

利用意图识别预训练模型，支持节点匹配任务，实现泛化性较强的跨分辨率、跨技术栈、跨 App 的节点查找能力，与现有的基于 XPath、Selector 等的线性条件节点选择模式形成互补。

在中长期来看，我们期望将 UI 交互意图识别作为大前端结构化信息提取的通用能力，在不同的业务领域进行如智能测试 bot、终端测试标准化知识组织与覆盖率评估、智能辅助测试用例编写与生成等方向上持续探索、落地。

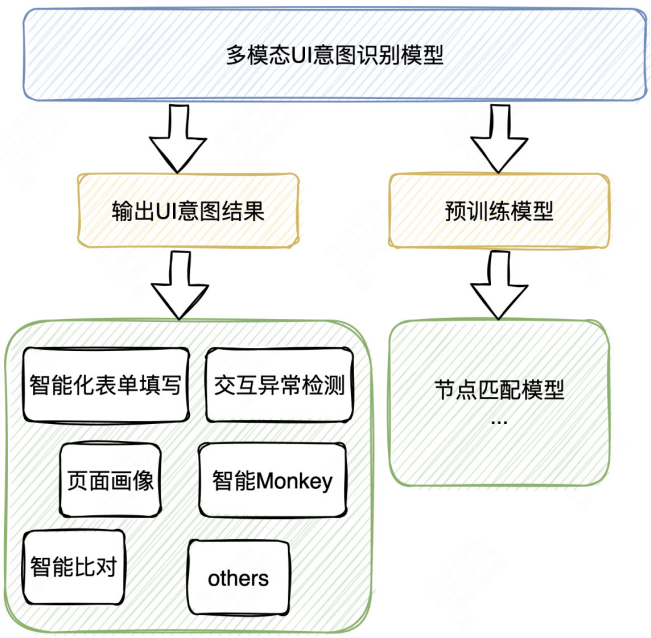


图 16　相关下游任务

附录

大模型时代下 UI 交互意图理解能力的意义

目前业界内大模型主要存在两种类型：大语言模型 [7]（LLM：仅支持文本模态输入输出）以及多模态大语言模型 [8]（MLLM：可以同时处理多种模态信息）。目前大语言模型具备比较良好的通用化逻辑理解能力，而多模态大语言模型能够同时基于文本、图像等模态信息完成理解、判断，但整体的逻辑能力水平相比大语言模型有一定差距，在一些多模态任务上判断、分析的精度尚不够令人满意。

基于这两种大模型，在实际任务解决上有两种相对应的主要模式：**【LLM as Controller】**与**【MLLM cognize Everything】**，UI 交互意图理解作为一种垂直领域能力，在两种模式中都具有相应的应用潜力。

LLM as Controller

该模式的核心思路是将垂直能力作为工具，LLM 作为总控，利用其逻辑推理能力，通过自然语言理解目标，然后进行决策，编排、调用工具，完成任务。这种范式下的典型实例有 HuggingGPT[9] 等。在这种范式下，LLM 能够与垂直能力优势互补，更好的完成多领域任务。

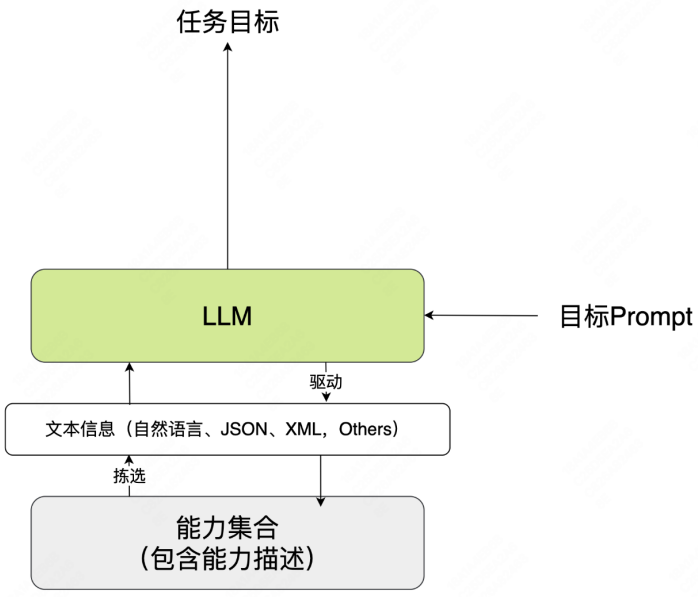


图 17　相关下游任务

以 HuggingGPT 项目为例，其主要思想是将 LLM 作为总控，将 HuggingFace 平台上众多的垂直能力模型作为工具集，用户可提出需求，LLM 根据需求调用垂直能力。最后 LLM 根据垂直能力返回的结果生成满足用户需求的多模态内容。

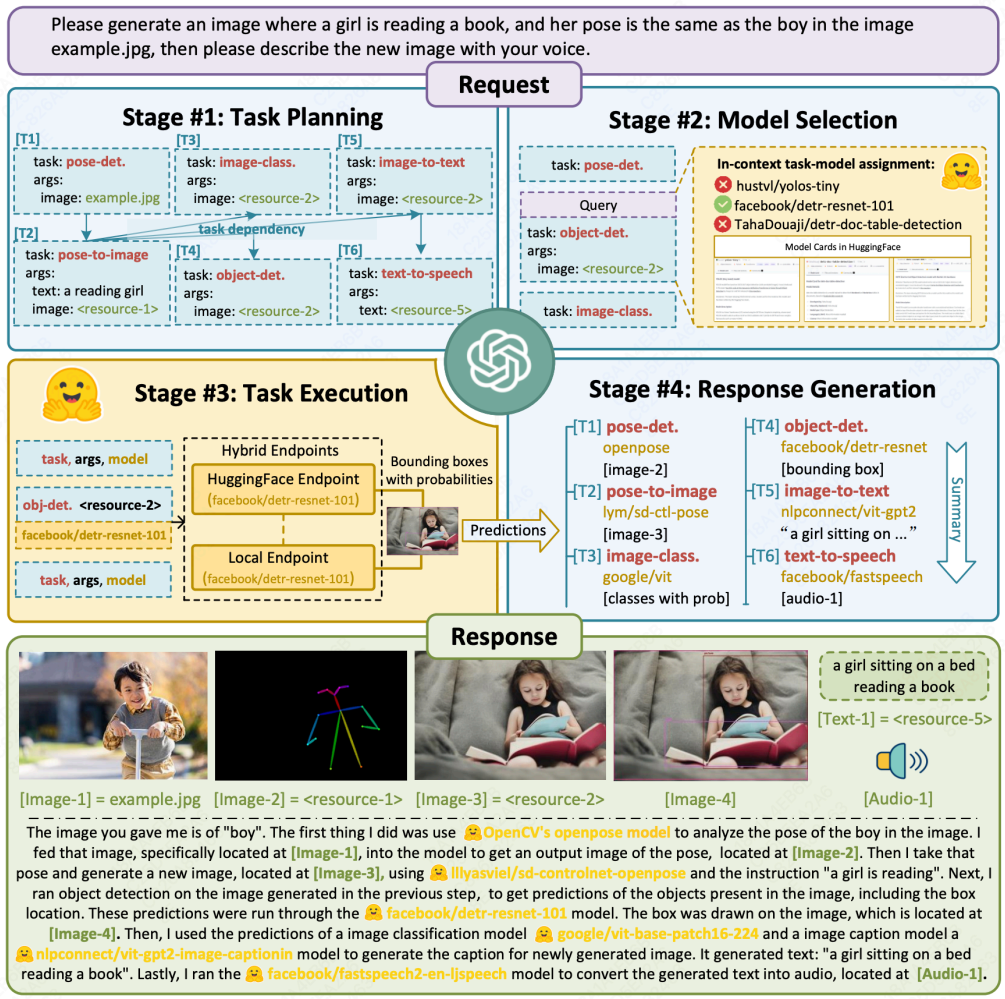


图 18　HuggingGPT 工作流程

可以看到在这个模式下，与其它的垂直工具能力类似，**UI 交互意图理解能力可以作为工具能力供 LLM 调用**，更好的完成 UI 交互相关的任务。

MLLM Cognize Everything

多模态大语言模型出现了之后，让我们看到了多模态任务通用化解决方案的曙光。具体

到 UI 交互意图识别任务中，我们尝试使用多种 MLLM 直接进行 UI 交互意图识别，总体来看 MLLM 已经具备不错的识别能力，但是在具体的坐标、内容分析方面上仍有偏差。UI 交互意图识别模型可以通过以下两种方式帮助 MLLM 在意图识别任务上进行性能提升：

- 将 UI 交互意图识别模型作为页面多模态信息 Encoder，通过微调的方式提升意图识别任务中的准确率。

这里以 MiniGPT 为例，介绍 Encoder 模式。

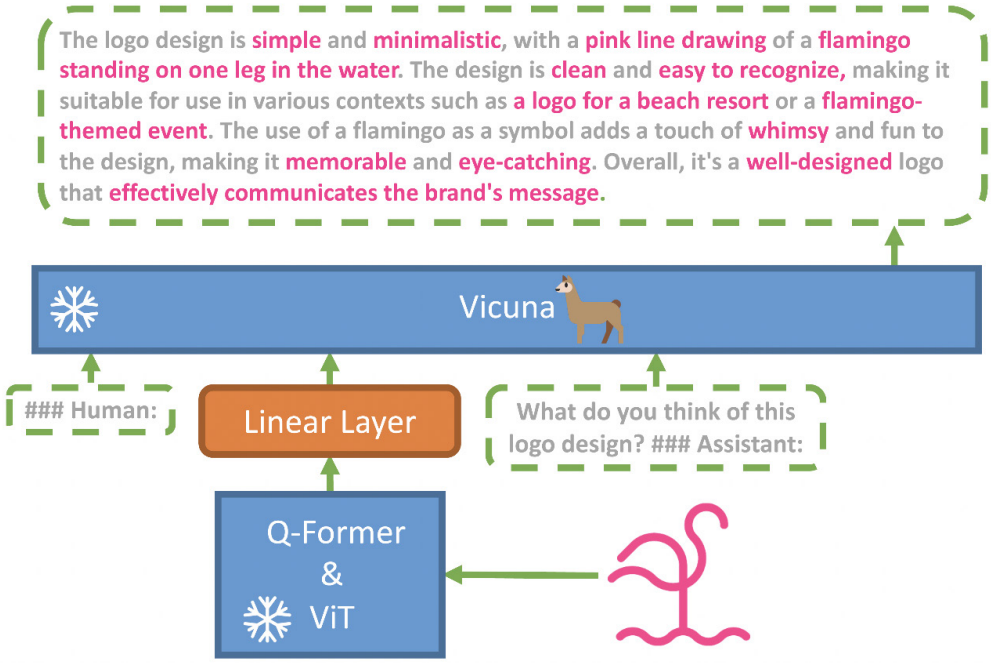


图 19　MiniGPT 模型结构

由上图可知，多模态大语言模型中一般由每个模态对应的模块来进行模态信息处理，如上图中 VIT[10]&Q-Former[11] 为图像模态处理部分，Vicuna[12] 是一种开源的 LLM。UI 交互意图理解模型可以替换图中 VIT&Q-Former 的位置，**作为交互意图信息的处理预训练 Encoder 与 LLM 结合进行多模态整合训练**，产出页面分析来辅助多模态大语言模型在大前端质量保障中的应用。

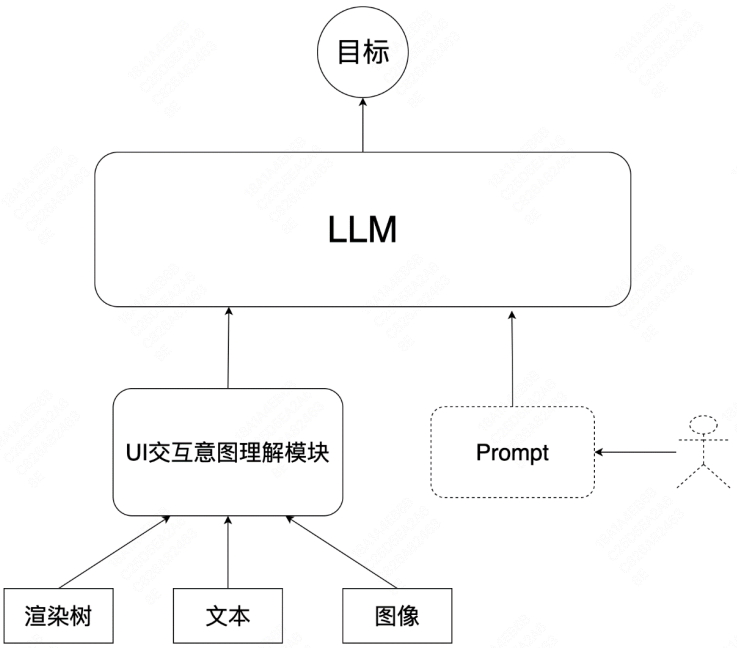


图 20　基于 UI 意图理解能力的多模态大语言模型结构

- 将 UI 交互意图识别模型做为信息提取工具，将识别出的结构化信息加入 Prompt 中，帮助 MLLM 更精确的进行意图识别。

总体而言，UI 交互意图识别是一种**简单轻量但效果不错的垂直领域能力**，只需简单少量的训练数据，即可实现在诸如跨 App、跨技术栈、跨业务等复杂场景下准确识别多种交互意图的能力。大模型领域日新月异，我们也将持续探索 UI 交互意图识别能力与该领域技术的结合方式，发挥其最佳效果。

本文作者

诗雨，张雨，永祥等，均来自美团到店事业群 / 平台技术部 / 质量工程部。

参考文献

[1] UI 交互意图：用户通过当前 UI 展示推断出来的不同模块的概念及交互功能。
[2] 学术产出：Appaction: Automatic GUI Interaction for Mobile Apps via Holistic Widget Perception
[3] F1 Score：精确率（Precision）和召回率（Recall）的调和平均值。
[4] k-means：一种最初来自信号处理的矢量量化方法，旨在通过最小化簇内方差将 n 个观测值划分为 k 个簇 .
[5] 兰德系数：Rand W M. Objective criteria for the evaluation of clustering methods[J]. Journal

of the American Statistical association, 1971, 66(336): 846-850.
[6] Selector：此处指在基于规则的测试脚本中选取 / 定位 UI 组件的工具。
[7] 大语言模型（LLM）：依据维基百科的定义，大语言模型是一种能够理解和生成自然语言的人工神经网络，它们利用了海量的文本数据和复杂的神经网络结构，可以通过预测下一个单词或使用提示工程来完成各种语言任务，但也可能存在不准确性和偏见。从定义中可知，狭义的大语言模型的输入和输出仅支持文本模态的信息。ChatGPT-3.5、Meta 的 LLaMa，以及 BLOOM、Anthropic 的 Claude 等都属于该类型。
[8] 多模态大语言模型（MLLM）：多模态大语言模型（英语：multimodal large language model，MLLM）是一种能够处理多种类型的数据，如文本、图像、音频、视频等，以实现通用的语言理解和生成的人工智能模型。MLLM 利用了海量的多模态数据和复杂的神经网络结构，可以在不同的模态之间进行转换、融合和推理，以完成各种语言任务，如文本分类、问答、对话、图像描述、视频摘要等。从定义可知，多模态大语言模型可输入或输出多种模态的信息。GPT-4V、LLaVA、Minigpt-4 等属于该类型。
[9] HuggingGPT：microsoft/JARVIS: JARVIS, a system to connect LLMs with ML community. Paper: https://arxiv.org/pdf/2303.17580.pdf (github.com)
[10] VIT（Vision Transformer）：VIT 是一种基于 Transformer 的模型，最初提出用于图像分类任务。与传统的卷积神经网络（CNN）不同，VIT 将图像看作是一个序列（或一组图像块），并使用 Transformer 的注意力机制来学习图像的全局特征表示。VIT 将输入图像划分为一系列的图像块（patches），然后将这些图像块转换为向量形式，并将其输入 Transformer 编码器中。通过自注意力机制和多层感知机，VIT 能够学习到图像的全局特征表示，然后通过一个分类头进行分类。
[11] Q-Former（Quasi-Recurrent Transformer）：Q-Former 是一种融合了 CNN 和 Transformer 的模型，用于处理序列建模任务，例如语言建模和机器翻译。Q-Former 将 CNN 模块和 Transformer 模块结合在一起，以在序列中捕捉局部特征和全局依赖关系。在 Q-Former 中，CNN 模块用于提取输入序列的局部特征表示，然后 Transformer 模块用于学习全局依赖关系和序列建模。相比于传统的 Transformer 模型，Q-Former 在一些序列任务中可以获得更好的性能，并且具有更高的计算效率。
[12] Vicuna：lm-sys/FastChat: An open platform for training, serving, and evaluating large language models. Release repo for Vicuna and FastChat-T5. (github.com)，具体的多模态训练流程可以参考该项目论文

招聘信息

美团到店平台技术部 – 质量工程部是到店事业群的技术质量保障团队，负责到店综合、到店餐饮、酒旅业务以及到店多个平台（业务平台、技术平台以及数据智能领域）的质量保障，通过建设基础工程能力，从研发全过程建设质量，保障到店业务与技术系统规模化快速迭代。我们注重技术创新，不断地探索技术边界，用技术驱动业务，团队技术氛围浓厚。目前正在招聘：

- 测试开发工程师 – 服务端 / 视频领域 / 数据智能 / 移动端
- 开发工程师 – 业务稳定性保障

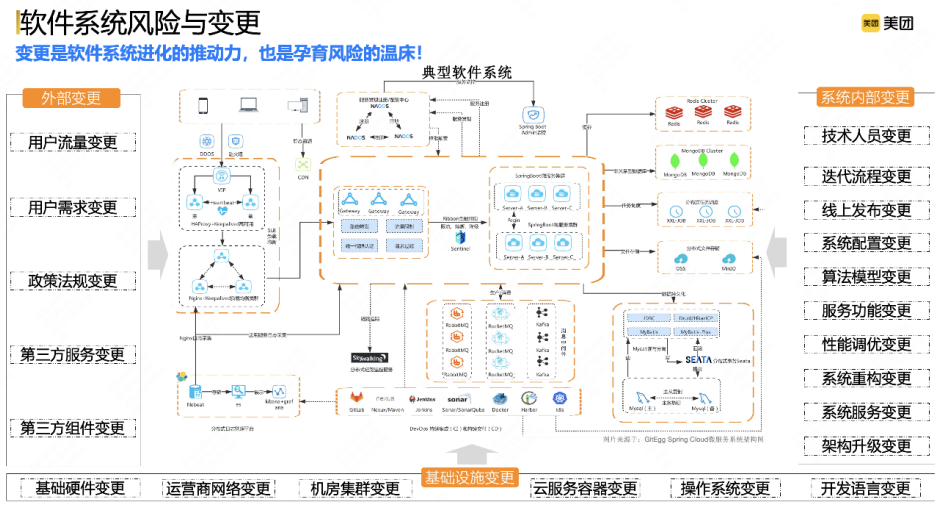
欢迎感兴趣的同学发简历到：xuwen19@meituan.com，邮件标题格式：姓名 – 岗位名称 – 社招。

代码变更风险可视化系统建设与实践

作者：桂来

1. 软件系统风险与变更

变更是软件系统进化的推动力，同时也是孕育风险的温床。如果一个系统没有了相应的迭代和变更，那这个系统就会逐渐失去了活性和价值。不过，随着系统进行了变更迭代，软件风险也会慢慢衍生，而规避变更引发的软件风险在质量保障领域是一个较大的挑战。通过对下面典型软件系统架构图分析，我们可提炼出 3 大类变更维度：



- **基础设施变更**：主要包括基础硬件变更、运营商网络变更、云服务容器变更、开发语言变更、操作系统变更以及机房集群的变更，这些基础设施迭代极大提升了系统底层的服务能力，一旦变更引发系统风险，其影响面通常也比较大。
- **系统外部变更**：比如用户流量突增、用户需求变化以及相关三方服务及三方组件变更，这些帮助系统不断衍生出新的迭代能力，同时也增加了系统稳定性风险的发生。
- **系统内部变更**：比如技术人员迭代、新功能发布以及系统整体架构的升级等，这是驱动系统软件进化的核心变更因子，也是最频繁的变更风险发生地。

在这里，我们先列举了一些比较常见的、因变更风险所引发的典型线上事故：

- 外部变更所引发的线上问题，某地的光缆被挖断导致整个服务有很大的影响。
- 代码变更典型问题，谷歌 Gmail 系统在发布新功能时产生的副作用而引发的功能上问题。
- 代码变更典型问题，Knight 公司在升级一段很老的代码时引发的异常逻辑功能发生。
- 配置变更引发的问题，所引发的“薅羊毛”事件。
- 人员操作变更，研发误操作引发的整个核心数据删除；

软件事故感知变更风险



图片来自网络

可以看到，在实际的工作中，由变更所引发的风险，对业务的冲击非常大。结合美团亿级流量的到家业务形态看，系统变更引发风险可能性进一步放大，变更风险的“蝴蝶效应”更加凸显，某个单点问题都有可能给整个到家核心业务带来极大的影响。

- 第一，从到家业务接入方看，美团内部业务包括外卖、闪购、医药等等，外部有众多的企业客户。
- 第二，系统参与相关方较多，包括 C 端用户、商家、配送骑手及各个平台。
- 第三，业务基于微服务架构模式，各个业务间调用关系复杂，核心链路非常长。另外，业务强依赖配置，一旦某个环节发生变更问题，相关方都会受到影响。

所以对研发与测试来说，洞察与规避变更引入的质量风险变得至关重要。

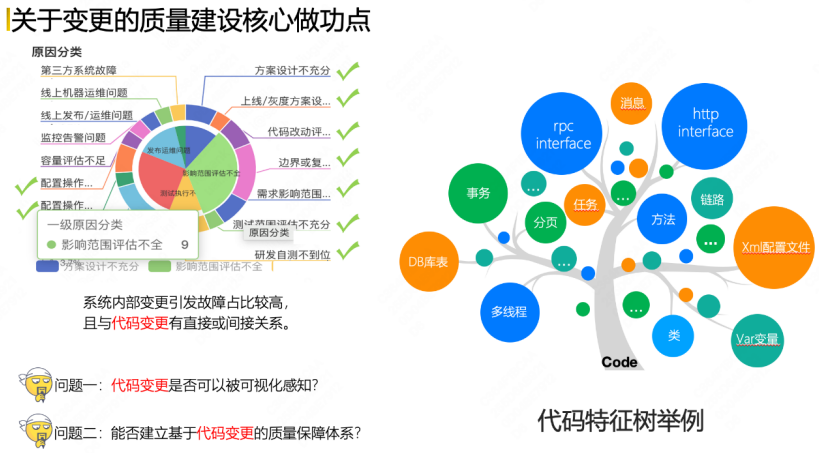


那么，关于变更风险，质量建设核心做功点在哪里？我们对历史线上问题分析发现，系统内部变更引发故障的占比较高，且变更与代码变更有直接或间接关系。因此，我们开始围绕**代码变更**这个核心变更因子，构建了质量建设的做功点。

随后，我们思考了两个关键问题：

1. 代码变更风险是否可被可视化，提升测试和研发感知能力。
2. 围绕代码变更风险，是否能够构建一套质量保障防御体系。

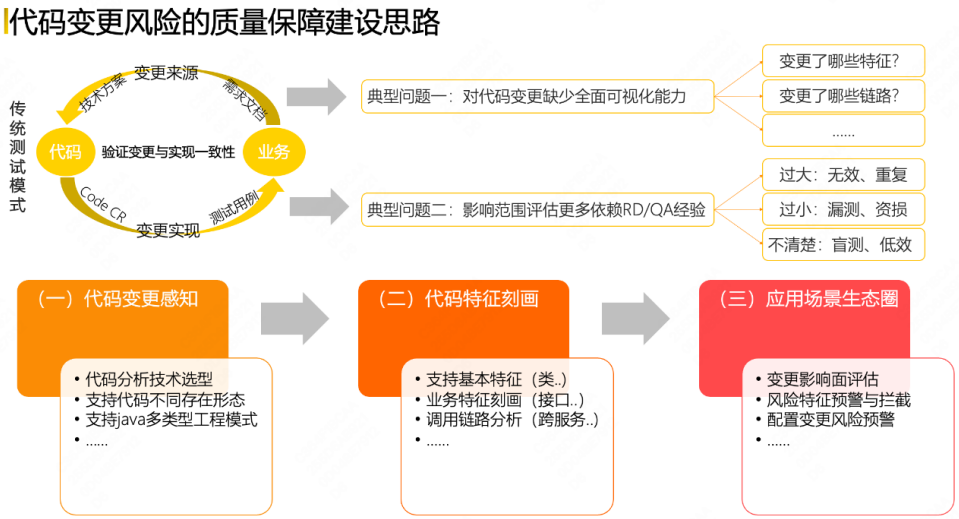
通过分析发现，结合下图的代码特征树，我们就可以更好地感知代码变更的可视化能力。然后通过各叶子节点，将所有相关特征很好地识别，并且做对应的质量防御策略。



2. 代码变更风险可视化（后羿）系统建设

传统测试模式存在两个典型问题：第一，对于研发开发的代码，缺少全面可视化分析能力；第二，对于代码变更所产生的影响范围有多大，实际上更多地依赖研发人员和 QA 的经验。所以，在这样的传统测试模式的典型问题情况下，我们希望从 3 个维度做质量保障建设方案：

- 第一阶段是代码变更的感知能力，重点解决对于不同的代码形态和不同的代码工程模式，如何能够最大范围地覆盖到所有的代码变更情况；
- 第二阶段重点做代码特征刻画，将所有变更代码抽离出不同的特征，打上作用标签；
- 第三阶段基于前两个建设能力，构建对应的应用场景生态圈，在不同测试流水线阶段，都能够将代码变更风险的刻画能力嵌入，不断做质量防御。

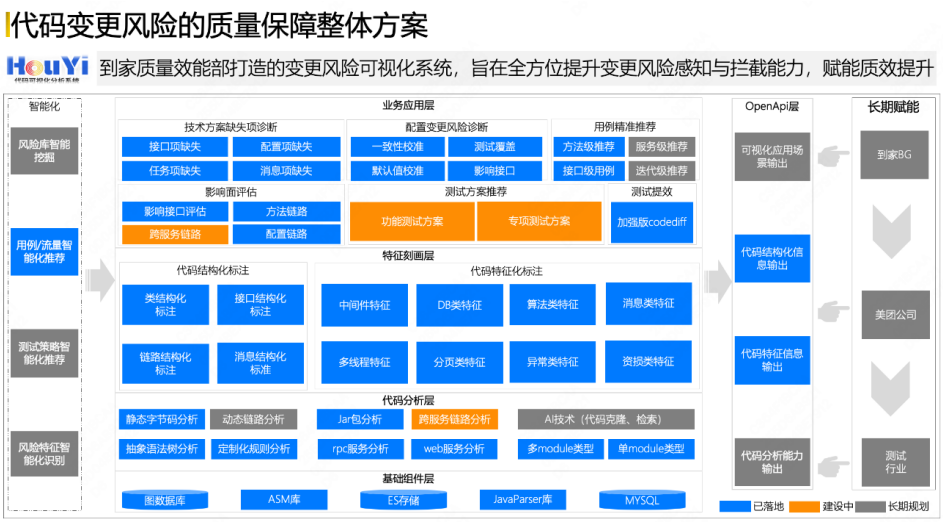


而代码变更风险的质量保障建设方案最终的落地形态，是打造一个代码可视化分析系统，在到家内部我们将其命名为后羿系统。顾名思义，我们希望该系统能像“后羿射日”一样，在质量风险评估和拦截层面能够做到更加精准，同时提升质量防御能力。该系统架构主要分为四层：

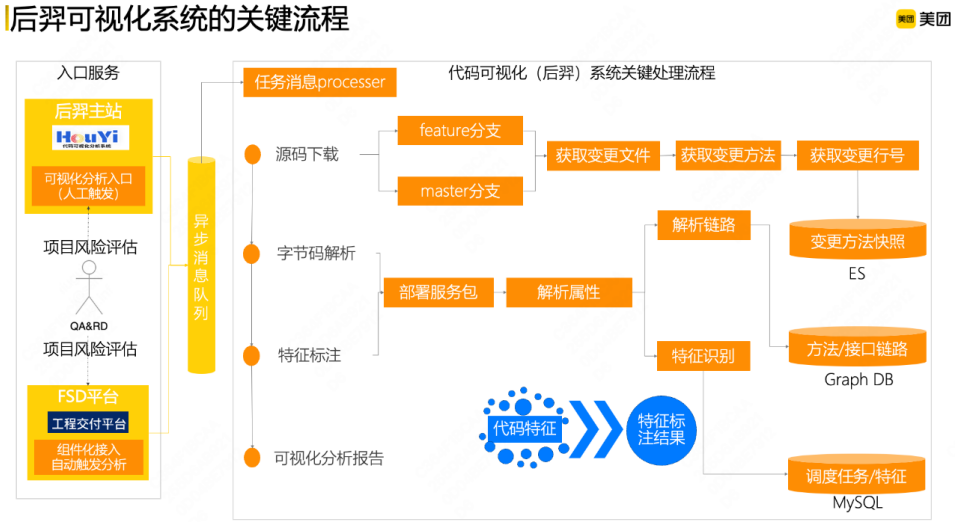
- 第一层是基础组件层。
- 第二层是代码分析层，重点解决对不同的代码形态都能够做到精准的代码变更分析和识别。

- 第三层做特征刻画层，核心解决整个代码的结构化标注和特征化标注。
- 第四层负责构建业务应用层，在不同的场景和环节下，将代码变更可视化能力嵌入到这个环节中，构建一个完备的质量风险拦截能力。

除此之外，我们也会引入智能化手段，赋能整个系统。同时，我们也将核心能力通过开放 Open API 方式，输出到其他的质量效率相关的工具平台上，赋能美团内部的其他工具平台。

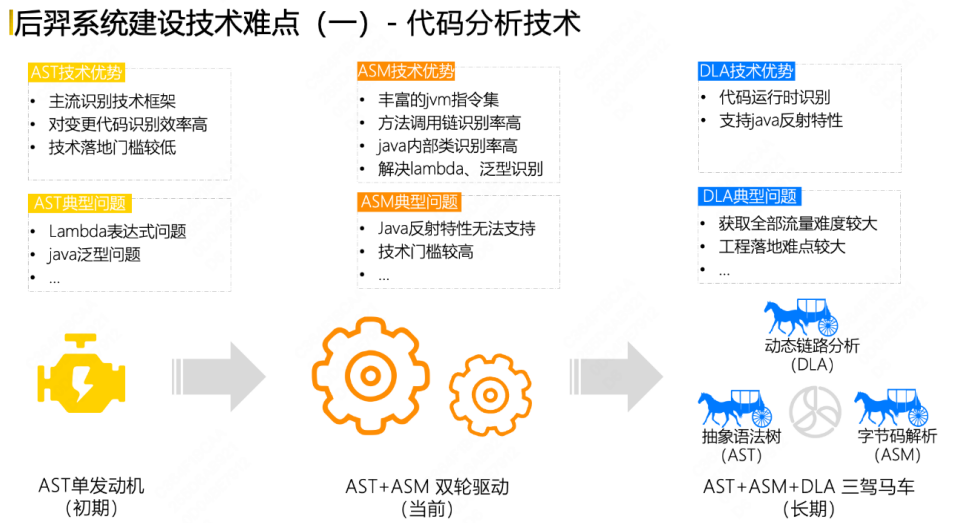


总的来说，对于后羿可视化系统的关键流程，在应用层的透出是通过两个入口进行感知：一是后羿主站；二是工程交付平台。我们通过异步消息感知分析任务和源码下载，获取对应的变更文件、变更方法和变更行号；同时再借助字节码解析能力，解析对应的调用链路变更情况，存储到图数据库里；再对变更代码做特征打标和识别；最后会产生一份可视化分析报告，直接给到对应的 QA 使用。



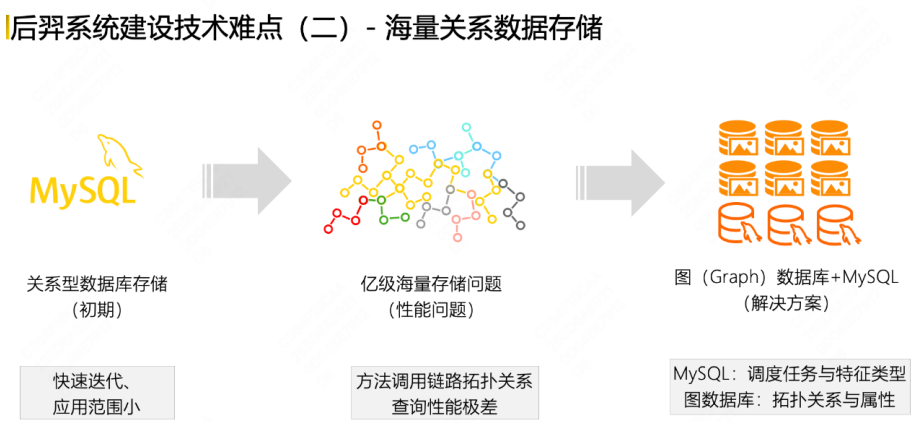
当然，在整个建设过程中，我们也遇到了一些技术层面的挑战。

第一个挑战是代码分析技术。在系统建设初期，通过 AST 单分析能力做代码分析和识别，但存在 Lambda 表达式识别问题和 Java 泛型识别问题，在此基础上引入了 ASM 字节码分析技术解决了前面的问题，但 ASM 也会有自己的 Java 反射特性相关问题无法识别。所以我们希望未来引入动态化代码分析技术，来解决反射类问题。



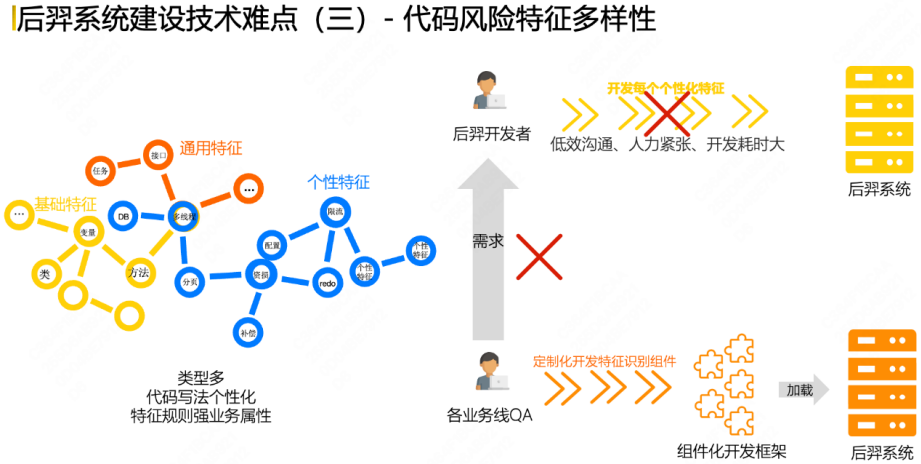
第二个技术难点是海量关系数据存储问题。在建设之初，通过关系型数据库做存储，但随着系统的广泛应用，存储了大量调用链路拓扑关系数据，但它的查询性能非常差。所

以在此基础上，我们通过探索引入图数据库的存储方式，解决了在海量数据关系存储数据的查询性能。



第三个技术难点是代码风险特征多样性，像个性化特征跟我们的业务有强相关性，比如资损特征、对应的分页特征和多线程特征。针对这种风险特征，我们之前的开发模式是系统开发者和业务 QA 深度沟通对应的识别策略，但这样的方式沟通效率和开发周期都比较长。

因此我们做了整体能力改进，通过开发一套组件化开发框架，将整个开发框架开放给各业务线 QA，他们可以自己开发定制化开发组件，加载到后羿系统，完成多样性特征的快速识别能力。



3. 代码变更风险可视化（后羿）系统实践

接下来，我们重点给大家介绍系统的实践应用落地情况。如下图所示，该图为后羿质量保障应用场景生态全景图。目前后羿系统整体建设了八个核心应用场景：

1. 技术方案层面校准诊断
2. 在 Code Review 阶段能力的增强
3. 变更影响面评估
4. 接口级的用例推荐
5. 配置变更风险诊断能力
6. 兼容性风险诊断能力
7. 代码特征风险预警
8. 开放 Open API 能力

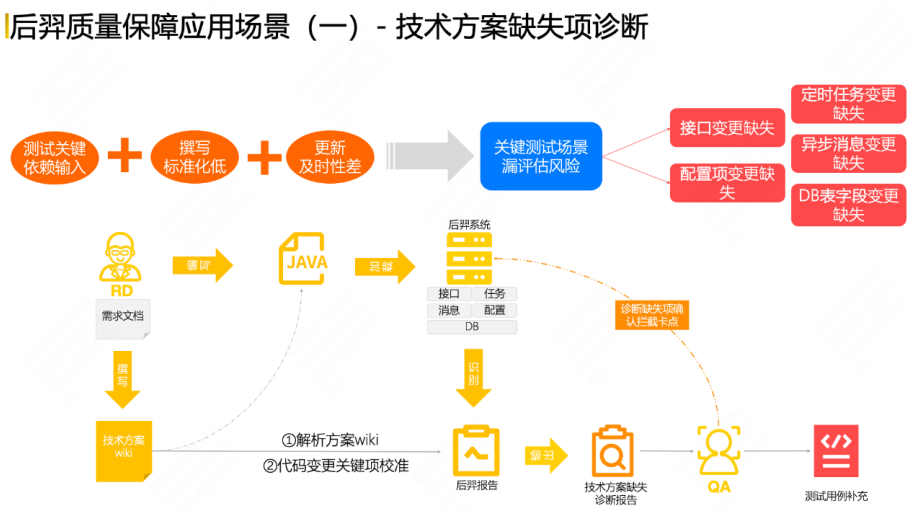
后羿质量保障应用场景生态全景图



首先技术方案缺失项诊断，在项目测试过程中，主要包含以下 2 个痛点：

1. 研发写技术方案时，标准化程度较低。
2. 中间反馈的问题对于技术方案更新时效性较差，但我们发现技术方案对 QA 来讲是一个关键性的依赖输入，从而导致 QA 在写测试用例时，往往会遗漏掉一些关键性的变更信息，比如接口定义变更缺失、配置项定义变更缺失、定时任务变更缺失、异步消息变更缺失以及 DB 表字段变更缺失，这些信息往往在技术方案里更新不全面。

基于这样的情况，后羿系统通过对代码的识别，能够真正拿到在本次代码层面上，真正变更了哪些信息，再拉取对应的技术方案做解析，再做关键变更信息项的 Diff，从而生成一份技术方案的缺失项诊断报告给到对应的 QA，QA 可以根据此报告做对应的诊断项卡点拦截，同时做测试用例的完善补充。



第二个应用场景是增强版的 Code Review 评审新模式，传统的 Code Review 也面临几个痛点问题：

1. 研发在整个 Review 过程中更多地聚焦在编码规范和架构设计合理性上。
2. 常态化的 Code Review 模式基于纯文本，它也有几个痛点：第一是无法针对 Review 的过程做变更方法的上下游快速跳转查看；第二无法做到对变更方法、改动方法的跳转或调用链路的业务逻辑呈现。

基于这样痛点问题，后羿系统打造了基于变更链路场景驱动 Code Review 新模式，核心解决在 Code Review 阶段能够更早地感知质量风险。其核心做法是后羿系统在 Review 一个变更文件时，能够快速提炼出变更文件里对应的变更方法、变更变量以及这些变更方法和变量上都具备哪些风险特征，从而让 QA 和 RD 快速抓到变更的重点信息。

在此基础上，我们也提供了变更方法的上下游快速跳转能力，基于 Review 过程中做变更方法的快速跳转，理解整个业务的上下游关系，同时在跳转过程中，能够将跳转的逻

辑点实时绘制成调用拓扑图，感知变更方法之间的业务逻辑关系，更好地从整个链路视角评估本次代码变更的影响情况，很好地解决在 Code Review 阶段的痛点问题。

后羿质量保障应用场景（二）-增强版CR评审新模式







第三个应用场景是变更影响范围评估，目前后羿系统构建了六大变更影响范围评估能力：

1. 代码基本属性，比如变更了哪些方法；
2. 能够支持到 HTTP、RPC 以及 JAR 包等不同形态代码工程类型；
3. 能够对通用风险特征做识别，比如事物递归兼容性问题；
4. 可以对定制化风险特征做识别，比如权限、算法特征；
5. 可以支持单服务影响，包括方法调用链路，接口特征、消息特征、任务特征等等；
6. 跨端服务的影响面评估，比如一个服务内部的调用链路和在不同的微服务之间的跨端变更点之间的链路影响是什么样子的，都能够做到更精准地影响范围评估能力。

后羿质量保障应用场景（三）-变更影响范围评估



影响面评估示例：

1. 一个变更接口被影响到了，那么它的下游到底是哪些变更方法影响的？我们能够很直观地看到这个变更接口下面有哪些新增的和变更方法所影响到的。
2. 下游所变更的影响方法之间的链路调用拓扑关系是什么样子？我们能够通过链路拓扑图，快速绘制出来对于这个接口下面所有变更方法之间的调用链路。
3. 对应的变更方法对于接口中间所有方法的调用链路详情，我们也能够通过这个能力快速做评估和刻画。
4. 对于一个接口下面所有方法的链路调用关系是什么样子？我们也提供了一个全方法链路的视图分析能力。

后羿质量保障应用场景（三）-影响面评估典型例子



第四个应用场景是配置变更的风险诊断，在比较复杂的大型业务上，整个系统对配置往往有强依赖关系，比如典型的灰度配置、降级配置以及内部逻辑相关控制配置项，对于整个系统的影响比较大，但往往 QA 和研发人员对于配置风险的把控实际上比较缺失，认为代码可能更多的是质量保障的重点，所以由于配置所导致的线上问题比较多，造成的结果比较严重。

基于这样的核心痛点问题，后羿系统重点从三个层面做了关于配置变更风险的核心能力建设：

1. 在配置的变更识别分析层，我们能够很好地对各种类型配置做精准识别：是新增还是变更。
2. 通过后羿影响面评估能力，拿到配置所影响的接口、影响链路以及对应影响的异步消息和定时任务。
3. 有了影响面评估后，我们能够更好地通过测试，将变更场景做到覆盖，由此构建了配置变更测试覆盖度量。因为我们在配置测试过程中对于测试结果的感知能力不强，因此通过基于流量挖掘、流量录制能力构建了实时感知配置测试的覆盖情况。在整个流水线层面，通过关键配置变更卡点能力更好地防御配置变更所引发的一些问题。

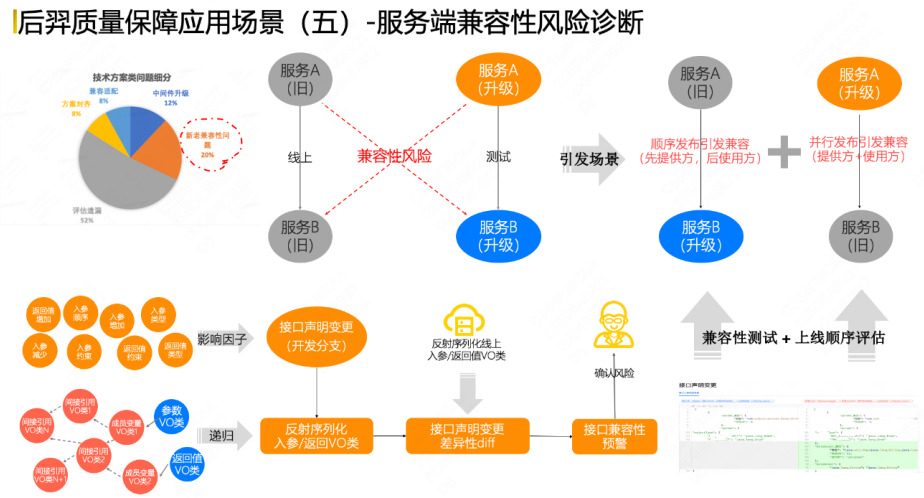
下图是我们目前所建设的应用功能页面：



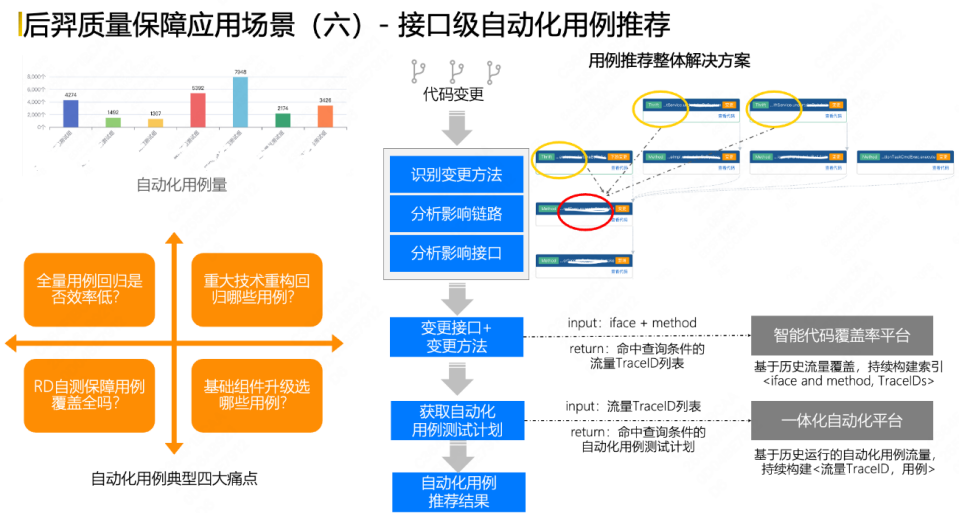
第五个应用场景是服务端兼容性的风险诊断。我们通过对线上问题做汇总分析发现，新老兼容性这类典型问题占比较高，我们尝试通过后羿系统解决，QA 能够做简单的兼容性问题识别，比如一个接口的入参返回值有明显的字段新增或类型变化会明确判断出来。

但在兼容性问题分析上，有一类不太明显的变化导致了兼容性问题，比如入参层面有一些约束条件、由可选字段变成了必选字段，这类变化实际上在整个代码定义层面很难感知到；另外一类是变更参数是通过内部间接调用 VO 类直接组装出来的，实际上对于 QA 也很难感知内部间接 VO 类变化的兼容性风险影响。

所以后羿系统基于这样的痛点，构建了反射和序列化，从而快速拿到对应的底层变更 VO 类所导致的对接口影响能力，给出兼容性接口预警，QA 根据这样的分析诊断报告，进一步评估对应的兼容性和上线顺序的合理性安排。



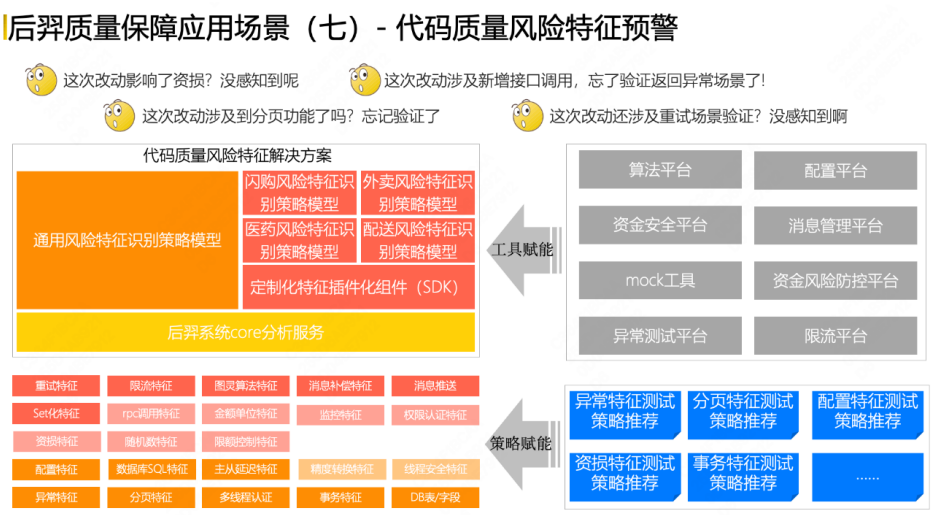
第六个是接口级自动化用例推荐，对于到家的复杂业务，我们沉淀的很多自动化用例怎么用，是全量回归还是选择性筛选，也是比较大的痛点问题。因此后羿系统基于所具备的识别变更方法、分析影响链路以及对应的接口能力，打通和到家智能代码覆盖率平台所具备的历史流量覆盖情况，能够快速拿到变更接口和方法，再借助一体化平台，拿到对应的自动化用例，做精准的自动化用例推荐，从而构建了用例推荐整体解决方案。



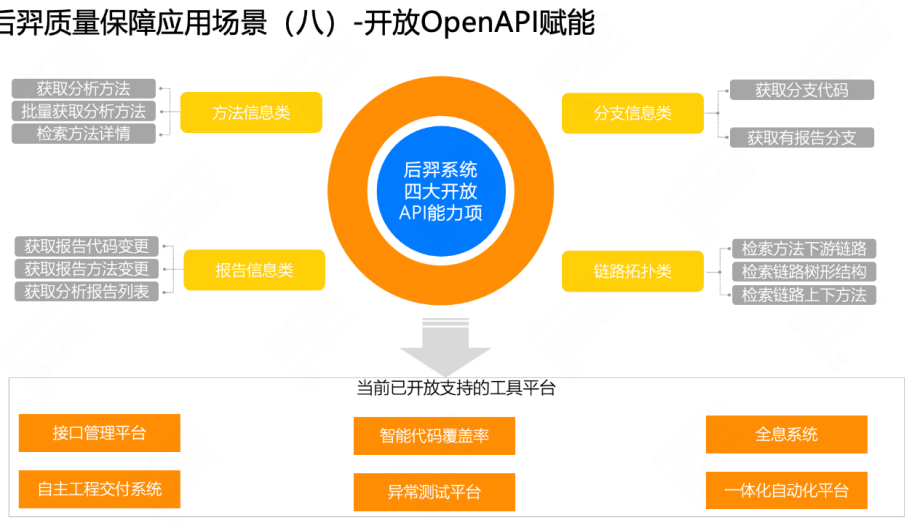
第七个典型的应用场景是代码质量风险特征预警，我们在质量建设过程中，往往会遇到一类比较特殊的场景需要验证，比如资损类场景，除了验证功能外，还需要对资损做个性化风险功能场景验证、异常场景验证、特殊分页逻辑、重试场景验证，但这些场景在整个代码过程中，我们往往很难发现这些特征风险，从而忘记验证特殊场景情况。

针对这个问题，后羿系统从两个方面构建了特征风险识别功能：一是系统会自己构建通用风险特征识别策略模型，二是各业务方也会自己打造对应的风险特征识别策略。

如下图所示（最下侧），是目前已经具备和未来将要建设的特征识别能力。有了核心识别能力后，我们再将对应的特征在所要验证过程中相应上下游的依赖平台做工具能力整合，对于不同特征构建出相对应具体测试策略的推荐策略，将这几个能力打通后构建出一套基于代码质量风险特征的体系化质量保障方案。



第八个应用场景的能力是开放 Open API 的赋能能力，我们希望将后羿系统分析识别的信息，通过开放 API 的方式透露出去，能够给到对应相关工具平台使用。目前在到家范围内已经将整个能力开放给了接口管理平台、智能代码覆盖率平台、全息系统、异常测试平台、自主工程交付系统以及一体化自动化平台这六个核心的效能工具建设平台。



4. 未来规划与展望

结合具体的实践，以及此前总结的经验。未来，我们将从四个方向做未来质量保障建设：

1. 代码分析技术的增强，希望能够通过动态链路分析技术，提升整体的分析准确性。
2. 风险特征识别技术，希望能够基于大语言模型相应能力对风险特征的分析和对应测试策略做推荐。
3. 进一步丰富应用场景生态圈，探索在不同测试环节、不同测试场景下，将后羿能力做更好的整合，赋能到具体的业务质量建设中。
4. 长期来讲，希望将系统的核心能力，通过开源共享方式赋能给相关测试领域。



5. Q & A

Q1：单次分析报告耗时是多久？同一个工程代码的报告间是独立的还是有关联的？

A：目前 1-2min 可产生分析报告。报告是基于迭代任务维度生成的服务级报告，比如一个项目迭代里有 5 个工程代码变更，那么这 5 个工程代码变更关系会作为一个整体分析，体现到报告里。

Q2：链路的拓扑关系是怎么实现的？

A：链路的拓扑关系基于 AST 技术和 ASM 技术进行分析，同时，结合线上 Mtrace 链路关系作为补充。

Q3：微服务多模块之间的服务调用链路可识别出关联性吗？如 HTTP 接口后续调用多 RPC 服务接口？

A：微服务多模块之间的服务调用链路可识别出关联性，比如 HTTP 的下游会调用了哪

些 RPC 接口，目前具备可识别能力。

Q4：底层方法的调用链非常多，这种有推荐策略吗？ DB 变更是扫描 Mapper 吗？

A：目前对于调用链路会进行风险特征打标处理，比如：链路包含资损特征、配置特征等，通过特征风险标签能够更好的让用户感知到链路风险等级，为后续测试策略提供关键信息推荐。目前 DB 变更是通过扫描 Mapper 进行识别的，基于 Mybatis 整体 DB 开发框架做变更扫描。

Q5：关于用例推荐，是推荐单例接口还是会组合成场景接口？

A：目前是推荐单接口用例，比如这个变更接口关联了 10 个自动化用例，我们会把这 10 个自动化用例推荐出来。

Q6：分析一个工程大概会花费多长时间？

A：目前要花费 1-2min 时间。

Q7：实际应用中，主要收益是什么？

A：主要收益是基于八大应用场景落地应用，所有应用场景都会给质量与效率带来价值收益，比如在兼容性问题上已成功拦截多起兼容性质量问题；在配置变更风险评估上，已成功拦截多起因配置默认值编码错误、线上线下配置不一致问题。

Q8：这套平台是否会对线上服务可用性带来影响？

A：目前这个平台对于线上服务可用性不会造成影响，针对线上环境进行分析时，核心操作是拉取线上对应的部署 JAR 包，不会对现实服务造成可用性影响。

Q9：这个系统有护城河吗？这个系统收益是什么？对业务有什么帮助吗？在业务上有什么体现吗？

A：说到“护城河”，变更分析底层技术上主要还是基于 AST 和 ASM 通用技术，核心是对各种业务代码编写方式的兼容支持、业务场景特征识别精度等维度上有一定的技术挑战。

主要收益集中在变更带来的业务影响范围的风险评估及对质量风险的有效拦截上，目前已成功拦截多起接口影响范围评估不准的 Bug、配置变更相关 Bug、兼容性 Bug 等。

在业务帮助上，一是八大应用场景能够给整个质量和效率带来提升；二是可提升对业务理解的效率，比如通过接口下游调用链路可视化能力，可很快理解这个业务链路关系。

Q10：链路拓扑过大，怎么解决，不同服务使用语言不同，字节码技术有啥推荐吗？

A：链路拓扑过大时，可通过链路聚合来归类提升可视化，针对美团是 Java 技术栈，重点在突破 Java 识别能力，其他语言服务分析后续会持续关注与解决。

Q11：可以分析出对上下游服务的影响吗？

A：是的。

Q12：分析代码变更，如何识别有效变更和无效变更？

A：比如变更了一段代码，它有可能没有调用方，我们叫做预埋类代码。类似这种代码在测试过程中很难用业务场景进行验证，而且未来上线后很可能会引入潜在风险，目前通过链路变更分析可有效识别此类预埋代码风险。

Q13：系统使用的阶段都是什么？准入阶段还是准出阶段？

A：目前系统在准入和准出阶段都可以使用，没有做严格限制。

Q14：识别出的问题噪音干扰多吗？

A：从噪音干扰层面讲，我们做了很多提升识别率的策略优化，比如 HTTP/RPC 接口识别率目前已达到 98% 以上，噪音干扰可做到很好的控制。

6. 本文作者

桂来，来自美团到家研发平台。

基于模式挖掘的可靠性治理探索

作者：玉鑫

1. 可靠性治理的痛点

对于亿级流量的线上系统来说，可靠性是至关重要的。从字面上理解，可靠性要求故障少、可信赖。与安全性一样，它们都是信息系统的固有属性之一，也是保障产品质量的关键因素。

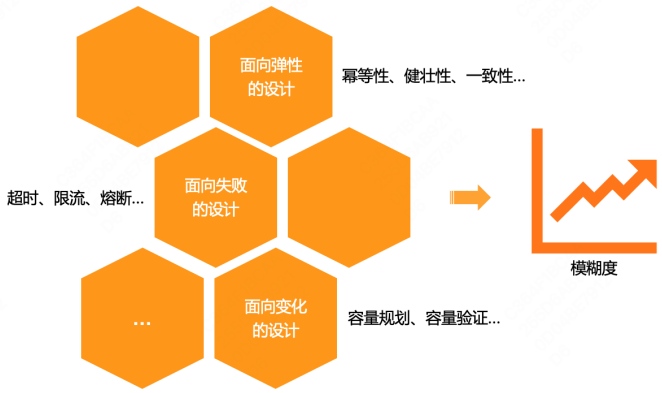
对照 Google 的可靠性模型来看，测试同学会投入很多精力在用例设计、测试执行、持续交付等环节上，研发同学则会更多关注监控、应急和故障分析等。但往往由于项目进度和人力因素，在设计和编码阶段对可靠性的投入和关注不足，导致后续需要付出更高的成本发现和解决潜在隐患。有鉴于此，我们希望能找到更低成本且以更有效的方式发现和治理这些隐患，从而提升系统整体的可靠性。



在研发设计阶段，我们需要关注系统弹性，考虑潜在故障风险、适应流量变化等，其中相关治理涉及幂等性、健壮性、一致性、超时、限流、熔断等场景。与一般功能测试相比，可靠性治理需要面对不同的服务和系统，发现并治理技术问题，在模糊度上有较大的提升和挑战。就目前而言，质量问题非常明确，但潜在风险策略和解决路径比较模

糊。因此，我们希望能找到办法识别并解决这些问题。

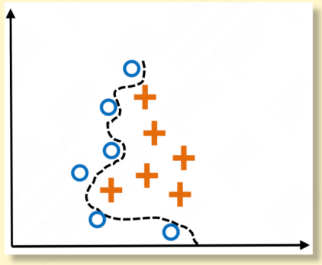
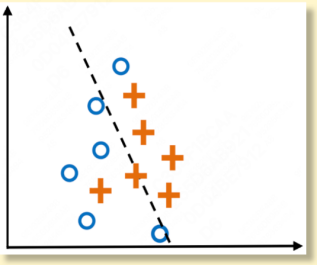
可靠性治理的挑战



模糊度的提升会带来两种最常见的现象：

- 一种是过于具体，Case by Case 解决问题，类似算法的过拟合，过拟合的问题在于对更广泛范围内的问题缺乏有效性。以幂等性为例，想验证一个接口是否幂等可以很快完成并很快补充接口幂等相关的测试用例，但是对不同的接口、服务、系统以及不同的幂等性设计，还有哪些问题和风险，我们没有办法关注到并控制这些风险。
- 另一种是过于泛化，类似算法的欠拟合，欠拟合的问题在于过度虚化导致没有抓住问题的共性特征。以主从延迟为例，主从延迟会给系统带来一致性风险，需要针对性做保护，并进行相关验证，因此我们可以制定规范、梳理 Check List 和测试模板，虽然这样可以最大程度在产研各环节提醒大家关注到这类问题，但并没有找到彻底解决问题的方法。

最常见的两种情况

Overfitting：很多时候我们只能Case By Case的处理单点问题，无法解决潜在的同类问题。例如一个接口存在幂等性问题，那么我们可以很快修复问题，并添加幂等性相关的用例。但还有哪些接口有问题是无法穷举的	**Underfitting**：我们清楚知道某一类问题需要被关注，但不能确定具体的解法。例如主从延迟会带来一致性风险，我们会强调相关验证的重要性，并制定规范，但由于没有确定的规则，并不能保证这一类风险得到控制
	

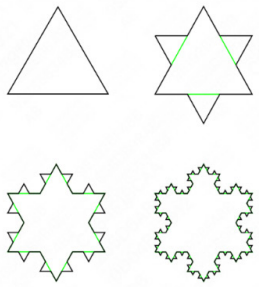
2. 模式的定义

类似这些问题如何找到更好的解决办法？我们重点看一下模式对可靠性治理的启发。模式在维基百科的定义是：揭示了这个世界上人工设计和抽象思想中的规律。

例如下图所示，计算机图形学中的经典分形图案柯赫雪花，是 1904 年瑞典数学家科赫提出。可以看到它有明显的规律，这样的分形规律在自然界无处不在。

维基百科中的定义： A pattern is a regularity in the world, in human-made design, or in abstract ideas

「模式」揭示了这个世界上，人工设计和抽象思想中的规律。



1904年，瑞典数学家柯赫构造了 “Koch曲线” 几何图形



技术场景的模式会更加丰富些，这类模式和可靠性治理想找到的模式非常接近。

举例缓存设计的两种常见模式：

- 第一种是 Cache-Aside（旁路缓存），也是使用比较广泛的一种方式，它只有在缓存没有命中时，才会查询数据库并更新缓存。
- 另外一种是 Write-throught（只写模式），这种模式在每次数据库变更时都会同步更新缓存。

对比第一种模式，第二种模式的优点是逻辑更清晰、操作简单、缓存命中率更高；缺点是不常请求的数据会被写到缓存中，导致缓存更大。

技术场景中的模式



Cache-Aside (Lazy Loading)　　Write-Through

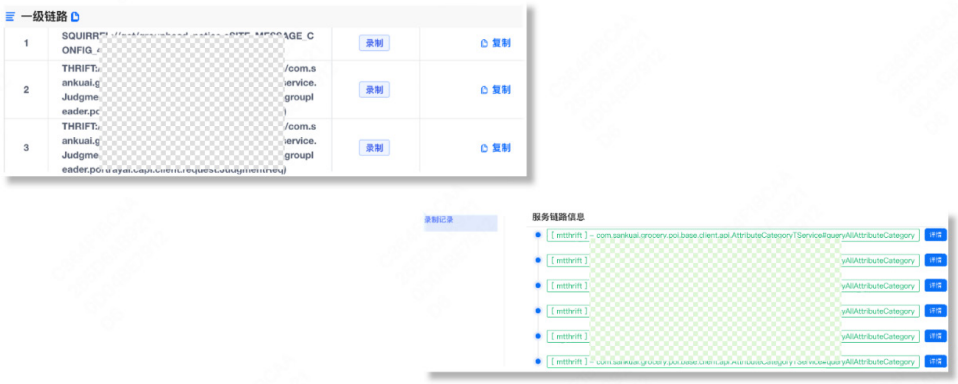
那么，我们如何找到这些潜在模式并应用到可靠性治理呢？我们现有的业务测试数据、专业知识积累、相关问题分析和复盘经验，都可以帮助我们找到治理这些通用技术场景的规律。在这里，很重要的一部分是真实的业务数据，我们可以从最基础的数据提取信息，找到解决共性问题的思路。

3. 大数据下的尝试

随着 JVM 非侵入式 AOP 解决方案的成熟，现在我们已经可以采集任意环境下的任意协议流量，以及这些流量依赖的数据关系，也可以在测试环境回放这些流量，包括线上真实采集的流量。这里依赖两个关键点：一是流量采集，这里涉及很多技术方案，这里分享主要是作为一个基础设施；二是有了全链路 Mock 能力，我们才能在测试环境进行各种流量的回放和验证。

流量采集技术的成熟

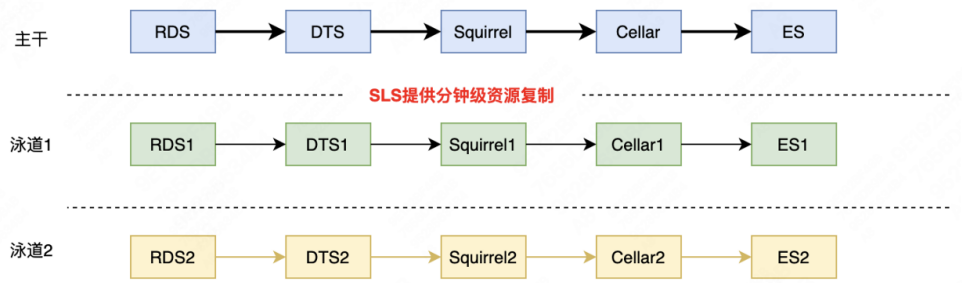
随着JVM非侵入式运行期AOP解决方案的成熟，我们可以进行任意环境下各种协议流量和依赖数据的采集。也可以在测试环境回放线上的真实业务流量。



另一个重要基础设施是环境隔离技术，经过快速发展，它已经具备了泳道级别的数据复制隔离，也支持一站式数据消息和部署环境的即拿即用。从最开始通过泳道降低人工测试相互影响，到现在一站式拉出一套环境，能支持各类专项检查独立运行，使用线下数据且不污染主干环境。

环境隔离技术的发展

依托分布式弹性块存储服务，测试数据隔离已经具备一站式解决方案，如存储编排快速构建和销毁、存储编排管理、数据隔离与自动路由、数据同步/导入等，业务在使用时只需做简单的配置即可做到测试环境数据“即拿即用”。



基于流量采集和环境隔离这两个能力的成熟，给自动化领域带来了很多新可能。流量采集信息包含了请求参数、返回值和调用链路等信息；环境隔离技术支持数据隔离、消息隔离、各种协议以及部署版本隔离。

在这种情况下，海量的业务流量可以直接转化成基于规则验证的接口自动化用例，也可以应用到基于业务模型的场景级用例，模式在这里更像是两者之间的“折中”，我们希望通过这种“折中”来解决可靠性治理的难题。

4. 典型实践分享

接下来，我们重点介绍基于模式的三个可靠性治理的典型实践，主要包括幂等性治理、依赖治理和越权治理三个方向。

4.1 幂等性治理

维基百科中，幂等的定义是数学和计算机科学中某些运算的性质，可以被多次应用，而不会改变最初应用之外的结果。在数学中也有相关的定义，就以一元运算为例，当 f(f(x))=f(x) 时，可以认为这个运算符 f 是幂等的；在计算机科学领域，HTTP 规范中也有对幂等的定义，即多个相同请求的副作用与单个请求相同，例如 GET、PUT 和 DELETE 是幂等的。

幂等性治理-背景

维基百科中的定义：Idempotence is the property of certain operations in mathematics and computer science whereby they can be applied multiple times without changing the result beyond the initial application

一元运算中，当运算f满足f(f(x))=f(x)时被认为是幂等的。

二元运算中，为集合S的元素x配备二元运算*，则*在满足x*x=x for all x ∈ S时被认为是幂等的。

HTTP/1.1规范中的定义：Methods can also have the property of "idempotence" in that (aside from error or expiration issues) the side-effects of N > 0 identical requests is the same as for a single request

Operation	GET	HEAD	OPTION	POST	PUT	PATCH	DELETE	TRACE
Idempotent	Y	Y	Y	N	Y	N	Y	Y

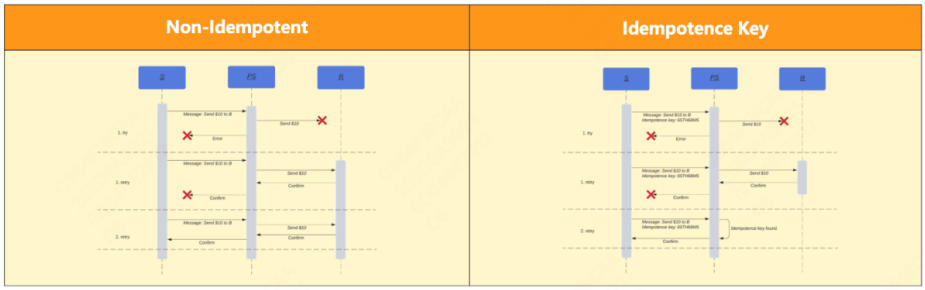
在大部分分布式系统中，请求超时、网络抖动等在线上环境中随时可能发生。幂等性设计是保证服务在高并发情况下高可用的关键。对于每天产生海量订单的线上业务，比如库存、交易、支付和财务等系统都需要通过幂等性避免超卖、重复支付、重复打款等问题的发生；同时幂等性也是消息队列、定时任务、分布式事务等技术类场景稳定运行的基础。

如下图举例，当一次调用部分成功的情况下，系统会触发重试，而幂等性可以保证在重试时，成功部分不再被重复执行。

幂等性治理-背景

业务场景中的幂等：在大规模分布式系统中，请求超时、网络抖动等随时可能发生。幂等性设计是保证服务在高并发的情况下高可用的关键。对于每天产生海量订单的零售业务，库存、交易、支付和财务等系统更需要通过幂等性来避免超卖、重复支付、重复打款等问题的发生。

技术场景中的幂等：幂等性除了对业务场景至关重要，同时也是消息队列、定时任务、分布式事务等技术类场景稳定运行的基础。



我们要挖掘通用模式，就需要分析幂等性所有可能的实现方案。

如下图是几种常见的幂等性实现方案：数据库层面的唯一索引；通过版本或其他状态、条件来限制操作执行的悲观锁和乐观锁；通过具体业务属性参数，构造具有唯一性的 Token 以及分布式系统中广泛使用的分布式锁等。

幂等性治理-模式挖掘

由于幂等性的实现方案有很多种，不同的实现方案需要不同的检查策略。在对获取的流量进行幂等性检查之前，我们需要确定哪些流量是要检查的，同时也需要结合流量的调用过程等信息标记检查的策略和优先级。

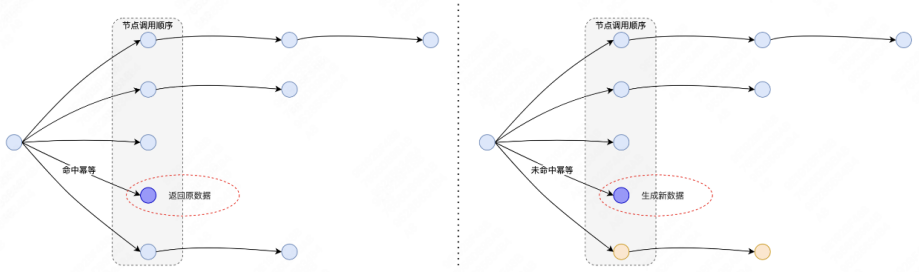
唯一索引	• 通过在数据库中的表设置唯一索引，保证索引相同的请求只能有一次写成功，可以通过先查后写，或者处理DuplicateKeyException保证幂等性
悲观锁	• 可以通过版本或者其他状态条件来实现，在数据库查询和更新操作时锁表
乐观锁	• 乐观锁只在数据更新时锁表，其他时候不锁表，所以比悲观锁效率高 • 有限状态机（FSM）也是乐观锁的一种，以单据相关业务为例，当状态机已经处于下一个状态，这时来了之前状态的改变是不能成功的，就保证了有限状态机的幂等性
Token	• 由一组有业务含义的参数组成UniqueKey，通过UniqueKey检查保证不会出现重复提交
分布式锁	• 对于分布式系统，构建全局唯一索引比较困难，这时候可以引入分布式锁，同一时间该流程只能有一个能执行成功，执行完成后，释放分布式锁（分布式锁要第三方系统提供）

尽管幂等性的实现方案有多种，但回到幂等性的本质，我们希望多次调用不会产生新的副作用，而系统中副作用的产生往往是通过“写”操作发生。

分析调用链路发现，当链路上某个节点不幂等而对资源产生副作用后，其后的多个节点都可以检测到相关变化。例如，前序节点通过数据库的写入生成了新的单号，后序节点的参数和返回值很可能会出现这个新单号。这样，我们就可以构造多次同样的请求，之后检查链路上的这些变化来验证幂等性。

幂等性治理-模式挖掘

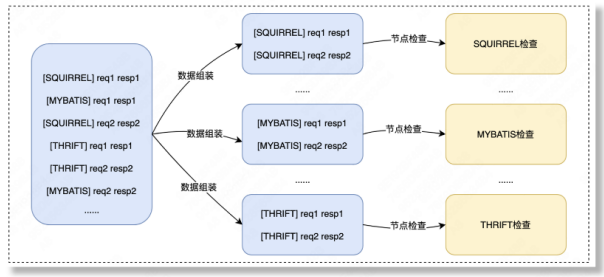
分析调用链路发现，当链路上某个节点不幂等而对资源产生副作用后，其后的多个节点都可以检测到相关变化。例如，前序节点通过DB的写入生成了新的单号，后序节点的参数和返回值很可能会出现这个新单号。这样，我们就可以构造多次同样的请求，之后检查链路上的这些变化来验证幂等性。



调用链路节点的类型包括了 MYBATIS、RPC、HTTP、MAFKA、CRANE 等，不同幂等性方案在不同类型的节点上有相应的表现，例如唯一索引，更多在 MYBATIS 节点上，不同类型节点的检查策略和优先级也不同。

幂等性治理-模式挖掘

调用链路节点的类型包括了MYBATIS、RPC、HTTP、MAFKA、CRANE等，不同类型节点的检查策略和优先级都不同。我们会将调用过程中不同类型的节点分开检查，通过对实际检测结果的数据分析，我们发现MYBATIS节点的有效性更高。



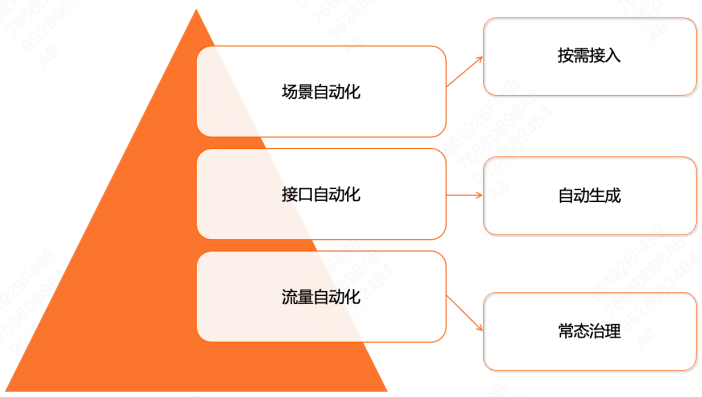
如下图，列举了部分节点检查策略和降噪策略。以 MYBATIS 为例，我们会关注到写 SQL 的内容和返回结果，结合索引冲突、锁失败等节点的异常返回进行降噪。比如 THRIFT 节点，我们会关注接口的参数和返回值变化，考虑随机 ID 的生成、时间戳字段等进行相关降噪，最终针对不同幂等性方案和不同节点类型形成通用整体策略。

幂等性治理-节点检查策略

节点类型	优先级	检查策略
MYBATIS	高	➢ **比对**：验证有增/删/改操作且成功的节点信息是否有变化 ➢ **降噪**： • 识别幂等表、主键冲突、抢锁失败等命中幂等性方案情况 • 识别log后缀的日志表、history后缀的历史表等
THRIFT	中	➢ **比对**： • 识别leaf.IDGen等随机ID生成接口 • 验证接口的请求和返回是否有变化 ➢ **降噪**：识别请求参数和返回值中的时间戳字段、随机字段等
SQUIRREL/ MAFKA ...	低	➢ **比对**：验证节点的请求和返回是否有变化 ➢ **降噪**： • ...

基于通用检查能力，我们可以应用在场景用例编写、流量用例生成和离线流量的自动分析上，通过分析每天线上、线下环境产生的真实流量，我们可以对增量和存量问题进行差异化治理和跟进。

幂等性治理-应用场景



4.2 依赖治理

随着微服务的发展，我们的系统变得越来越复杂，调用链路越来越长，例如单接口的下游依赖多达上百个，任何外部依赖的抖动都会成为核心业务的威胁，很多时候系统内部或外部的一些错误被激活，没有得到正确处理，就会在服务内部不断传播，导致系统偏离正确的服务状态，造成服务失败，最终导致业务失败，引起用户可以感知的故障。

依赖治理-背景

伴随分布式微服务的发展功能被分解到各个离散的服务中，系统正在变得越来越复杂、调用链路越来越长，如提单接口的下游依赖可以多达上百个，任何外部依赖的抖动都会成为核心业务的威胁，而提升交易服务可用性需要提升系统容错能力，即保障交易系统能够在链路中的某些一个或多个组件故障发生时继续正常运行，系统容错能力在高可用性或生命攸关的系统中尤其重要，如面向用户端的服务。

依据“Chain of threats（威胁链）”，Error（错误）和Failure（故障）Fault（故障）之间存在因果关系。可以笼统地解释故障发生之前产生几个错误，通过传播最终会引起故障。起因为系统内部的或外部的错误被激活，经过裁定后定义为服务编排内部错误，这时对外部用户是无感知的，但是错误在服务内部不断传播，导致系统偏离正确的服务状态造成服务失败，最终会导致业务失败引起外部用户可感知的故障。对于可用性高达99999的服务，当服务规模大于10000时，小概率的硬件故障每天都会发生。



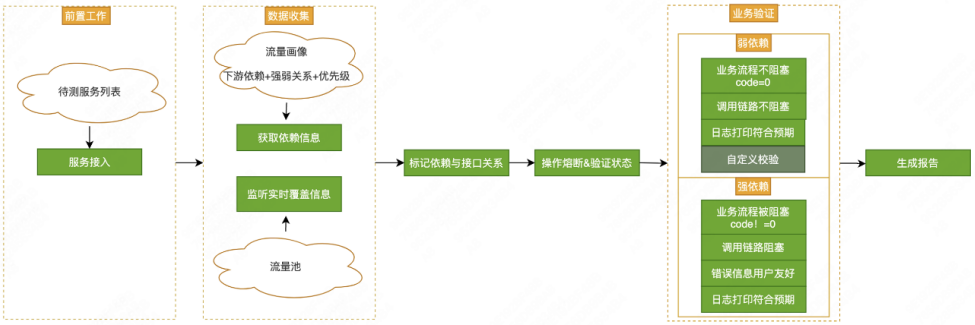
在业务上可以通过依赖分级和熔断策略，保障弱依赖发生故障时，核心流程依然可用。因此我们需要进行依赖治理，而依赖的治理关键在于如何自动化完成分级合理性以及熔断策略有效性的验证。

类似前面，我们会采用回放业务流量的方式，但基于依赖治理，我们的策略是修改依赖的 Mock 结构，构造依赖故障场景，进行相关验证。

我们的预期是如果命中了弱依赖，我们期望业务主流程不被阻塞，调用链路也没有阻塞，日志打印和返回信息都符合预期，没有异常表现；如果命中强依赖，验证策略则相反。

依赖治理-模式挖掘

业务上可以通过依赖分级和熔断策略，保障弱依赖故障后核心流程可用。依赖治理的关键在于如何自动化的完成分级合理性及熔断策略有效性的验证。



具体的策略是我们依据接口和依赖关系构造指定依赖故障场景，注入异常后，分析这个异常是否被捕获。如果直接抛到了外层或者接口返回值有相关的异常信息，那当前是强依赖；如果注入依赖后，后续的调用链路被阻断，认为当前依赖是强依赖。反之，则是弱依赖。

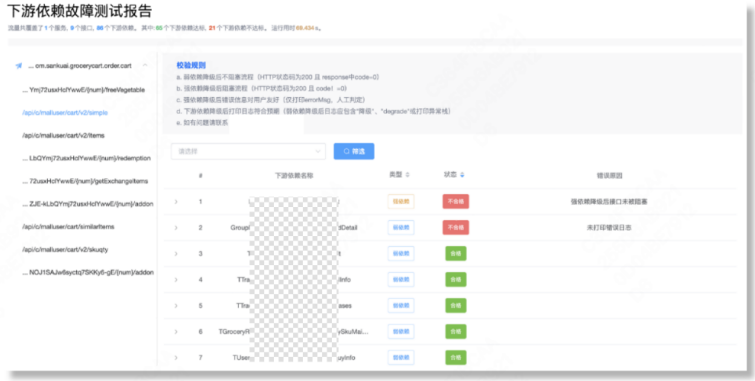
具备这样闭环依赖分级识别以及熔断有效性的治理能力后，我们就可以周期性地对核心服务进行下游依赖等级治理和对熔断策略有效性进行自动验证。

运营内容主要包括两方面：配置检查和业务验证。

- 对于依赖等级正确与否的检查，每周运行，发现依赖等级与熔断策略不相符的情况，推动治理。
- 对于业务验证，每天运行，持续产生每天增量报告，针对强依赖业务未被阻断、弱依赖业务未被处理，对应的异常等问题推动修改。

依赖治理-应用场景

运营主要包括两方面，配置检查和业务验证。针对配置正确与否的检查每周运行，发现依赖等级与熔断策略不符合的推动修改，业务验证每日运行完成后产出业务验证报告，针对强依赖业务未阻断、弱依赖业务处理异常等问题推动修改。



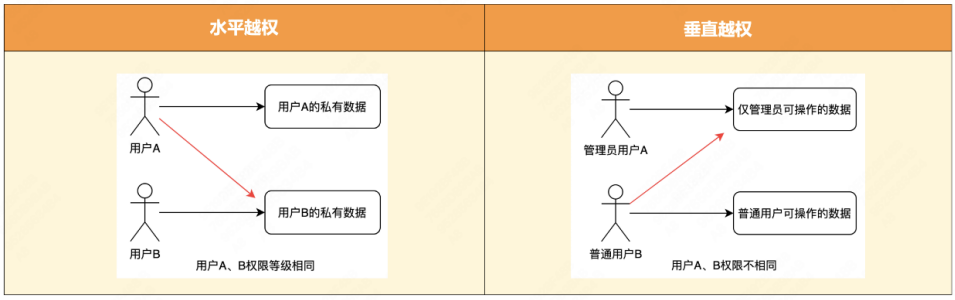
4.3 越权治理

越权访问是 Web 应用程序中一种常见的漏洞，它存在范围广、危害大，被 OWASP 应用程序安全项目列为 Web 应用十大安全隐患第二名。对于这种商户、用户规模大，交易频繁的线上业务来说，更是存在比较大的安全和合规风险。

越权就是两个同等级用户，一个可以操作另一个数据；垂直越权则涉及到不同等级用户，例如普通用户可以操作管理员才有的权限数据。

越权治理-背景

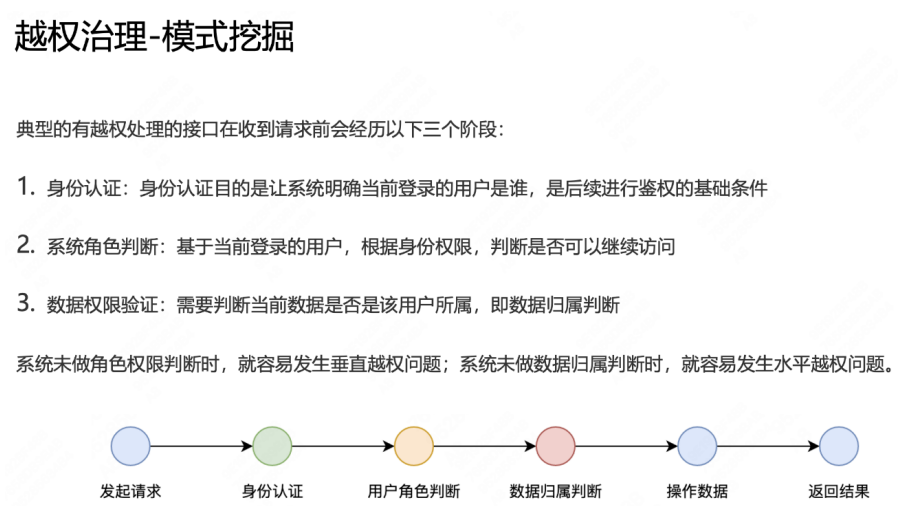
越权访问是WEB应用程序中一种常见的漏洞，由于其存在范围广、危害大，被OWASP列为WEB应用十大安全隐患的第二名。对于商户、用户规模大，交易频繁的线上业务来说更是存在较大的安全及合规风险。



典型的有越权处理接口在收到请求后经历以下三个阶段：

- 第一步是身份认证，让系统明确当前登录的用户是谁，是后续进行鉴权的基础条件，每家公司和业务可能会有多套鉴权系统。
- 第二步是系统决策判断，基于当前登录用户信息，根据身份权限判断是否可以继续操作。
- 第三步是数据权限验证，判断当前数据是否是该用户所属，即数据归属判断。

当系统未做角色判断时，容易发生垂直越权问题；当系统未做数据归属判断时，容易发生水平越权问题。



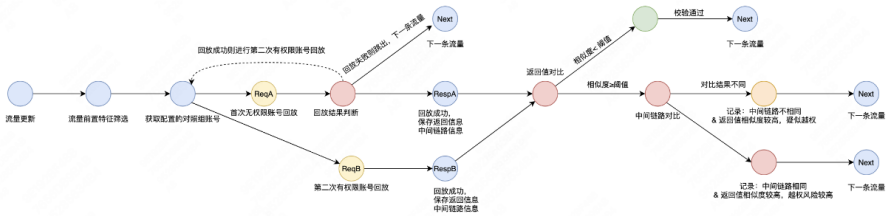
我们可以通过回放业务流量构造对应场景，验证接口是否有做权限控制。

第一次回放，会结合识别到当前流量鉴权方式，构造一个无权限账户进行回放，其余的依赖数据保持不变；第二次回放与第一次类似，只不过需要构造一个有权限账户信息进行回放；比对两次回放结果以及调用链路，验证这个接口是否有相关的鉴权逻辑；再结合调用链路对比以及原始流量的调用链路，比较有效地识别当前的鉴权场景，兼容一些场景通过返回值没有办法完全识别到是否有做鉴权的情况。

越权治理-检查策略

对采集到的业务流量，我们可以进行如下操作来验证接口是否有权限控制。

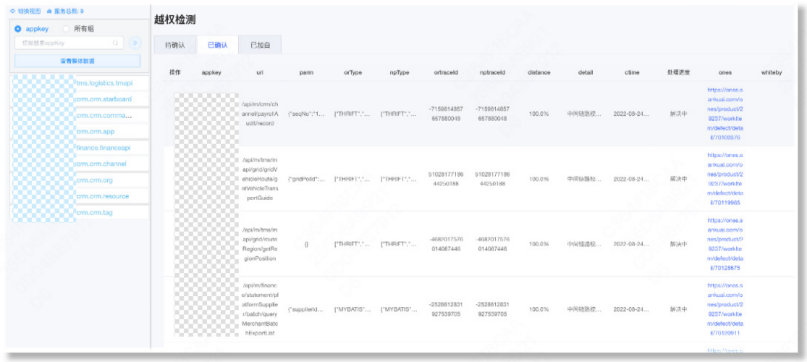
1. 第一次回放，结合识别到的鉴权方式修改Mock信息，构造无权限账户信息回放，其余依赖数据不变
2. 第二次回放，结合识别到的鉴权方式修改Mock信息，构造有权限账户信息回放，其余依赖数据不变
3. 返回值对比：比对两次回放接口的返回值，验证接口有鉴权能力
4. 调用链路对比：对比两次回放以及原始流量调用的链路，应对那些仅靠返回值不足以验证权限的场景



在实际应用中，要考虑我们所使用的流量质量、有效性以及鉴权方式等进行筛选。目前越权检查经过优化和适配不同业务，已经可以自动化、常态化对新增流量进行检测，并将结果同步到报告中，进行日常运营，也支持问题确认、加白和工单创建等。

越权治理-应用场景

目前越权检查经过优化适配业务，已经可以自动化、常态化的对新增流量进行检测并将结果同步至自动化建设平台进行日常运营，包含问题确认、加白、一键创建工单等。



以上三个治理能力，已经对美团优选部分核心服务默认开启，可以低成本自动化实现相关问题的常态治理及运营。目前覆盖了 500+ 服务、2 万 + 接口和 8000+ 下游、累计发现并治理问题有 1000+。近期已经开始陆续接入到公司内其他业务线进行应用。

建设收益

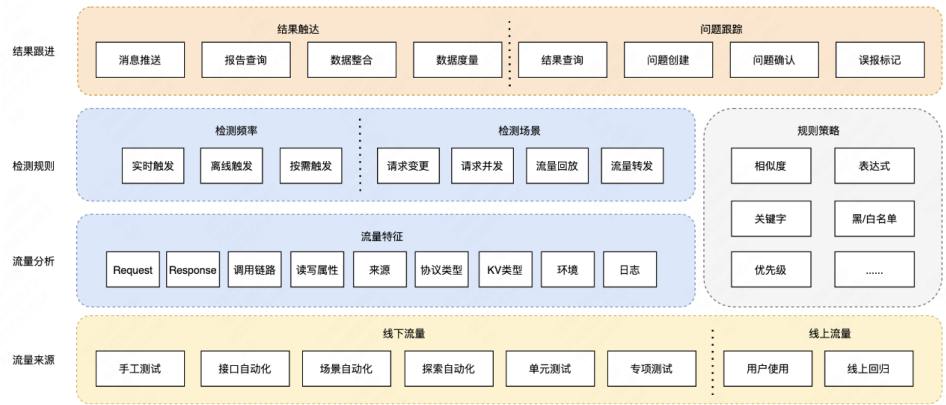
以上治理能力已对部门核心服务默认开启，低成本、自动化的实现了相关问题的常态治理及运营，并被接入到公司内多条业务线。

覆盖服务	覆盖接口	覆盖下游依赖	发现问题
500+	20000+	8000+	1000+

通过以上 3 个案例，我们可以看到共性能力和解法，因此后续的规划主要是建设通用基础设施，包含线上、线下以及不同来源的流量积累、流量分析，在其上进行模式挖掘、结果跟进和运营，在这样体系基础上，不断迭代底层能力，构建更丰富的可靠性治理场景。

未来展望

建设通用的基础设施，迭代流量分析和模式挖掘的底层能力，构建更丰富的可靠性治理场景。



5. Q&A

Q1：怎样预防配置异常造成的故障？

A：用相关事件举例，我们的一些配置平台包含一些规则，可以字符串形式或者一些类型形式存储，系统对这些配置的兼容性或表现，我们可以构造这些配置的异常场景，比

如它当前是一个数值类型的配置，那我们可以构造这个配置的异常值、边界值以及空值，比如它包含分隔符的字符串，我们可以用特殊分隔符以及特殊字符，构造异常配置的获取，验证一个配置的兼容性和可靠性的规则是当读取到这样的异常配置后，我们本来能正确回放的流量，在返回值、抛出异常、日志和调用链路层面有哪些相应表现。很多线上配置变更，因为人为原因和系统默认规则没有兜住的情况下，会引入这样的异常配置健壮性验证，有比较好的保障。

Q2：越权场景检查，链路对比是指走过的链路对比吗？还是每个调用点数据对比？

A：对于越权场景，我们可以识别到它在哪个环节进行了鉴权相关调用，不同的鉴权体系，有对应的权限相关服务和基础接口，构造有权限和没权限的场景后，它会识别没权限后的链路调用变化，比如链路节点数量以及哪个节点返回可以发现大部分问题，如果在这层没有识别到是否做鉴权，我们会关注每个节点的请求和返回，通过其他维度信息增强发现的有效性，降低噪声。

Q3：怎么自动构造接口里没有权限的用户？

A：在原始流量里，我们可以识别到它当前的鉴权方式以及它使用了哪些鉴权服务。这样，我们可以基于这个鉴权方式和服务构造有权限或者没有权限的用户。

Q4：可靠性治理系统是自研的，还是开源的，研发工作量多少人月？

A：可靠性系统建设的思路，目前是自研，它基于美团的基础设施和能力，展开上层解决可靠性问题的实践；研发工作量其实还好，它的点在于我们能找到哪些可以进行治理，快速迭代相关能力，并且在这一过程中不断补全新能力加进去，因此它是一个持续的过程，整体规划上，我们会考虑可靠性，从服务和代码的每一层，从机器、资源、基础组件到服务自身、上游流量和网关分层，拆分不同节点，构建不同策略，这样会有一个整体投入。

Q5：流量限流和服务降级如何实现？

A：美团有服务限流和降级能力的基础设施 Rhino 平台，服务降级是研发根据当前依赖等级，结合具体业务分析它是否是一个可降级的依赖，再配置对应的熔断策略，当降级时，是绕过当前故障进行降级还是在故障恢复后 Fallback，这样的相关规则和策略都可以配置化，自动化验证这些策略是否生效、是否符合预期。

Q6：在有了这些能力基础上，基于模式的可靠性治理用例占比多少？价值怎样评价？

A：我们想基于模式找到一个通用的治理策略和能力，这样的话，我们就可以将线上、线下海量流量数据都应用到这里，不需要 QA 同学投入成本，编写对应的用例，对于研发来说，我们希望直接确认和解决一些已识别到的问题，只需要花费问题确认和修复的成本。对于可识别用例占比，因为它是基于全量流量，所以随着时间的积累，历史场景以及新场景会相应覆盖到，它的用例有多少，取决于流量池以及流量质量和代表性。

Q7：流量回放 Mock，使用字节码，还是沙箱模式？

A：这里用到美团的基础设施能力，它在采集过程中，基于字节码增强采集，回放能力也是使用到了同样的能力。

运维

超大规模数据库集群保稳系列之一：高可用系统

作者：张洪 李军 运洋 翊睿

本文整理自主题分享《美团数据库的高可用系统》，系超大规模数据库集群保稳系列的第一篇文章。对数据库而言，非常核心的就是如何保证其高可用性。本文围绕 4 个方面的内容展开，包括高可用简介、高可用部署、重点模块的设计思考以及对未来思考。希望能够对大家有所帮助或启发。

| **B 站视频**：美团数据库高可用系统

0. 出品人说

在数据库集群规模迅速扩大的背景下，如果出现故障，如何快速恢复成百甚至数千个集群的数据和服务，是很多大型互联网企业面临的重要挑战。线上部署了几十万的微服务，数据库结构和拓扑随时在发生变更，系统重构、内核升级、硬件设备汰换、机房搬迁等等，也都会对数据库的稳定工作产生一定的影响。作为整个 IT 系统中最为重要、最为底层的服务，即便遇到了极小概率事件的冲击，也会造成非常大的影响。对美团数据库团队来说，”低垂的果实已经摘完”，我们开始着力应对这些小概率事件对业务造成的冲击。

数据库稳定性保障的破局之道：一方面是提升平均无故障间隔（MTTF），另一方面是提升应急响应能力，即缩短平均修复时间（MTTR）。在这个两个目标的指引下，美团数据库团队从能力驱动和故障驱动两个维度来构造整个稳定性保障的闭环体系。

从能力驱动的角度，我们借鉴了 Google 的稳定性保障体系。在最底部的三层，通过

故障演练 / 预案建设、复盘、可观测性的维度，思考怎么缩短故障处理时长；中间四层更多的是围绕研发需求、设计、上线、变更管控来降低故障的发生概率；顶层是产品运营，即通过面向内部用户的运营，指导业务对数据库进行选型和合理的使用，不断提升产品和平台易用性，并针对业务特点提供相应的解决方案。

从故障驱动的角度来说，包含事前预防和发现，事中故障定位，事后恢复、复盘和改进等等。从事前、事中、事后的全生命周期以及软件开发的各个阶段，全面提升管控和应急响应能力。

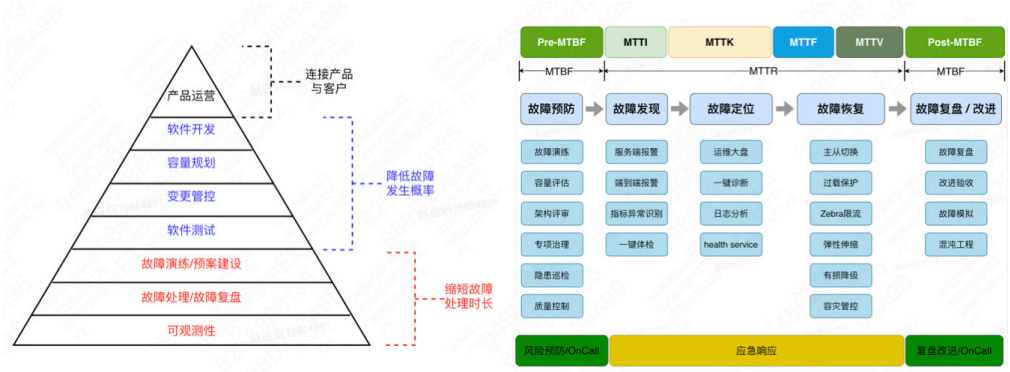


图 1　数据库稳定性保障的破局之道

基于过去多年保稳定方面的实践，本次沙龙将从如何提升进攻和防守能力，如何提升快速恢复能力，以及在进攻、防守、恢复形成闭环后，如何让人、系统、流程更好的协同和应对大规模故障几个方面，围绕数据库的高可用系统、数据库攻防演练建设实践、数据库容灾体系建设、数据库自治服务平台建设等 4 个议题进行介绍。希望能给从广大数据库从业者、业务研发人员带来启发和帮助。

1. 高可用简介

1.1 面临的挑战

首先分享下美团数据库高可用面临的问题和挑战，主要从 3 个层面进行展开：

第一个挑战是实例增长越来越快。下图 1 截取了 2019 年 1 月到 2022 年 1 月的数据，可以明显地看到实例规模的增长非常迅速，在大规模场景下，如何保证每一个实例的高可用性是一个非常大的挑战。大家都知道，保障几台机器稳定运行，跟保障几万台甚至

几十万台机器的稳定运行，其复杂度完全不在一个量级。

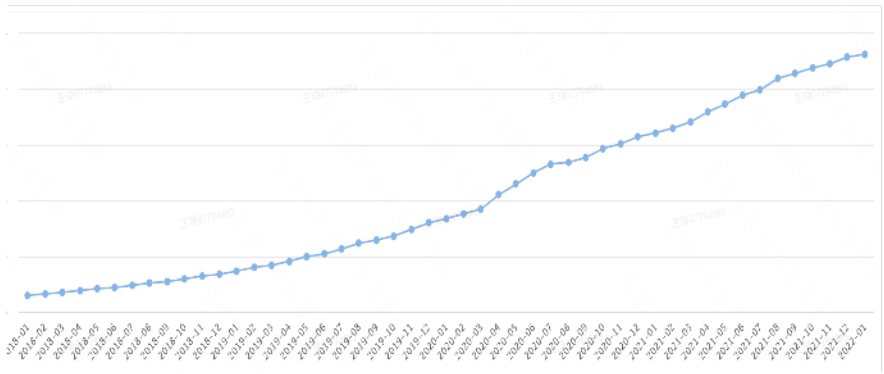


图 2　实例增长曲线

第二个挑战是可用性（RTO）要求越来越严。美团业务类型偏在线实时交易，对系统可用性有非常高的要求，特别是即时配送要求更高。在业务发展的早期阶段，体量并发也不高，对系统可用性要求可能只有 99.9%。但是随着业务体量快速增长，对系统可用性的要求就会不断增加，特别是比较偏底层的数据库系统，从 99.9% 到 99.99% 甚至更高。

系统可用性	宕机时间/年	宕机时间/月	宕机时间/周	宕机时间/日
99.9%（3个9）	8.76小时	43.8分	10.1分	1.44分
99.99%（4个9）	52.56分	4.38分	1.01分	8.66秒
99.999%（5个9）	5.26分	25.9秒	6.05秒	0.87秒

图 3　系统可用性对应的宕机时间

第三个挑战是容灾场景的复杂性。容灾场景主要分成三个层面，第一个是常规容灾，比如日常软件、硬件或者网络故障；第二个是 AZ 容灾，即机房层面，如机房断网、机房宕机等；第三个是 Region 容灾，即更大空间容灾，典型的是城市级容灾，目前主要还在解决 AZ 级容灾，分如下五个阶段：

	定义	PaaS容灾能力要求	生产环境容灾演练方案和达标要求
第0阶段	PaaS不具备单点失效的容灾能力	无要求	不要求演练
第1阶段	PaaS具备单点失效的容灾能力	单个AZ内数据面和控制面（包括运管平台）任意一个实例失效时RTO/RPO达标，失效实例恢复时不影响业务可用性[1]	单个AZ内，使用演练工具向特定PaaS任意一个实例（容器或VM或物理机）注入故障演练验证RTO/RPO，停止注入故障时验证业务可用性，所有实例都需要达标
第2阶段	PaaS具备AZ内所有实例（容器或VM或物理机）同时失效的容灾能力	单个AZ内数据面和控制面所有实例同时失效时RTO/RPO达标，失效实例恢复时不影响业务可用性	多集群模式PaaS会先演练验收单个AZ内任意一个集群的所有实例同时失效时的RTO/RPO，所有PaaS（不区分是否为多集群模式）最终都需要演练验收单个AZ内特定PaaS所有实例同时失效时的RTO/RPO，停止注入故障时都会验证业务可用性，所有实例都需要达标。
第3阶段	AZ内所有PaaS所有实例（容器或VM或物理机）同时失效的容灾能力	单个AZ内所有PaaS所有实例同时失效时RTO/RPO达标，包括特定PaaS组合所有实例同时失效的容灾能力，失效实例恢复时不影响业务可用性。	演练场景1：使用演练工具向单个AZ内特定PaaS所有实例和下游依赖PaaS的所有实例同时注入故障演练验证RTO/RPO，停止注入故障时验证业务可用性，所有实例都需要达标 演练场景2：使用演练工具向单个AZ任意一个业务Appkey使用的PaaS所有实例同时注入故障演练验证RTO/RPO，停止注入故障时验证业务可用性，所有业务Appkey使用的PaaS实例都需要达标 演练场景3：使用演练工具对单个AZ所有PaaS所有实例同时注入故障验证RTO/RPO，停止注入故障时验证业务可用性，所有实例都需要达标。 （备注：3个场景都需要演练验证通过。PaaS如有其他组合场景也需要演练验证通过）
第4阶段	AZ出入口网络同时断开的容灾能力	单个AZ内所有PaaS的RTO/RPO达标，AZ恢复时不影响业务可用性	断开机房出入口网络验证PaaS的RTO/RPO，恢复AZ出入口网络时验证业务可用性，所有PaaS都需要达标

图 4　AZ 容灾能力等级

从图 4 可以看到，我们将 AZ 容灾分设第 0 至第 4 共 5 个阶段，简称 L0–L4。随着等级的提高，场景越来越复杂，相应的规模也越大。从容灾规模维度看，单点 –> 单个集群 –> 某个业务依赖的集群 –>AZ 内的集群，不同规模要求的能力是完全不一样的，除了规模之外还有容灾的场景也会在变化。

- “L0–L1”这两个等级侧重面向常规容灾，是实例级容灾。
- “L2–L3”这两个等级侧重面向 AZ 容灾，相比 L1 有非常大的跨越，因为既要解决“L0–L1”面临的常规容灾问题，还要解决一个很核心的问题，即整高可用自身是否能够快速恢复，以及高可用依赖的下游服务是否具备容灾切换能力。由于高可用本身是一个系统，它有数据面和控制面，有上下游依赖，所以先保证自己是可用的，才能保证数据库的 RTO 和 RPO。
- L4，从 L3 到 L4 又有一个很大的跨越，因为 L3 是的规模是相对可控的，而 L4 直接是断 AZ 的网络，AZ 的大小不同，它涉及更大场景是更真实的 AZ 容灾。

1.2 发展历程

接下来，分享一下美团高可用系统发展历程。总的来说，美团的高可用发展历程是根据不同的阶段的矛盾和挑战，做出的相应解决策略和方案。到目前为止，有三次比较大的系统架构迭代：

- 第一代架构是 2015 年之前，称为 MMM（Multi-Master Replication Manager for MySQL），该架构包括接入层 VIP、Agent 和 Manager，这架构本身存在很多问题，比如 VIP 接入无法支持跨机房、跨网段；Agent 和实例绑死，本身也没有高可用，维护起来比较困难；Manager 还是单点。
- 第二代架构是 2015-2019 年，称为 MMHA（Meituan Master High Availability），是在 MHA 的基础上结合美团数据库生态定制的架构，解决了第一代架构中 VIP 接入和 Agent 误判的一些问题，但在 2019 年以后，由于实例规模变得很大且增长迅速，该架构管理起来异常复杂，Manager 是单点且本身没有高可用。同时，2019 年整个 PAAS 在演练机房故障，所以当时这个架构逐渐暴露出各种稳定性问题。
- 第三代架构是 2019 年，新开了基于 Orchestrator 定制的高可用系统，实际开始时间是 2018 年，2019 年开始灰度。当时 MGR 在业界已经有些公司在使用，MGR 通过分布式协议来解决高可用难题，我们对此进行了深入的思考：我们是否能引入一个类似分布式协议解决高可用系统面临的问题，本质上就是把 MGR 中内置的分布式协议放到高可用系统实现。

后来根据这个想法落地的架构就是当前的高可用系统，是一个 Raft Group，多节点、多机房部署。以前主要关注主从切换，但新系统类似八爪鱼，所有在集群里的节点都是统一托管，如主库、从库、Ripple（后续有说明）等；另一方面，它是一个高可用、高性能、大规模并发处理架构，我们部署了多个 Raft Group 分组来托管不同地域和服务等级数据库集群。除此之外，我们也正在思考新一代去中心化的新架构，将在文章的最后一个章节进行介绍。

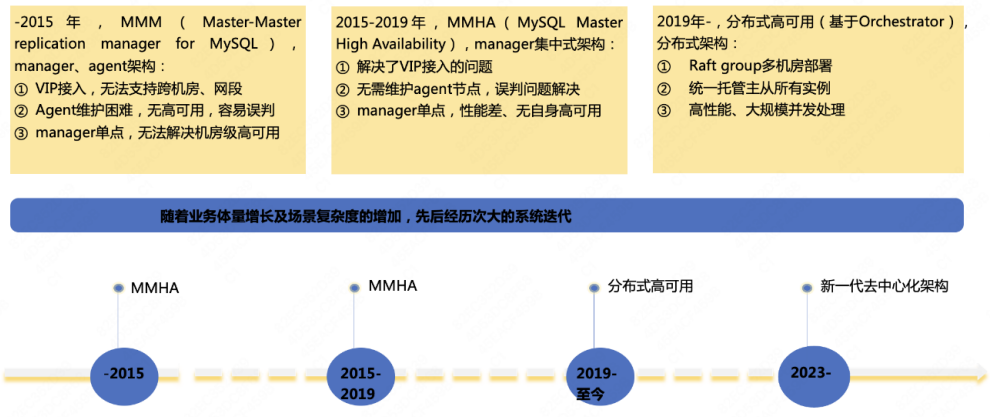


图 5　美团高可用架构的发展历程

2. 高可用部署

2.1 高可用架构（数据流、控制流）

这部分主要分两条线：控制流和数据流。

- 数据流：如果业务应用要访问数据库，它是如何拿到数据的，一个 SQL 过来是如何把数据返回回去的，业务应用通过访问中间件看到 MySQL 的拓扑，数据流比较简单，也是业界比较通用的做法。
- 控制流：业务应用要通过访问中间件来访问到正常数据库拓扑，需要用高可用组件来解决故障转移难题。

由下图 6 可看出，高可用组件分四个部分：HA Core 是经典的 3 节点 Raft Group 部署（也可以是多节点部署）；HA 平台是管控系统，如主动切换、状态机查看、兜底等；HAservice 是 HA 的 API 服务，负责跟外围系统交付；调度系统包括（Scheduler、Worker），是一个流程调度服务，处理状态机的轮转。元数据存储是外围依赖的核心服务，负责基础配置的管理，当配置变更后会通过它下发到所有的中间件节点，保证业务看到的是正常的数据库拓扑。

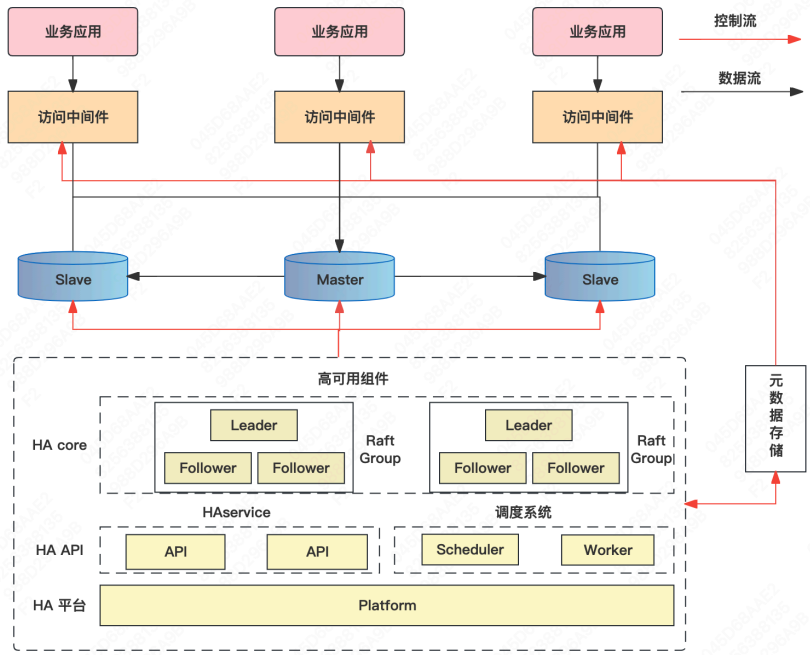


图 6　高可用架构图

2.2 高可用部署（HA Core、微服务、数据层）

下面我们围绕 4 个高可用组件来展开介绍一下，高可用部署在应对 AZ 级容灾或 Region 级容灾的策略。

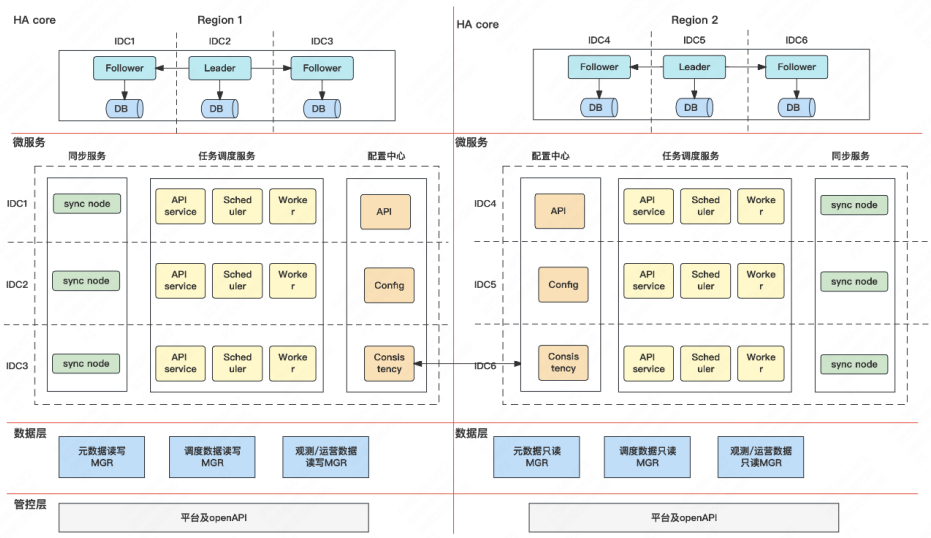


图 7　高可用部署架构

一是**HA Core 部署**，它有三个特征：一是多 Region，如下图 6 所示，以红色线为分界，左边是 Region 1，右边是 Region 2；二是多 AZ，每一个 HA Core 是 3AZ 部署；三是多集群，每个 HA Core 的 3 节点集群会托管 MySQL 千级别实例。

二是**微服务**，有同步服务、调度服务和配置中心。

- 同步服务：简单来说，我们的工程师在 RDS 申请集群或 DB 之后的信息会全部同步注册到 HA Core 服务里面去，相当于 HA Core 是一个“八爪鱼”，它能发现这些信息。
- 任务调度：它也包括 API Service、Scheduler 和 Worker，主要做状态机任务执行。这两个服务都是多 Region、多机房部署，它们本身没有状态。
- 配置中心：业务应用访问中间件和高可用之间的数据同步纽带，是 MySQL 的节点发现和处理的核心组件，是双 Region 部署，有自己的组件，比如有 API、Config 和 Consistency。

三是**数据层**，微服务除了配置中心，其他服务没有状态，因为状态都在数据层，数据层服务都是 MGR 集群，而且这些集群各自作用不一样，MGR 是单 Region 写、多 Region 读，所以这里我们回到最开始说的 Region 容灾，这些数据需要做单元化或者隔离，现在这些集群还是单 Region 写的。

四是**管控层**，它是正常的平台服务，常规化的部署。

3. 重点模块设计

3.1 故障发现（减少漏判、降低误判）

故障发现有两个核心指标：

- 第一个指标是不要漏判，如果故障没有判断，那 RTO 其实根本就不生效。
- 第二个指标是降低误判，因为我们知道 RTO 不可能等于 0，即 RTO 一定对业务有影响，如果一天误判几次，那对业务是无法接受的，对业务是有损的。

先说漏判，如下图 8 所示的故障探测通道，每个探测通道完全独立，如普通探测通道，心跳探测通道、从库探测等完全独立决策，谁先发现都可以作为决策依据，现阶段这些通道都是在服务端主动并发发起探测。另外一方面就是业务端，访问中间件有业务端的

报错，也纳入故障探测和决策，这个是业务视角的故障，不管什么问题只要业务达到有损阈值就可以判断决策，这部分还在建设中。综合多通道、以及客户端和服务端判定，哪一个通道发现了问题，直接开始做决策，不受其他通道影响，这是解决漏判问题的策略。

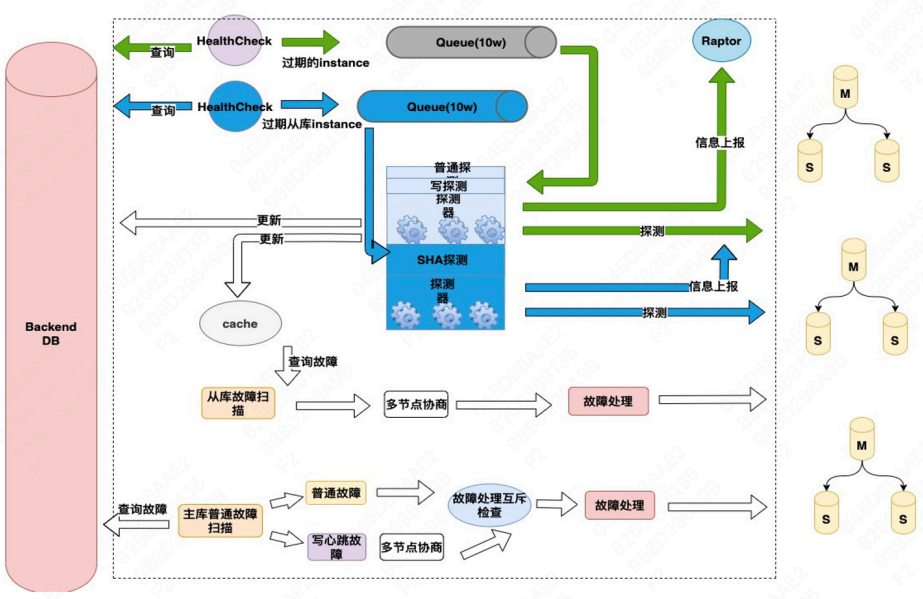


图 8　故障发现

再说减少误判，我们引入了多节点协商机制，比如我们遇到某些问题会引入“民主”决策让大家发挥更多意见，最终汇总得出结论。HA Core 是多节点，当每个节点独立探测，探测后也会独立去做判断和分析，如只有一个 Follower 节点认为故障是没用的，还需要 Leader 的分析并向其他 Follower 节点发起协商共同决策，只有大多数节点认为故障才会把它确定为故障，会注册故障并进行故障处理流程。图中的 Backend DB 是 Raft Group 每个节点各自独立的存储，各自探测的拓扑以及判定决策信息等都存在本地。

3.2 故障选举（选举因子、选举策略）

所谓故障选举一定是多个从库，一主一从不存在选举。美团的 MySQL 集群现状是 1 主多从，所以选举异常复杂，因为我们要保证容灾 N+1、多 AZ 甚至多 Region 部署，选举时选谁做主库就非常重要，主要有两个影响因素：选举因子和选举策略。选举因子 + 选举策略 = 决定谁是新主。

(1) 选举因子是影响选举的核心要素，一次故障有 20 多个选举因子会共同影响如何排序。

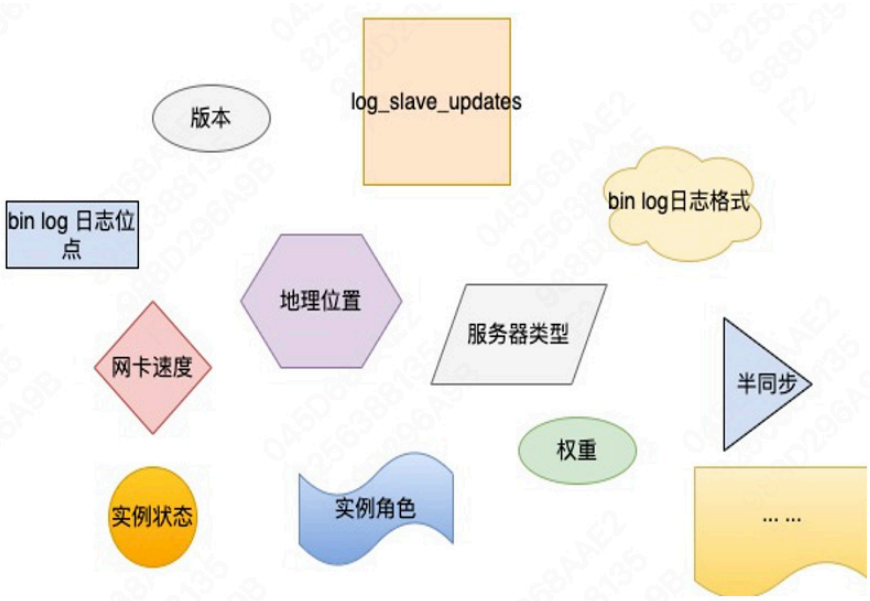


图 9　选举因子

举两个例子，第一个是版本，一主五从的集群可能有三个版本，如果让最新的版本作为新主就会有问题，要尽量不让它作新主，因为如果新主的版本是最新的，但从库都是比他落后的低版本，就会出现一些兼容性问题；第二个是权重，假设其他因子都一样，但存在运维和例行维护的情况下会人为说把权重降低，或标记一个实例不能作为新主，那么权重 100 的比权重 50 的有更大机会成为新主。每个选举因子都有权重，最终做综合排序，如版本、binlog 格式、权重、服务器配置等，综合选举因子的选举结果是 1M (first master)。

但 1M 并不一定是最优，有业务是北上跨 Region 部署，如之前主库在北京，按照选举因子排序选出的 1M 到上海去了，显然是业务无法接受的，因为老主库在北京说明绝大部分业务部署在北京，一旦让它跨 Region 写入到上海，那么 RT 会增加很多，所以引入了选举策略。

(2) 选举策略是同机房优先 > 同中心优先 > 同区域优先，是一个灵活的 fallback 策略。按照选举策略重新选举一个新主 (称为 2M)，如果 2M 和 1M 是重叠的，就认为这个

1M 是满足业务诉求的 Master，会将 1M 作为最终的主库。但有时候 1M 和 2M 的排序相差很大，这时我们尽量让那种没办法改变的因素以它为基础，把其它能改变的因素对齐，比如地域很难改变，但是要改位点等容易改变，最终权衡后选举出新主库。

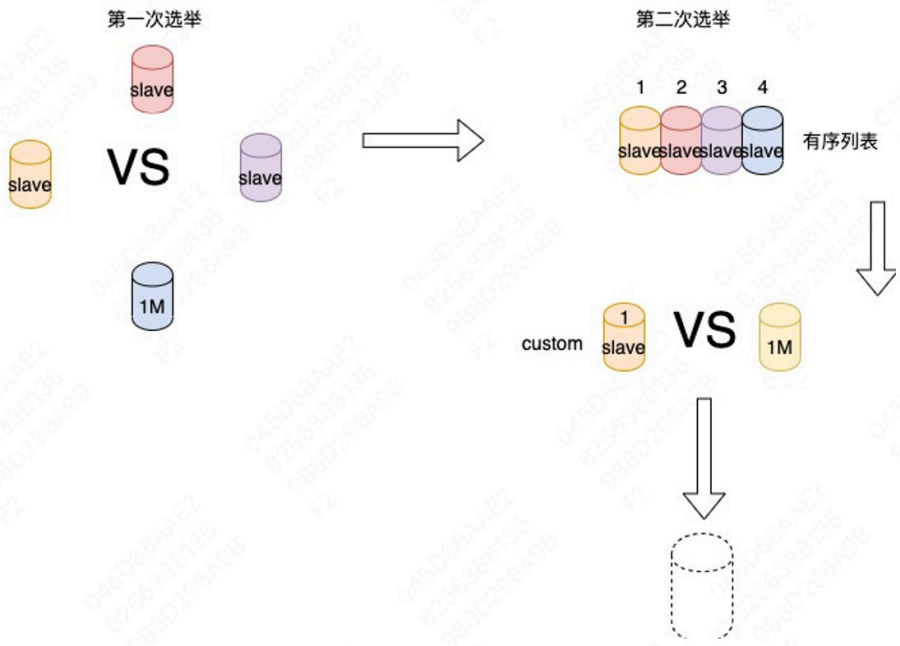


图 10　选举策略

举例：

- 如图 11（主库 M，从库 S1、S2、S3、S4 四个实例），S2 的选举规则 (promotion rule) 是一票否决的 Must Not，那它一定不能做主库，即使选不出来其他，它也做不了主库。
- S1 和老主库是同 AZ 的即都是 AZ1，S1 比 AZ2 的 S3 和 S4 有更高优先级，即更大的机会作为新主。
- S4 权重 100，S3 权重 90，S4 比 S3 权重更高，即使 S4 是独享容器，它也有更高的选举权。

实例	promotion rule	自定义权重	AZ	服务器类型	CPU核数	.....
M	/	/	AZ1	/	/	
S1	prefer	90	AZ1	独享容器	32	
S2	must not	100	AZ1	共享容器	16	
S3	prefer not	90	AZ2	物理机	48	
S4	prefer	100	AZ2	独享容器	48	

图 11　选举示例

这个例子说明最终选举出的主库是综合考虑后的排序，受各种因子影响。

3.3 数据一致性（四个风险及解决方案）

为什么要保证数据一致？在我们现在这种规模的业务场景下，可用性优先策略已经没办法覆盖所有的业务场景，但是在主从架构下面，数据丢失又无处不在，参考图 12 为例，数据丢失的风险点比较多：

- 问题 1，可能 binlog 未实时落盘。
- 问题 2，IO 线程没拿到最新数据。
- 问题 3，SQL 线程没有和 IO 线程对齐，包括从库之间 binlog 位点也不一样。
- 问题 4，从库不完整事务问题，在主库事务提交时它是完整的，但是它通过 IO 线程同步给从库时不是按事务粒度去同步而是按事件 event 粒度同步，如果事务未完整接收也可能会产生数据丢失不一致。

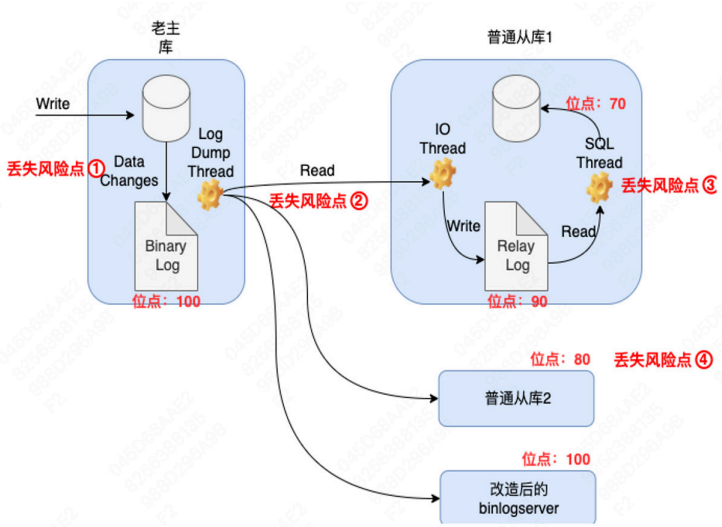


图 12　四个风险

- 针对问题 3 和问题 4，其实归纳一下就是说只要有从库拿到数据，不管是否对齐，我们是有策略能够保证一致性，这个策略叫 S1，只要数据同步过去了则就可以通过 S1 保证一致性。
- 针对问题 2，就是主库 Event，没有任何一个从库有获取全，这种情况必须解析老主库 binlog 以及计算 binlog 位点并获取到数据，并在切换中补齐数据，即在开放给业务写流量之前，会将数据给新主库补全保证一致性，这个策略叫 S2。
- 但 S1+S2 也不能保证 0RPO，服务器宕机时拿不到老主库 binlog，没有办法计算、解析和处理，不能保证 0RPO，大概有 20% 到 30% 的比例。

接下来，讲一下针对老主库宕机拿不到数据时的解决方案（这个叫策略 S3）。在此，也参考 MySQL 主从库架构通过 IO 线程和 SQL 线程去解决一致性的思路。第一个难点，IO 线程要尽量把数据拿全，这是非常关键的点，也是很难的事情；第二个难点，SQL 线程能够快速把数据应用下去，达到这两方面才能保证 0RPO。

第一个难点解决方案如图 13 所示，核心是 Ripple，它是一个高性能的 binlog 订阅服务器，作用类似 IO 线程，保证快速的获取 binlog。第一个特性是它的性能极高，因为普通从库 IO 线程除了处理 binlog 接收请求外，也处理很多比如写 Relay log、一些锁逻辑处理，以及配合 SQL 线程一致性的工作，而 Ripple 的任务就是快速将数据全部拿过来而不用处理其它逻辑，binlog 同步速度是普通 MySQL IO 线程的 3 倍以上；第

二个特性是可以配置强一致，支持半同步机制，如果你认为可以牺牲 RTO，比如可以牺牲 3 分钟但不能丢失数据，那么配置多副本的半同步超时策略（如 3 分钟），由于 Ripple 的性能高，实际超时时间远远小于这个值，退化概率非常低；第三个特性，它本身是一个存算分离的架构，即 Ripple 是轻量级容器 +EBS 云盘，用这样一种架构来保证数据完全。

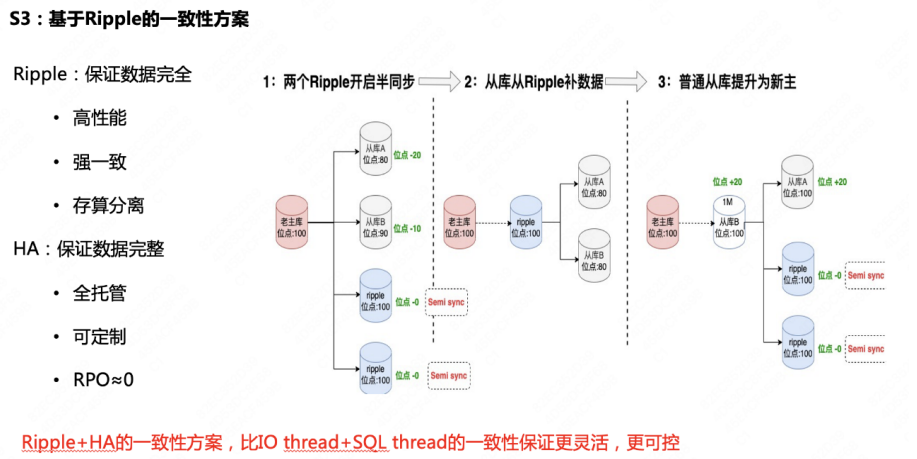


图 13　基于 Ripple 的一致性方案

第二个难点是事中从 Ripple 补数据，Ripple 毕竟不是普通的 MySQL，所以需要 HA 去兼容，但 HA 处理数据比 SQL 线程灵活很多，因为 SQL 线程在需要实时处理数据来保证一致性，而 HA 只有在故障期间才处理数据，可以拉长 RTO 做很多定制。还有这里提到的 RPO ≈ 0，为什么是近似，因为虽然可以做到 RPO=0，但要接受 RTO 的损耗。如果你既要 RTO，又要 RPO，那么可以在策略灵活配置。

在目前业务规模下没办法兼得 RTO 和 RPO，如果想要 RTO 就要牺牲一部分 RPO 或者反过来。本身里面有一些不可控因素会影响核心指标，比如事务的大小、延迟、完整性等，我们将这些配置通过策略开放给业务，让业务自己去决策。举个例子，如果你的业务有很多大事务，但你又要想 0RPO，那么你就需要配置更长的 RTO 损耗。

- 可用性优先：可根据业务特性自定义，承诺 RTO 不保证 RPO。
- 一致性优先：可根据业务特性自定义，保证 RPO 但不会无限制，RTO 会控制上限。

- 不可控因素：事务大小、延迟、事务完整性等。

分类	策略套餐	容灾能力水平	总耗时上限	S1：策略一		S2：策略二		S3：策略三
				S1耗时上限	是否开启	事务数量上限	S2耗时上限	S3耗时上限
可用性优先	通用-A1	优先保证RTO	-	-	-	-	-	-
	业务模版-A	取决于实际配置项	可配置	可配置	开启 / 关闭	可配置	可配置	可配置
一致性优先	通用-C1	优先保证RPO	-	-	-	-	-	-
	业务模版-C	取决于实际配置项	可配置	可配置	开启 / 关闭	可配置	可配置	可配置

图 14　可用性优先 / 一致性优先

3.4 多机房高可用

常规场景，Leader 故障后会在 4 秒之内快速选举一个新 Leader 出来继续工作，但也有一些场景，如下图 14 所示，MySQL Master 和 HA 的 Leader 都在 AZ1，如果 AZ1 宕机之后怎么办？其实就出现右边这个图。老 Leader 在处理 AZ1 里的 MySQL 节点故障的同时，由于自身 Leader 也在 AZ1，会中断老 Leader 的处理状态机并选举一个 Leader，但新 Leader 并不知道老 Leader 状态机如何处理，就直接导致切换的失败。

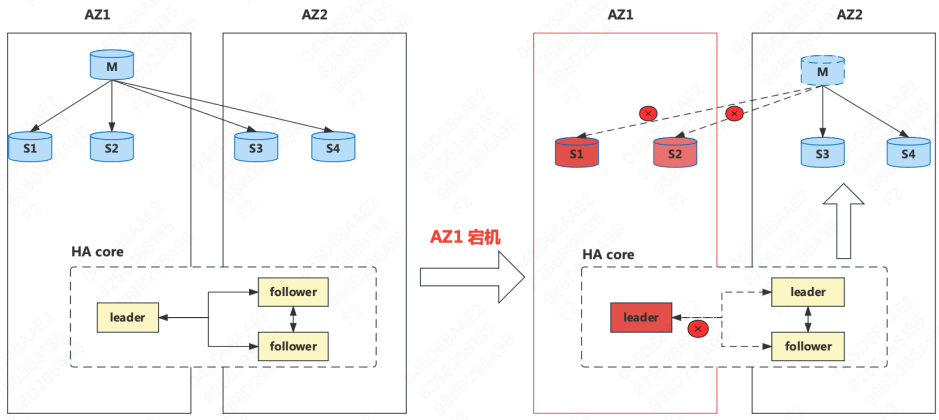


图 15　多机房高可用

多机房高可用的的核心解决思路是保证各个节点之间状态机的连续性。所谓状态机的连

续性，就是 Leader 实时把状态机同步到所有 Follower 节点，一旦 Follower 节点重新变为 Leader 点后，它会继续老 Leader 的状态机，会有动作的临界状态判断。如下图 15 所示，它根据执行的代价判断是回滚还是继续执行，如果回滚，就会把老 Leader 状态机全部回滚之后从零开始处理；另外一方面，如果由于集群的拓扑和状态被破坏了，导致回滚状态机比较麻烦或回滚不了，那么就会继续执行老 Leader 状态机没有执行完的动作。

保证多节点状态机连续性：

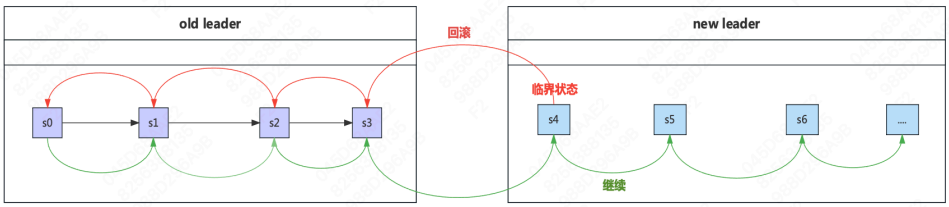


图 16　保证各个节点之间状态机的连续性

- 状态同步：Leader 实时将状态机通过 Raft 同步到所有的 Follower 节点
- 临界状态：根据执行代价确定的状态机临界点
- 回滚：新 Leader 执行状态机回滚，包括对应的操作
- 继续：新 Leader 继续执行老 Leader 的状态机

3.5 配置下发（双 Region 下发）

由于配置服务是双 Region 部署，分为同 Region 下发和跨 Region 下发。同 Region 下发会写到存储层，而 config-server 会更新最新配置，并将配置并推送到客户端。跨 Region 下发，比如说北京、上海和深圳都有业务服务节点，最终把它推送下去的时候也会走一致性服务（Consistency-Server），即一旦某个 Region 更新后，我们会把数据推送到另一个 Region 里，Region 之间的数据完全一致，如果另外一个的 Region 有业务服务节点，就会继续走同 Region 下发流程。

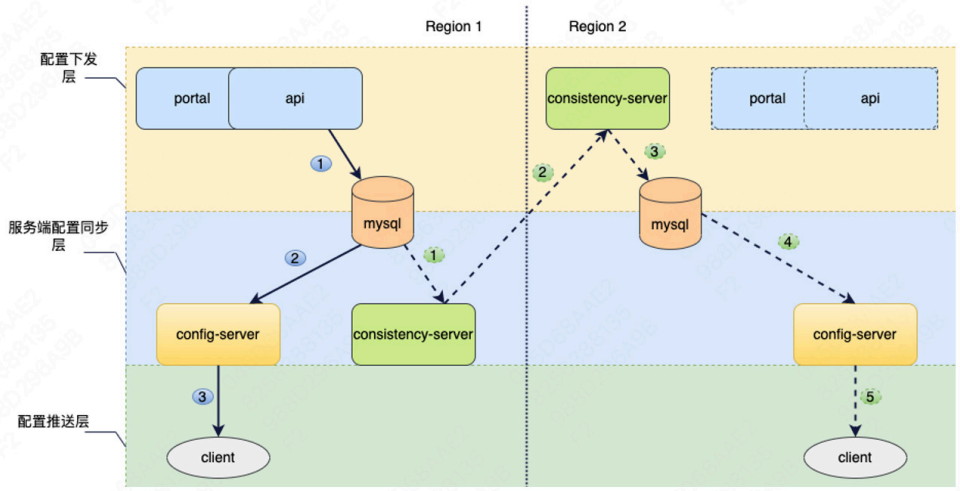


图 17　双 Region 下发

4. 未来思考

最后，分享一下对高可用未来的一些思考，主要包括以下三个方面：

- 提升容灾能力，主要是 AZ 级容灾和 Region 级容灾。这两方面我们还在建设中，AZ 级容灾需要减少依赖，不能减少的需要 AZ 级闭环独立部署，以及提升大规模并发处理的能力等；Region 级容灾方面，我们在做一些单元化的思考和方案，尽量让包括数据层等所有服务闭环，不要跨区域访问。
- 去中心化架构。业界也有数据库把高可用内置到 MySQL，如 MGR，内置到 MySQL 有非常多的有优势，但对于美团一主多从架构是现状前提。另外，MGR 架构对网络抖动的容忍度较低，以及对请求延时有一些增加，导致大部分业务场景没法接受。所以，我们在做另外一种思路，即把高可用内置到 Proxy 进程，让 Proxy 自带数据库高可用的能力，跟内置到 MySQL 的思路类似。
- 去依赖化、集群化。将 HA 的 Service、Scheduler 和 Worker 及配置中心等下游依赖去掉，希望内置到 Proxy 进程后，内部数据通过 Raft/Gossip 协议同步而不再依赖中心化服务，让它完全做集群化，这也是我们 2023 年在思考的策略。

超大规模数据库集群保稳系列之二：数据库攻防演练建设实践

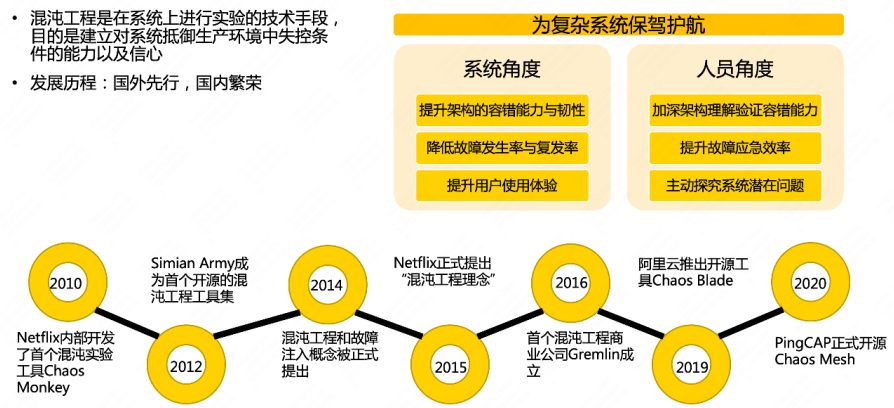
作者：占全

1. 背景

1.1 初识混沌工程

首先我们先了解一下什么是混沌工程？简单而言，混沌工程是在系统上进行实验的技术手段，目的是建立对系统抵御生产环境中失控条件的能力以及信心。这主要体现在两个方面，从系统角度来讲，混沌工程可以提升我们架构的容错能力和韧性，降低故障发生率和复发率，提升系统用户在使用时的体验；从人员角度来讲，通过混沌工程我们可以加深对架构的理解，在处理故障时能提高整体的应急效率，并且能主动去探究系统中潜在的一些问题。

混沌工程最早可以追溯到 2008 年，当时奈飞公司（Netflix）因数据库故障造成了一次巨大的损失，于是他们便围绕着稳定性体系做出了一些探索和建设。直到 2015 年，混沌工程的理念才被奈飞正式的提出。2019 年，阿里云推出了开源工具 Chaos Blade；2020 年，PingCAP 开源了 Chaos Mesh，这也标志着国内在混沌工程领域也有了成熟的产品。



来源：中国信息通信研究院，2021 年

1.2 现状

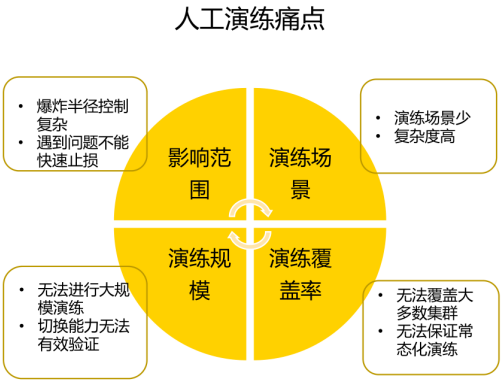
下面是当前美团数据库运维的一些现状，这里从五个维度展开：首先集群规模（包括集群的大小和数量）的线性增长；第二，集群访问量的持续增加，当然这也是美团业务增长的必然结果；第三，线上遇到的故障种类也越来越多；第四，单位时间故障的影响也越来越大；第五，一些小概率事件的发生也随着集群规模的扩大成为了可能。





1.3 痛点 & 作用

基于以上的问题，我们整体上对于数据库集群的稳定性要求也越来越高。所以，围绕着稳定性建设，数据库团队在故障的预防、发现、分析、恢复以及事后复盘 5 个方面做了很多工作。其中在故障预防方面，我们会通过故障演练探索不同故障场景对我们业务的影响，提升故障发生时业务系统整体的容错能力。早期的时候，我们通过人工的方式来做故障演练，但人工演练在以下四个方面存在很大的问题：





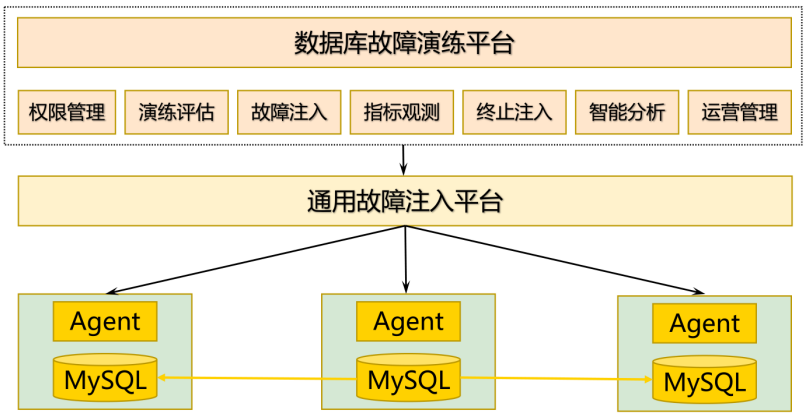
1. 在演练场景方面，人工演练能演练的场景比较少，演练场景的复杂度也比较高；
2. 在演练覆盖率方面，人工演练无法覆盖大多数的集群，也就无法保证常态化的故障演练；
3. 在演练规模方面，人工演练没有办法进行大规模演练，在遇到一些机房或者交换机级的故障时，切换能力无法得到有效验证；
4. 在影响范围方面，人工演练在整个演练过程中不能很好地控制爆炸半径，遇到问题不能快速止损。

基于人工演练的这些痛点问题，我们设计并建设了数据库故障演练平台，这个平台的作用主要体现在以下四个方面：第一，验证故障发生时组件的防守能力；第二，通过数据库大规模容灾演练，验证数据库集群的容灾能力；第三，可以验证故障发生时相关预案（业务、DBA）的有效性；第四，可以预知故障对业务造成的各种影响。

2. 建设实践

2.1 体系架构

当前，数据库故障演练平台体系架构主要包括如下六个模块：



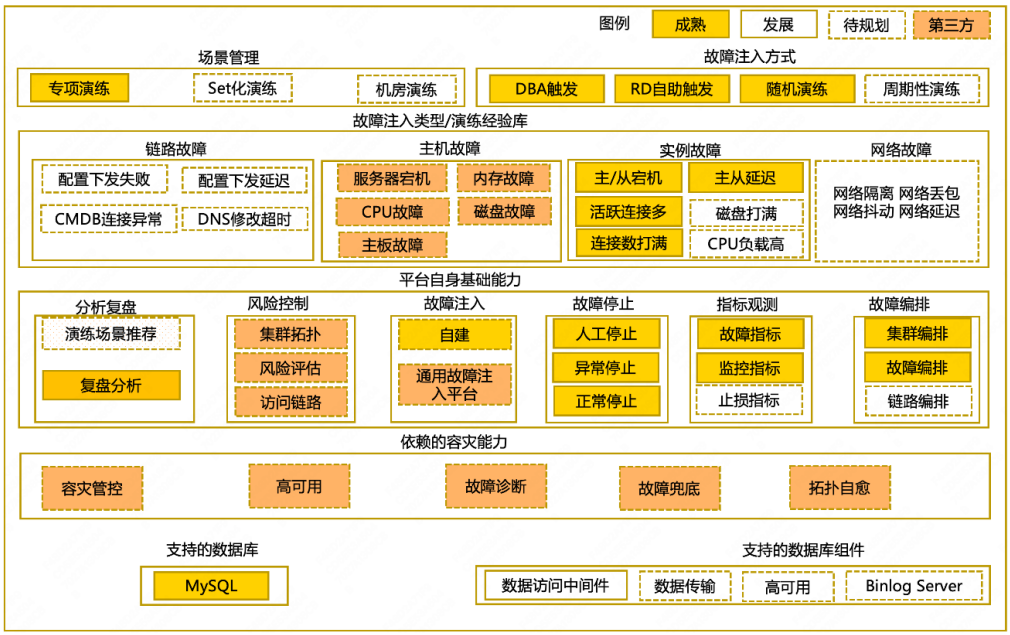
1. **权限管理**，控制演练发起人，可以对哪些集群发起故障演练；
2. **演练评估**，演练之前为了避免演练风险，会对集群做一些检查，比如集群的高可用是否开启，拓扑是否符合要求，演练时间是否处在集群访问的高峰期等；
3. **故障注入**，在相关的检查通过后，真正去执行故障的下发；

4. **指标观测**，在整个演练期间会通过采集一些可以反映稳态的数据，比如以业务的访问成功率、业务的响应延迟等等，来分析故障演练对业务造成的影响；
5. **终止注入**，在演练过程中发现整体影响比较大，或者有一些非可控的影响，可以立刻终止故障演练，避免故障演练对线上业务造成的影响进一步扩大；
6. **智能分析**，在演练结束之后，可以通过采集一些反应故障注入或者是防守组件的相关数据判定本次故障演练是否符合预期。

当前故障注入支持的主机类型比较全面，可以在物理机、虚拟机、容器上做故障演练。

2.2 能力全景

这部分主要介绍一下我们故障演练平台的能力全景，它主要规划了我们当前数据库故障演练平台的一些核心能力，整体的细节如下图所示：



- 从支持的数据库来看，当前我们主要支持 MySQL 的故障演练；
- 支持的数据库组件有数据库访问中间件、数据传输、高可用系统和 Binlog Server，我们会在后续的迭代中逐步支持，当前我们正在接入数据库访问中间件的一些演练场景；
- 在整个故障演练过程中，我们会依赖高可用、容灾管控、故障诊断、故障兜底以

及拓扑自愈等多个方面的能力，主要是验证上面提到的这些组件在整个演练过程中的表现是否符合预期；

- 当前，平台自身已经形成了从故障编排、故障注入、指标观测、故障终止、分析复盘的一个闭环。

2.3 故障注入能力

关于故障注入能力，当前支持的故障场景主要包括 5 个（如下图所示），目前我们主要建设的是宕机类（包括主库宕机、从库宕机）、主从延迟以及从库活跃连接多，由于当前线上 80% 的故障都是由这几个场景引起的，这也是我们目前没有建设过多故障场景的主要原因。后面我们还会规划建设高可用的链路故障，比如数据库和高可用同时出问题时，高可用的表现是怎么样的，是否能够顺利的进行切换。

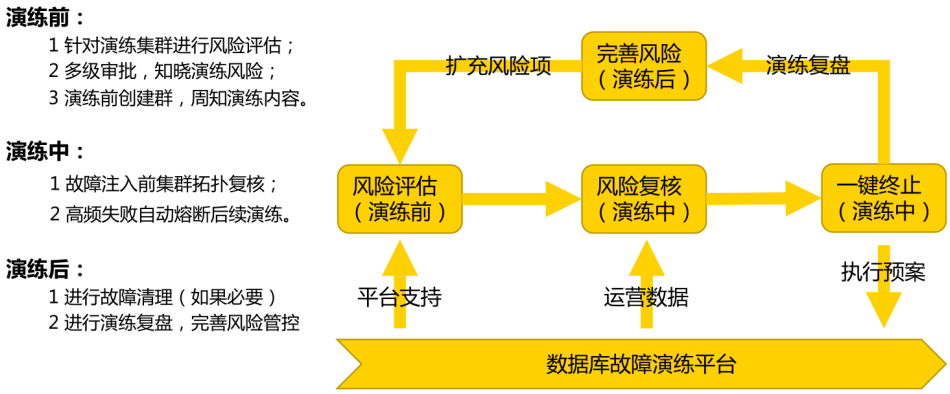


在故障编排方面，我们通过建立一个任务内的子任务间的串行故障和并行故障的编排机制，来实现演练任务的统一编排。

在并发能力上，当前故障演练平台保守的并发故障注入能力是在一分钟实现 5000+ 节点的故障注入，目前这个能力已经可以满足我们对于大规模故障演练的整体需求。

2.4 演练流程

整个演练流程主要分成三部分，演练前、演练中和演练后，同时我们也形成了风险评估、风险复核、一键终止、故障复盘的能力闭环。



在演练前，我们主要做三个方面的工作：

- 针对演练集群做风险评估，比如演练集群的高可用是否开启、拓扑是否符合要求；
- 利用多级审批机制，让业务和相关的 DBA 知晓整个演练的风险；
- 在演练前创建相关交流群，把故障演练相关的信息在群里进行周知。

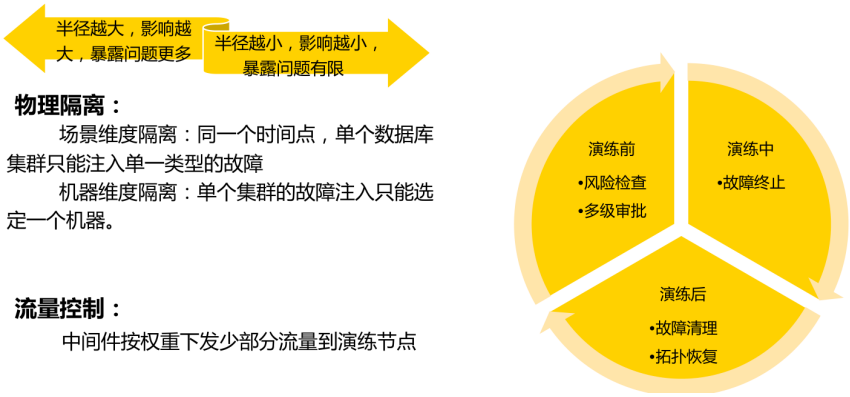
在演练中，有两方面的工作：

- 在故障注入前，做一下前置检查，主要目的是在建立演练任务后和故障注入前这个时间段，防止因集群拓扑变化产生风险，比如演练的场景是从库宕机，这种情况下如果发生切换，故障注入的节点很有可能变成了主库；
- 如果一个任务里含有多个子任务，而某一个子任务发生了故障注入失败，那么后续演练我们会自动终止。

在演练结束后，我们会进行故障清理，完成后做演练的数据收集，用于复盘，并据此完善相关的风险管控流程。

2.5 爆炸半径控制

提到混沌工程，就不得不提到爆炸半径控制的问题，当前我们主要从两个维度、三个方面做了一些控制。



第一个维度是物理隔离，它包含两方面：

- 场景维度隔离，即在同一时间点、单个数据库的集群只能注入单一类型的故障，主要是为了避免在单个演练任务中如果针对一个数据库集群注入多种故障，整体影响不容易被评估。
- 机器维度隔离，单个集群故障注入只能选定一个特定机器，而不能选择多个机器。

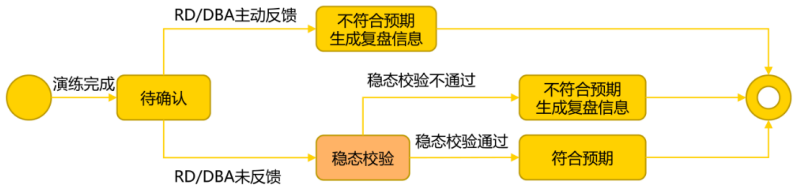
第二个维度是流量控制，主要是通过中间件按权重下发少部分流量到演练节点实现，流量控制能从一个更精细的维度来控制演练的风险。

另外，我们在演练前、演练中和演练后都有相关的检查或措施，能够保证在演练期间出现问题后迅速把影响降到最低，这也是另外一种变相的爆炸半径控制。本质上来说，爆炸半径和通过演练发现的问题数量会有一个正比的关系，即爆炸半径控制得越小，整个故障演练对业务的影响也就越小。但与此同时，整个故障演练暴露出来的问题，就会比较有限，所以在建设过程中要做一个取舍。如果想发现更多的问题，那爆炸半径就要设计的大一些。当前，综合考量多种因素，我们还是把爆炸半径控制在一个相对合理范围去进行故障演练，让业务以一个更低的成本进行故障演练。

2.6 演练复盘

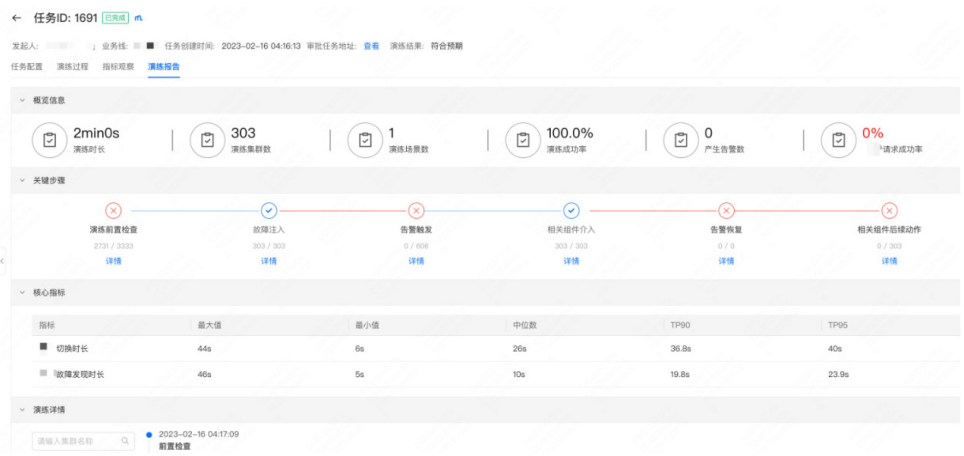
当演练结束后，我们需要对整体的演练结果做客观的评价分析，进而判断这次演练整体是否达到预期。演练复盘内容包括演练整体流程、存在的问题以及针对问题形成的相关改进点，它可以用来指导后续的演练。如果本次演练不符合预期，我们会通过演练复盘功能打通故障演练平台和故障复盘两大产品，自动生成演练相关的时间线，从而更好地进行人工复盘。

• 提供故障演练的复盘报告，为 RD/DBA 人工判断演练是否符合预期提供数据支撑
• 演练不符合预期的场景下（包括 RD/DBA 人工反馈和系统自动依赖稳态判定不符合 SLA ），打通 故障演练平台 和 故障复盘两大产品，自动生成时间线用于人工复盘演练的结果



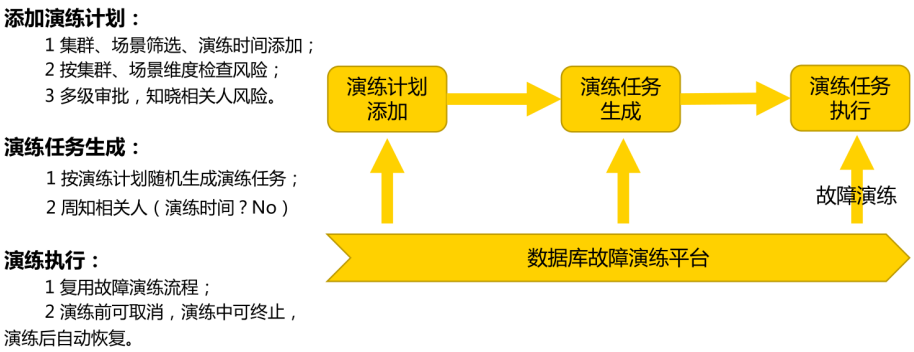
演练复盘主要是分为四部分：

- 第一部分是演练复盘的概览信息，它主要是包含六个方面：① 演练时长；② 本次演练涉及的集群数；③ 演练场景数；④ 演练成功率，即故障注入的成功率；⑤ 演练过程中产生的告警数；⑥ 演练期间业务请求的成功率。
- 第二部分是关键步骤，这里收集了在整个演练过程中非常关键、非常核心的步骤，以及它整体的指标数据，这些可以帮助我们分析平台本身亦或是防守组件的相关表现是否符合预期。
- 第三部分是在演练过程中涉及的核心防守组件的指标数据，比如针对主库宕机，我们主要关心两方面数据：一是故障注入后，高可用组件用了多长时间才发现这个故障，二是在发现了这个故障后切换花了多长时间。通过这些数据，就可以知道在本次演练过程中防守组件的表现是否符合预期。
- 第四部分是演练详情，它是针对单个集群的演练过程绘制的完整的时间线信息，可以通过这些信息看到一个演练集群从前置检查到故障注入，到告警产生，到防守组件的介入，再到故障清理呈现出的一个非常详细的时间线数据，如下图所示：



2.7 随机、无通知演练

从 2022 年开始，我们开始建设随机、无通知演练能力。之前故障演练平台主要是在特定时间、特定场景下的演练，并且在整个演练过程会组织相关的人员协同跟进整个演练过程。但是，故障的发生是一个非常随机的行为，它不会给人足够的时间去准备、应对。而随机无通知演练功能，就是希望建设这样的能力，可以在非特定时间、非特定集群、非特定场景进行故障演练，在一定程度上填补了故障演练的空白，也是故障演练平台向混沌工程演进的一个里程碑。整个演练过程主要分成三个部分：



第一部分是需要添加演练计划，其主要内容包括三个方面：

- 可以通过集群、场景、演练时间这 3 个维度，去控制哪些集群可以演练哪些场景，在什么时间可以进行随机无通知演练；
- 按集群、场景维度组合去做相关风险检查，以此来保证添加的这些集群可以进行

相关场景的演练；

- 和常规故障演练相似的是，它也提供了多维审批，可以让相关业务人员或者 DBA 知晓个整个演练的风险。

第二部分是演练任务生成，有了演练计划后，我们按演练计划随机搭配生成演练任务并进行周知，和常规演练最大不同是我们不会周知演练的具体时间（会周知演练集群、场景），主要是为了模拟故障发生的随机性。随着后面整体能力的升级迭代，我们会逐步去掉在集群、场景维度的周知。

第三部分是演练任务执行，它完全复用我们常规故障演练的功能，也可以在演练前取消、演练中终止，演练后可以自动恢复整个集群的拓扑。

2.8 演练模式对比

下面我们将常规故障演练、随机无通知演练以及混沌工程作一下对比，如下表所示：

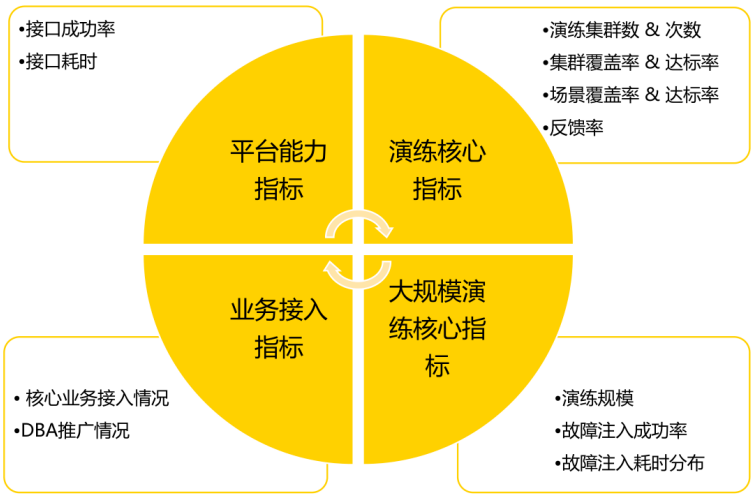
演练类型	演练目的性	故障场景选择	有人值守	事先周知
常规故障演练	强，验证已知的问题场景是否会再次造成事故	指定集群的指定场景	是	是，明确知道演练时间及相关风险
随机、无通知演练	中，已知故障场景在有限集群范围内，是否会产生问题	一定集群范围内的已经成熟的场景	否	中，有限集群、场景范围内，不确定具体演练时间及相关风险
混沌工程	弱，检验未知的故障场景是否会造成事故	所有集群的所有场景的完全随机的组合	否	否，完全无周知

- 首先从演练目的维度来看：常规故障演练目的性非常强，它主要是验证已知的问题场景是否会再次造成生产上的事故；随机演练主要是在已知的故障场景、在有限的集群范围内，是否会产生问题；混沌工程的演练目的性最弱，它主要是检验未知的故障场景是否会造成问题。
- 从是否有人值守的维度来看：常规故障演练需要有人值守；随机无通知演练是朝着无人值守方向发展的，但当前鉴于我们的能力还在建设初期，所以也会增加一些相关的值守机制来控制演练风险；混沌工程是完全无人值守的，因为也无法去值守一个混沌工程的“混沌”实验。
- 从事先周知的维度来看：常规故障演练需要明确进行周知；随机无通知演练当前

在有限集群和有限场景范围内进行演练（会周知演练集群、演练场景），不会周知具体的演练时间；混沌工程是完全无周知的演练形式。

2.9 演练运营体系

下面是数据库故障演练平台的运营体系的建设，主要关注四个部分：



首先是演练核心指标：

- 一是在一段时间内，共进行了多少次演练，涉及演练集群数有多少个；
- 二是在一段时间内演练集群和演练场景的覆盖率以及达标率；
- 三是故障演练结束后的反馈，即演练是否符合预期。

第二部分是大规模演练的核心指标，我们主要关注三个方面：

- 第一方面是每次大规模演练的演练规模；
- 第二方面是故障注入的成功率；
- 第三方面是故障注入的耗时分布，主要统计在某一个时间点，完成了多少实例的故障注入。

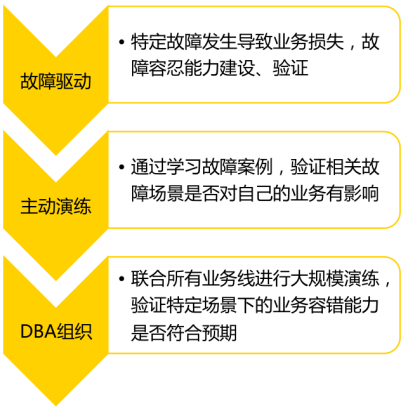
第三部分是业务接入指标，一方面统计核心业务的接入情况，同时也会反映一下 DBA 在不同业务线的推广情况。

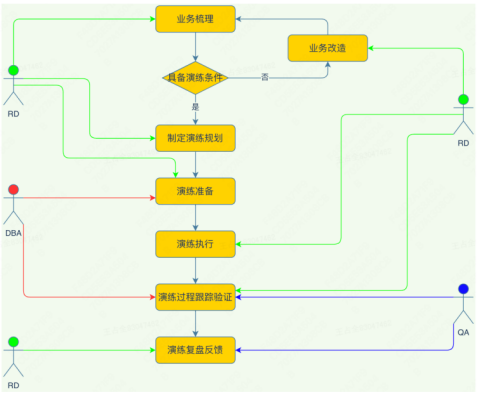
第四部分是平台能力指标，一方面是平台接口成功率，另一方面是平台接口的相关耗时。

3. 落地情况

3.1 演练推广

本部分会介绍故障演练平台在美团内部的落地情况。首先在推广层面，我们主要采用三种形式：

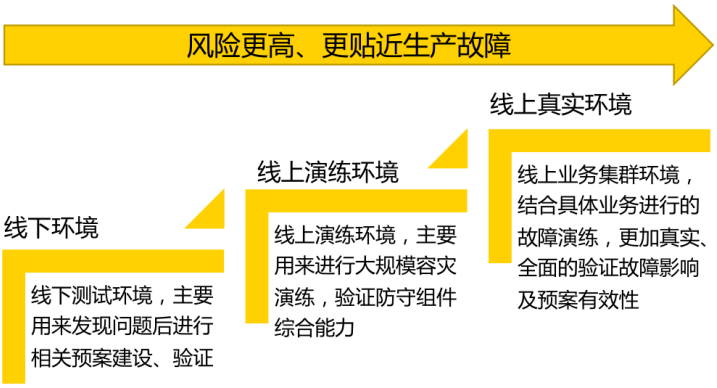




- 第一种形式是故障驱动方式，如果线上发生了特定的故障场景，导致业务损失，我们就会进行相关故障场景的容忍能力建设以及验证。
- 第二种形式是主动演练，通过定期学习一些故障案例，判断相关的故障案例中的相关故障场景是否会对目前负责的业务产生影响，也可以通过故障演练平台做简单的摸底。
- 第三种形式是 DBA 主动策划组织，比如 2022 年，我们进行了三次规模比较大的演练，主要验证在特定故障场景下，业务的容错能力、高可用的防守能力是否符合预期。

3.2 实验环境

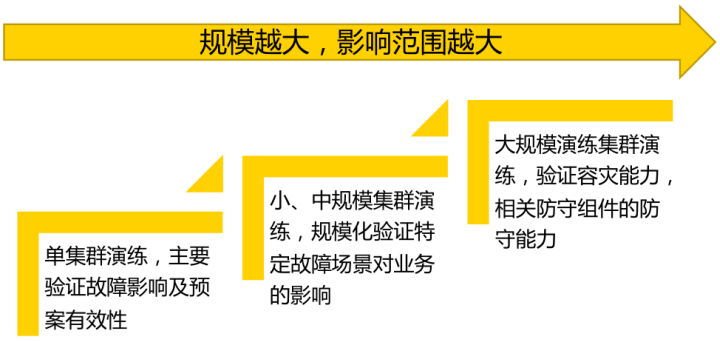
在混沌实验的环境选择上，我们目前提供了三种不同的环境：



- 第一种是线下环境，主要是用来发现问题后进行相关预案的建设和验证；
- 第二种是线上演练环境，主要是通过大规模的容灾演练，去验证防守组件的防守能力是否符合预期；
- 第三种是线上真实环境，就是我们真正的线上业务集群，主要通过具体业务进行相关故障场景的演练，能够更加真实、全面地验证故障的影响，以及相关预案的有效性。

3.3 演练规模

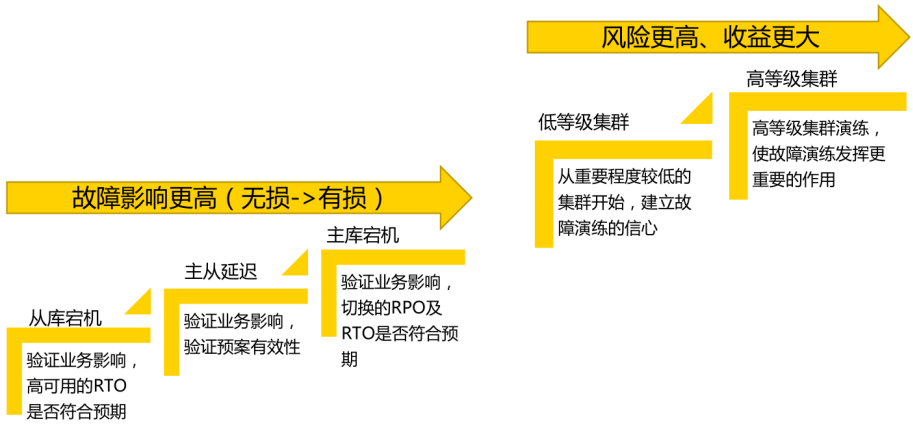
接下来是演练规模，演练的规模越大，影响的范围越大。我们的演练规模可以划分成三类：



- 第一类是单集群演练，主要验证故障影响及预案有效性；
- 第二类是中小规模演练，主要通过规模化的演练，去验证特定故障场景对业务的影响；
- 第三类是大规模的集群演练，主要验证容灾能力，以及相关防守组件的防守能力是否符合预期。

3.4 场景 & 等级

在整体推进方面，我们目前会从两个维度进行演练：



- 第一个维度，我们会根据故障影响的严重程度，做无损到有损场景的演练；
- 第二个维度，演练会从低等级集群开始，最终过渡到高等级集群，这样做的目的是先给业务建立一个对于故障演练的信心，之后逐步在高等级集群中发挥故障演练的重要作用。

3.5 演练运营数据

下面是常规故障演练、随机无通知演练以及大规模故障演练的一些相关数据：

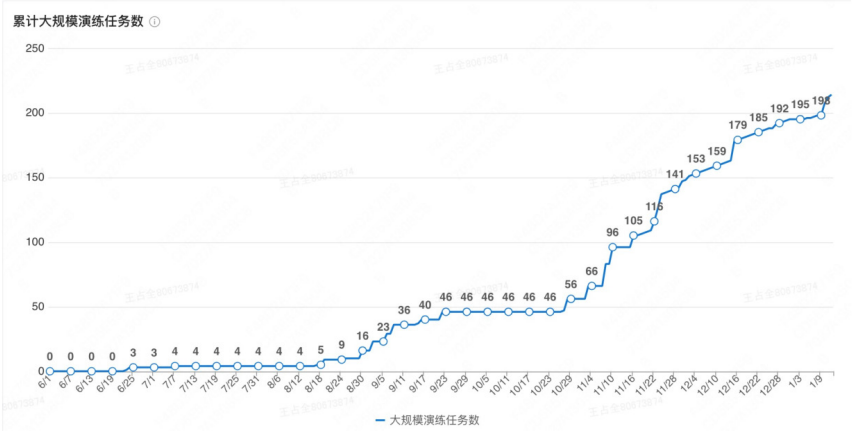
常规故障演练

支持的故障场景数	支持的故障注入方法数	演练次数 （集群*该集群参与演练次数的累加）	单集群演练次数
5	9	877	360
MySQL集群总数	参演集群数	集群演练覆盖率	参演业务数
■ ■	■	6.01%	53

随机无通知演练

随机演练故障场景数	随机演练故障注入方法数	随机演练次数	随机演练参演集群数
4	5	25	8

大规模故障演练

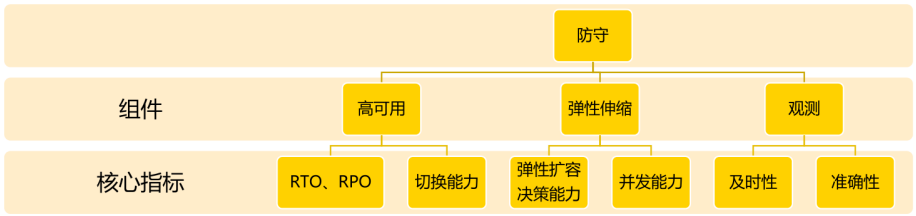


- 从常规故障演练的数据可以看出，当前我们整体上的常规故障演练覆盖率是偏低的。
- 随机功能演练是 2022 年第四季度才建设完成的能力，到目前还是初级推广阶段，所以整体上演练次数比较少，2023 年的主要目标是通过在核心业务线的推广，不断提高演练集群的覆盖率，以期发现更多的问题。
- 大规模演练从 2022 年下半年开始，在演练频次有非常大的提高，这主要是为了更好地验证数据库底层依赖的防守能力在面临大规模故障时的表现。如果发现相关组件存在问题，我们也可以更好地辅助这些组件去做迭代，辅助建设相关的能力。上图统计的是单任务演练规模超过 100 个集群的数据，但如果按照 20 或 50 个集群进行统计汇总，那么大规模演练的任务次数还会更多。

3.6 防守体系能力验证

在防守体系能力建设层面，我们主要验证防守组件的三个核心能力：

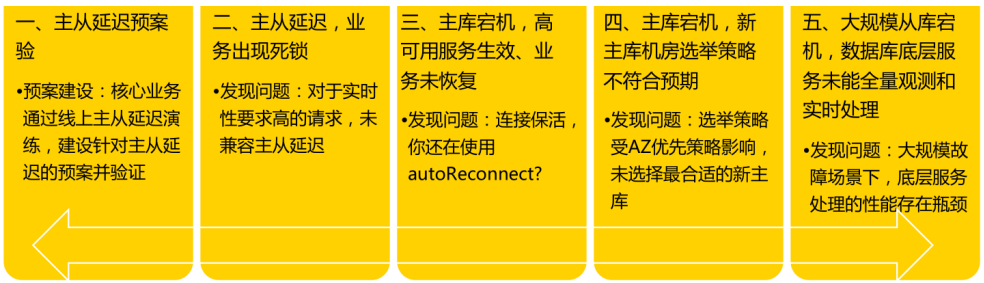
有助于发现防守组件问题，辅助修复验证
周期性、大规模的故障演练，整体提升防守能力



- 第一个从高用来看，主要验证 RTO 和 RPO 是否符合预期，以及在大规模故障场景下，其切换能力是否达标。
- 第二个从弹性伸缩来看，主要验证在单集群或规模化演练前提下，整体弹性扩容决策能力是否达到期望，以及在小规模场景下实地验证弹性扩容的并发能力。
- 第三个从观测来看，主要验证当大规模演练发生时，观测工具的及时性以及准确性。

3.7 业务演练收益

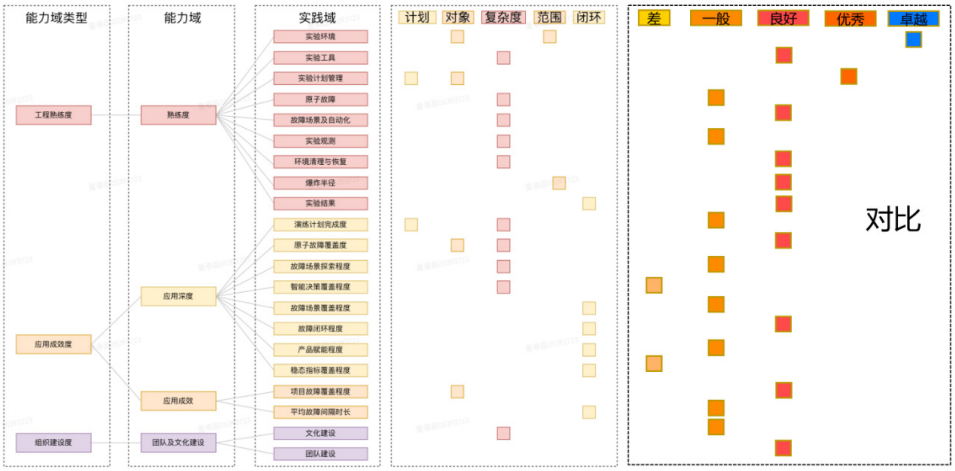
诚然，演练会对业务方造成一些“打扰”，但是业务方也会积极配合我们做这项工作，因为故障演练对业务方也有一些收益：一方面，业务方可以通过演练发现一些隐藏的问题，从而规避更大的风险；另一方面，我们在发现业务问题的同时，也能验证相关防守组件的表现是否符合预期。更多的细节，大家可以参考下图：



4. 未来展望

4.1 混沌工程成熟度模型

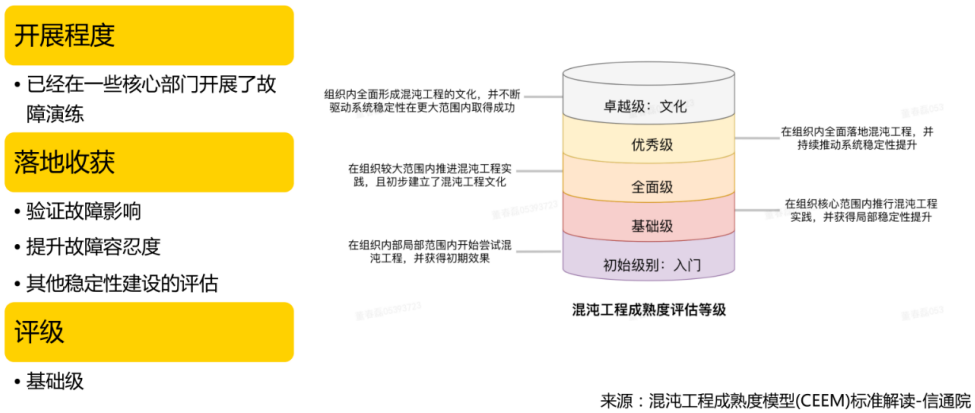
下图是混沌工程成熟度模型（CEEM），右边是我们当前的数据库故障演练平台和标准的对比。可以明显地看到，当前我们在一些维度上做的还不够，比如当前故障场景的覆盖程度、稳态指标的覆盖程度、实验观测、文化建设等方面都存在一定的不足，而这些不足其实也是我们后续着重要改进的点。



来源：混沌工程成熟度模型(CEEM)标准解读-信通院

4.2 混沌工程成熟度等级

关于混沌工程成熟等级，我们也做了一些总结：

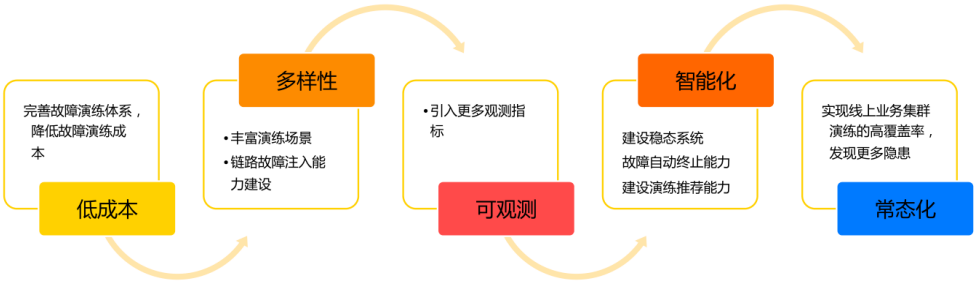


来源：混沌工程成熟度模型(CEEM)标准解读-信通院

- 从开展程度来看，当前我们只是在一些核心部门进行了故障演练，在稳定提升方面会有一些相关建设的成果；
- 从落地收获来看，一是可以验证故障的影响，二是可以提高业务系统对于故障的容忍度，三是可以通过数据库故障演练平台，评估我们在稳定性建设方面的一些成果；
- 从整体评级来看，我们当前还处于基础阶段，后面也会继续在公司更多的核心部门推广数据库故障演练平台，持续为业务系统稳定性提升做出更大的贡献。

4.3 具体规划

在最后部分，我们分享一下关于未来的规划。在具体路径层面，我们会从五个方面进行开展工作：



- 第一，要降低单次演练成本，比如降低单次演练的介入率，无论是业务还是DBA。
- 第二，丰富故障演练场景，建设链路故障注入能力，从服务端故障向端到端故障做一个扩展。
- 第三，增强可观测性，当前整个故障演练平台在观测指标方面相对比较缺乏，后续我们希望通过引入更多的观测指标，从而更好地观测整个混沌工程的实验过程。
- 第四，智能化，主要聚焦三方面：① 建设整个稳态系统，因为当前我们整个故障演练是否符合预期，主要还是通过人工或者 DBA 来进行判断，有了稳态系统后，可以通过稳态系统去判断；② 根据稳态系统相关指标建设故障自动终止能力；③ 建设相关的演练推荐能力，因为现在业务线推广演练时，可能还不清楚我们该在哪些集群演练哪些场景，当有了这个能力后，业务在演练时会有更强的针对性。
- 第五，希望能够实现演练常态化，即通过常态化演练去发现业务端更多的隐患。

超大规模数据库集群保稳系列之三：美团数据库容灾体系建设实践

作者：瑞超

1. 容灾介绍

我们通常会把故障分为三大类，一是主机故障，二是机房故障，三是地域故障。每类故障都有各自的诱发因素，而从主机到机房再到地域，故障发生概率依次越来越小，而故障的影响却越来越大。

容灾能力的建设目标是非常明确的，就是要能够应对和处理这种机房级和地域级的大规模故障，从而来保障业务的连续性。近几年，业界也发生了多次数据中心级别的故障，对相关公司的业务和品牌产生了非常大的负面影响。当前容灾能力已经成为众多 IT 企业建设信息化系统的必选项。



2. 业务容灾架构

2.1 容灾架构演进

容灾架构从最早期的单活形态（同城主备）到同城多活形态，再演化到异地多活，根据这个路径可以将容灾分为容灾 1.0、容灾 2.0、容灾 3.0 三个阶段。

- **容灾 1.0**：容灾体系围绕数据建设，多以主 – 备的方式部署，但备用机房不承担流量，基本上都是单活结构。

- **容灾 2.0**：容灾视角从数据转换为应用系统，业务具有同城双活或同城多活能力，采用同城双活或同城双活加异地冷备（两地三中心）的部署架构，除冷备以外的每个机房都有流量处理能力。
- **容灾 3.0**：以业务为中心，多采用单元化架构，容灾基于单元间的两两互备实现，根据单元的部署位置可以实现同城多活和异地多活。采用单元化架构的应用本身具有很好的容灾能力和扩展能力。

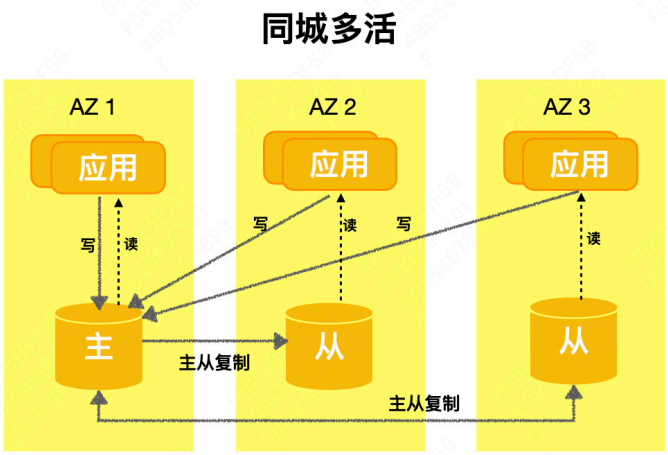


由于各公司所处发展阶段不同，采用的方案也会有所区别，美团大部分业务处于 2.0 阶段（即同城双活或多活架构），但对于大体量、有地域容灾及有地域扩展性要求的业务则处在容灾 3.0 阶段。下面会介绍一下美团的容灾架构。

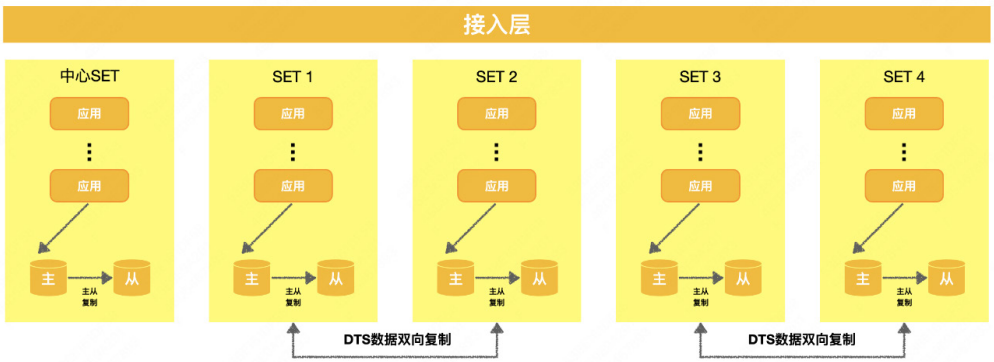
2.2 美团容灾架构

美团的容灾架构主要包括两种，一种是 N+1 容灾架构，一种是 SET 化架构。

N+1 架构：在业界也称散部或者多 AZ 部署　案，将容量为 C 的系统部署在 N+1 个机房，每个机房能提供至少 C/N 的容量，挂掉任何一个机房时，剩余系统仍能支撑 C 的容量。该方案的核心是把容灾能力下沉到 PaaS 组件来完成，在出现机房级或者地域级故障的时候，由各个 PaaS 组件独立完成容灾切换，实现业务恢复。整体架构如下图所示，业务上表现是多机房、多活形态，数据库采用这种主从架构，单机房处理写流量、多机房的负载均摊读流量。下面要讲“数据库容灾体系建设实践”就是面向 N+1 架构的。



单元化架构：也叫 SET 化架构，这是一种偏应用层的容灾架构，它将应用，数据，基础组件按照统一的维度切分成多个单元，每个单元处理一部分闭环流量。业务以单元作为部署单位，通过单元互备方式实现同城容灾或者异地容灾。一般金融业务或者超大规模的业务会选择此类架构，它的好处就是流量可以闭环且资源隔离，具有很强的容灾能力和跨域扩展能力，不过 SET 化架构的落地需要业务系统做大量的改造，运维管理也较为复杂。简化示意图如下：



美团内部的大部分业务都是 N+1 架构，外卖和金融等业务采用了单元化架构。总体上美团内部既有同城多活，也有异地多活，两种容灾方案并存。

3. 数据库容灾建设

3.1 面临的挑战

超大规模的集群带来的挑战：公司业务高速发展，服务器规模指数级增 ，数据中心规模越来越大，大机房已有大几千数据库集群，上万个实例。

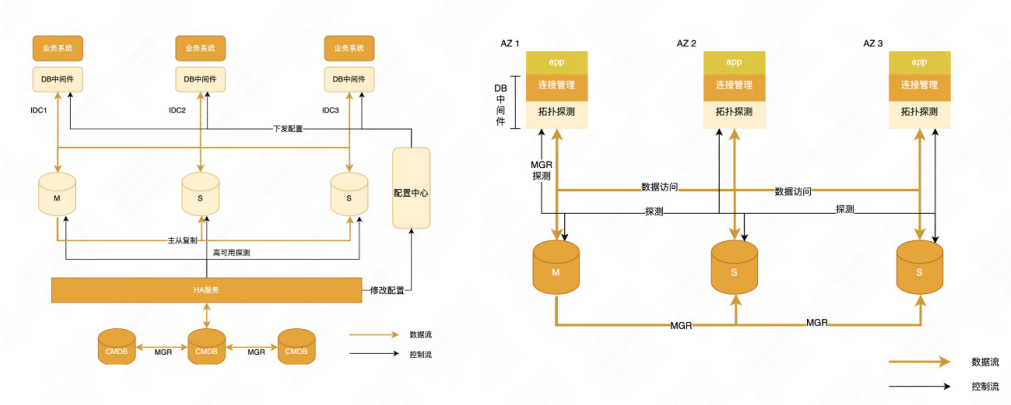
- **性能问题**：高可用系统的故障并发处理能力出现明显瓶颈。
- **容灾失效风险**：管控链路随集群数量的增加变的越来越复杂，一个环节出问题就会导致整体容灾能力失效。
- **故障频发**：集群数量和规模变大，使原来概率很低的大规模故障变成了稀松平常的故障，其发生的频次和概率越来越高。

演练成本高、频次低：核心能力验证不充分，大规模故障的应对能力处于不可知状态，已知容灾能力“保鲜”困难。拿应对机房级大规模故障的相关能力来讲，很大一部分是处于不可知状态或者仅存在于“纸面”分析中，而对于已验证过的能力随着架构演进迭代，“保鲜”也很困难。

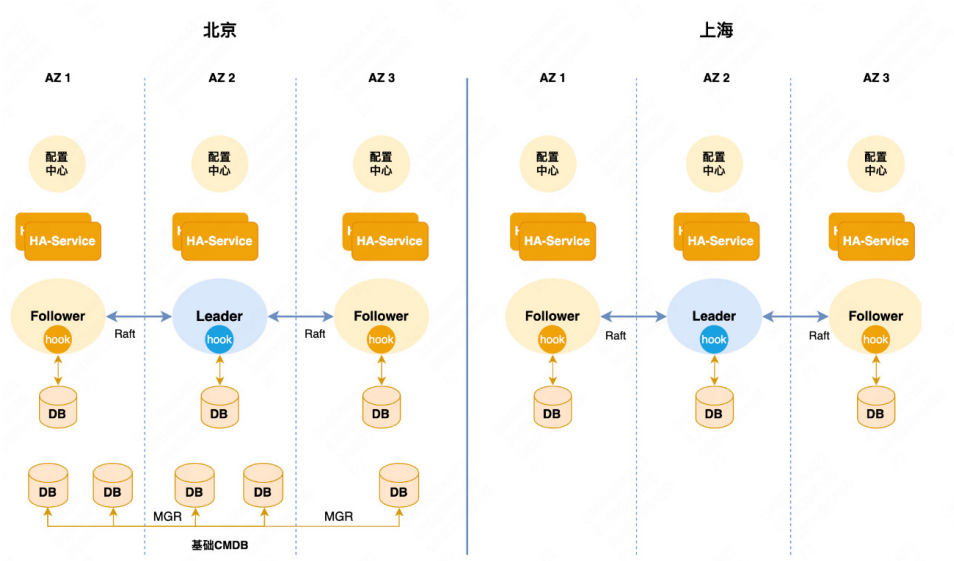
数据库作为有状态的服务之一，本身建设应对大规模故障能力的难度和挑战都相对更大。

3.2 基础高可用

数据库架构 在美团主要有两种一种是主从架构，一种是 MGR 架构。

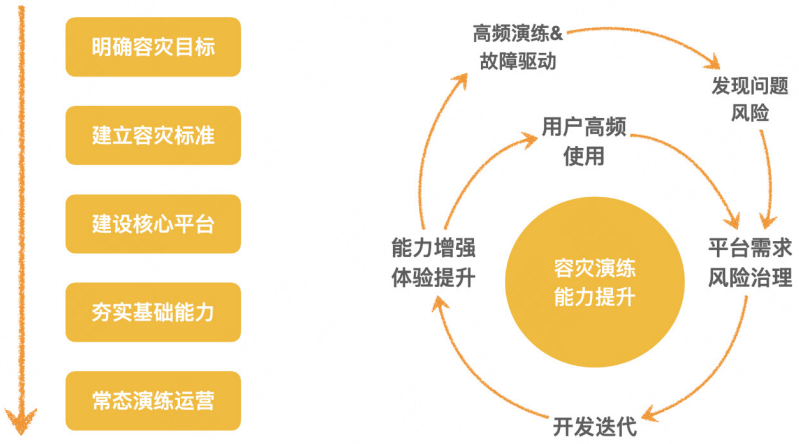


- **主从架构**：应用通过数据库中间件访问数据库，在故障发生时，高可用做故障探测、拓扑调整、配置下发，进而应用恢复。
- **MGR 架构**：应用也是通过中间件访问数据库，不过中间件对 MGR 做了适配，内部叫 Zebra for MGR，中间件自动进行拓扑探测感知，一旦 MGR 发生了切换，新拓扑会被探测到，数据源会进行调整，进而业务恢复。
- **美团的高可用架构**：美团主从集群的高可用是基于 Orchestrator 二次开发的，本质上是一个中心化的管控架构，如下图所示，有多个高可用分组，每个分组托管一部分数据库集群，分组在北京和上海实现两 Region 部署，底层核心组件只在北京部署，比如我们的核心 CMBD、WorkflowDB 等，一旦北上专线出现问题，上海侧的高可用会失效不可用。



3.3 容灾建设路径

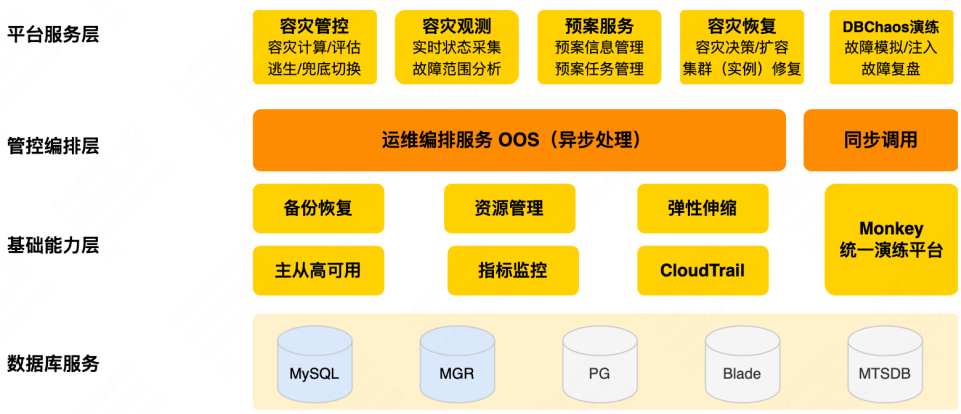
- **容灾建设路径**：确定容灾目标、制定容灾标准、建设容灾平台、夯实基础能力、演练验证和风险运营。
- **容灾建设飞轮**：内环是平台能力建设，从容灾需求的提出到研发上线，体验提升，用户使用，发现问题提出新需求，不断的迭代提升。另一个方面就是完善演练平台建设，开展高频演练（或者真实故障驱动），发现问题、提出改进，促近平台能力持续迭代提升。



3.4 平台能力建设

为了建设提升数据库服务的容灾能力，内部成立了容灾管控项目 DDTP（Database Disaster Tolerance Platform），专注提升数据库应对大规模故障的能力，核心包括基础容灾管控和故障演练两大能力，分别对应两个平台产品：一是容灾管控平台，一个是数据库演练平台。

容灾管控平台主要专注于防守，它的核心功能主要包括事前逃生、事中观测以及止损、事后恢复等，数据库演练平台则专注于进攻，支持多种故障类型和多种故障注入方式，具备故障编排，故障复盘等核心能力。这个系列的第二篇《数据库攻防演练建设实践》就是对演练平台的详细介绍。接下来，我们将重点介绍一下容灾管控平台的主要内容，首先看一下全景图：



- **数据库服务**：包括 MySQL、Blade、MGR 等基础数据库服务。
- **基础能力层**：主要是备份恢复、资源管理、弹性伸缩、主从高可用以及指标监控能力，这些能力是稳定性保障的基本部分，但在容灾场景下需要进一步加强，以处理大规模故障场景。
- **管控编排层**：核心是运维编排服务 OOS（Operation Orchestration Service），会把基础能力按需编排生成对应的处理流程也叫服务化预案，每个预案对应一个或者多个具体的运维场景。容灾预案也在这个范畴。
- **平台服务层**：是容灾管控平台的能力层，包括：1）**容灾管控**，容灾计算评估和隐患治理，还有故障前容灾逃生、故障中的兜底切换，故障摘流等。2）**容灾观测**，明确故障范围，支持故障中的容灾决策。3）**容灾恢复**，故障后通过实例修复、集群扩容等功能快速恢复集群的容灾能力。4）**预案服务**，包含了常见故障应急预案的管理和执行等等。

3.4.1 容量达标

数据库建立了一套 N+1 容灾计算标准，分为 6 个等级，如果集群容灾等级≥ 4 级则容灾达标，否则容灾不达标。

0	1	2	3	4	5
单机（共宿主机）	多机部署	多机部署	多个机房	多个机房	多个机房
		主从HA	主从HA	主从HA	主从HA
				各区域满足 N+1	各区域满足 N+1
			SET化评估		各地域容量对等
				SET化评估	
					SET化评估

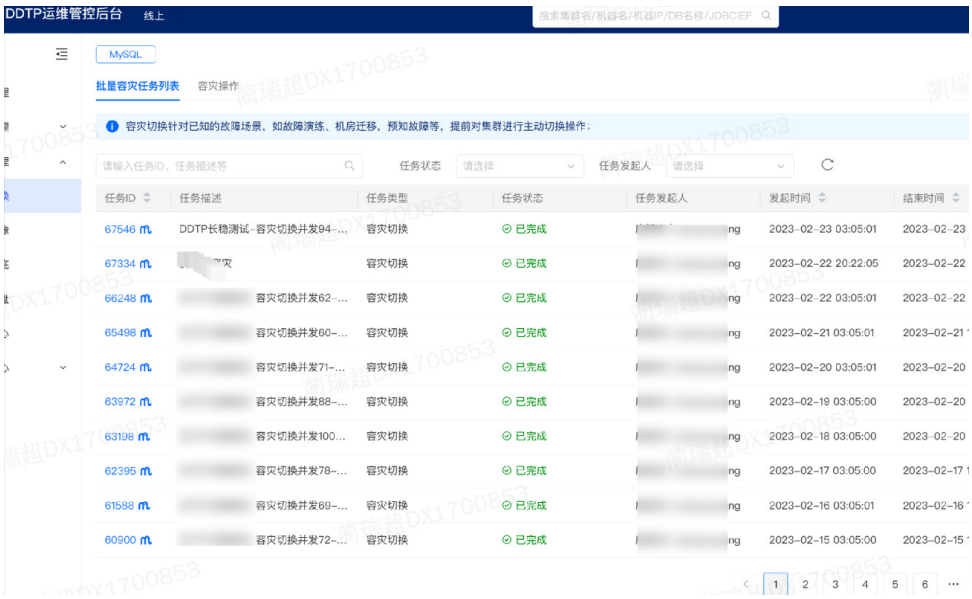
从标准可以看出，从等级 3 开始就是多机房部署了。3 级和 4、5 级的区别是，3 级不满足 N+1 要求，即如果一个机房的节点都出问题，剩余节点无法承担峰值流量。等级 4、5 都是具备 N+1 要求的，等级 5 会满足 region 间容量对等。除基础标准以外，SET 化集群有特殊规则，比如路由策略要闭环、SET 集群的绑定机房要统一、互备 SET 容量要对等、集群内机型要统一等。这些规则都会纳入容灾计算来确定集群的最终容灾等级。

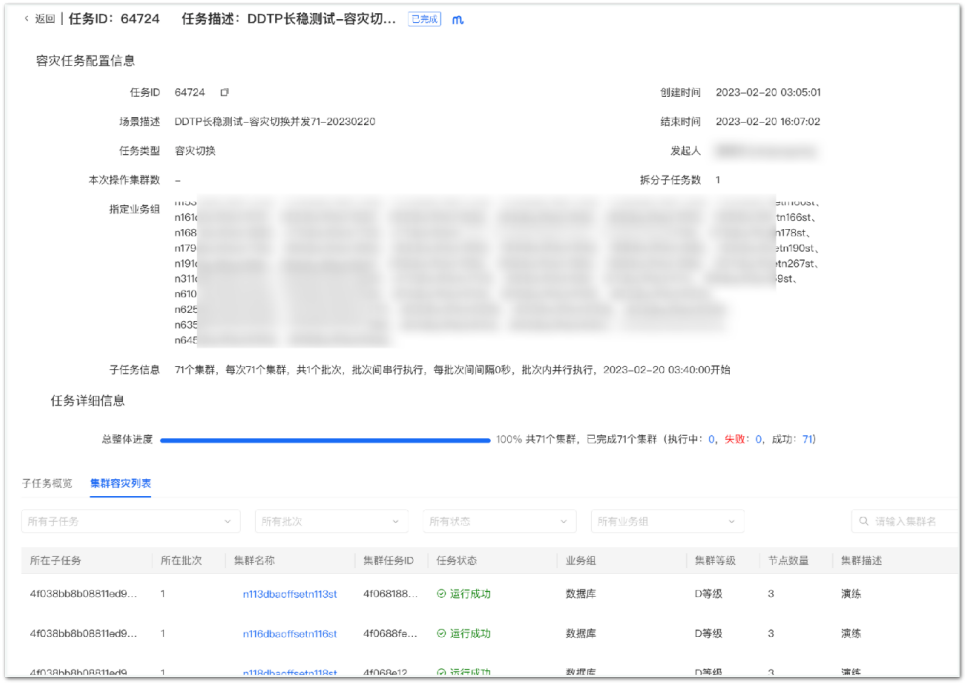
在基础容灾数据建设中，会把上述规则代码化、计算流程化，通过近实时的方式做基础数据“保鲜”。容灾数据是容灾管控平台上用于逃生切换和事中止损的基础数据，同时还会基于容灾数据建设风险隐患（即容灾不达标隐患），并通过一定的运营治理来消除这种隐患。

3.4.2 故障前逃生

故障前逃逸能力就是批量主库切换和从库摘流，主要用于在故障前收到预警，提前感知灾难来临，快速将一个机房的所有数据库服务切走或者下线从库流量，以降低真实故障带来的影响。

我们知道对于主从架构的集群，如果因为断电或者断网发生故障切换，很可能会发生数据丢失。数据一旦丢失，业务需要进行确认并做善后工作，有时候会非常繁琐。如果能够在事前逃走就会把这些风险都规避掉。同时除了主库逃走以外，从库也可以提前把流量“摘掉”，从而做到故障对业务方“无感”。





3.4.3 故障中观测

在大规模故障发生的时候，一般会出现告警轰炸，电话咨询轰炸等情况，如果没有全局的故障感知能力，就会使故障处理比较混乱，处理时间比较长，让故障影响放大，所以我们建设了容灾观测大盘，它能够实时、准确、可靠地对故障进行观测，以确保值班同学能够掌握故障的实时情况。

如下图所示，如果发生了故障，可以快速拿到故障集群或者实例列表，并在对应的页面上发起兜底切换动作，进而实现快速止损。对观测大盘的核心诉求就是要实时、准确、可靠。可以通过减少服务依赖来提升自身的可用性。



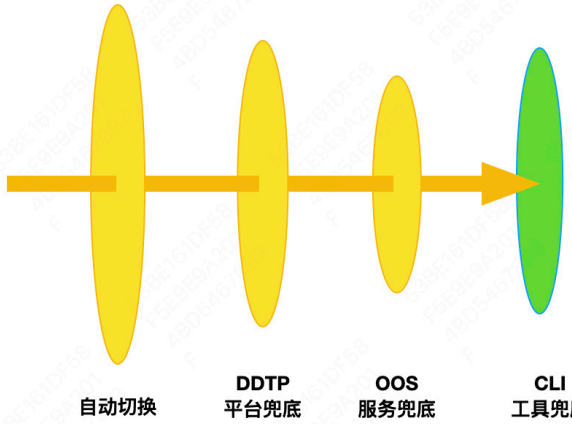
3.4.4 故障中止损

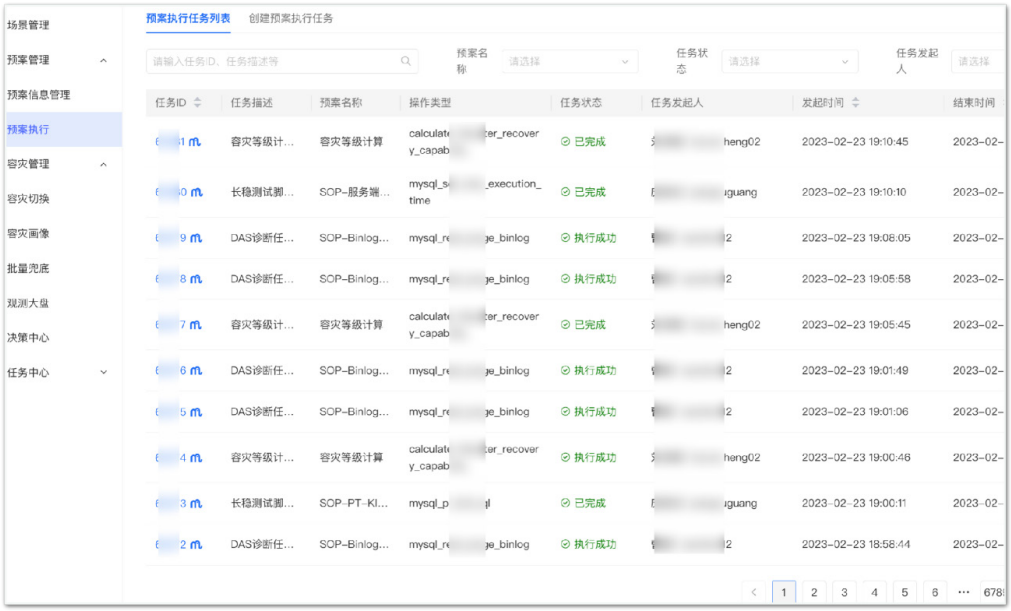
在介绍故障中的止损之前，先了解一下预案服务。预案服务的核心功能就是管理常见故障以及对应的各种处理预案，并提供执行控制能力，让预案可以方便、可控地运行。



故障止损：在有了预案以后，我们就可以进行兜底止损。如下图所示，当大规模故障发生的时候，HA 会自动进行故障处理。如果集群切换失败或者漏切，那么它就会进入兜底阶段。首先从 DDTP 平台化兜底，如果平台受故障影响不可用，可以在运维编排层进行兜底。如果运维编排服务也失效，则需要人工通过 CLI 工具进行兜底。CLI 是

DBA 最底层的工具，它和高可用是两个独立的链路。CLI 会进行集群拓扑探测、选主选举、拓扑调整、配置修改、配置下发等逻辑，等同于手工集群切换步骤。





总体原则优先提升高可用自动切换的成功率，减少透传到兜底阶段的集群数量。其次提升预案可靠性，优先选择白屏，逐级下沉，易用性下降，可靠性提升。

3.4.5 故障后恢复

虽然集群具备 N+1 能力，一个机房故障的时候，集群剩余节点是能够支撑峰值流量，

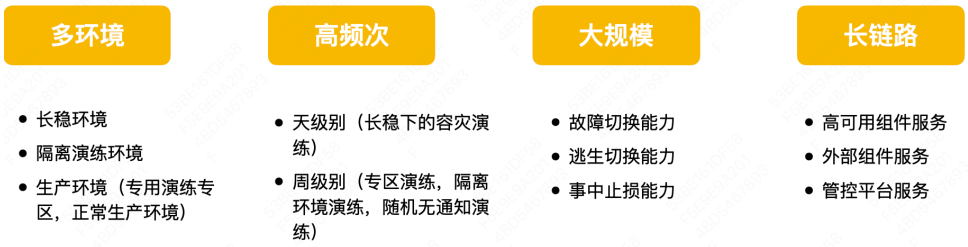
但它不具备再一次 AZ 故障的容灾能力，所以在故障后会根据各机房的资源情况，通过容灾决策中心快速进行集群扩容来补齐核心集群的容灾容量，使其重新具备 AZ 容灾能力。

上述方案有一个比较大的弊端就是需要有足够的资源来进行扩容，这是非常困难的，目前我们在建设更快速的恢复能力，如实例原地修复，集群原地扩容等，在 AZ 恢复之后，可以快速复用发生故障的机房内的机器资源，实现容灾快速恢复。



3.5 演练体系建设

各项基础容灾能力不能只存在于架构设计、理论评估层面，必须实打实的可用，这就要需要通过演练进行验证。容灾管控项目之初，就制定了以演练为抓手的策略，验证并驱动各项基础能力的提升。 截止目前，已经初步建成了多环境、高频次、大规模、长链路的演练体系。



- **多环境**：我们建设了多种演练环境，满足各个 PaaS 组件的各类容灾演练需求。一是容灾管控平台的 稳环境，二是线下专用于演练的隔离环境，三是生产环境，有演练专区以及正常生产环境。
- **高频次**：目前能做到天、周级别。天级别属于常态化的演练，主要是在长稳环境下自动发起，几百个集群的演练规模；周级别主要是在隔离环境、演练专区定期组织的断网、断电真实演练等。
- **大规模**：是在生产环境开展的演练，用于验证基础高可用、兜底预案、逃生预案、容灾恢复等功能的大规模、高并发处理能力，确定管控系统的服务容量。
- **长链路**：整个容灾链路涉及到很多组件，包括 CMDB 数据库、流程数据库，高可用组件，配置中心、预案服务等，我们会逐步把这些组件都纳入演练，可以让一个或者多个组件服务同时故障，发现潜在问题，验证多服务的多节点同时故障对于整个故障处理能力的影响。

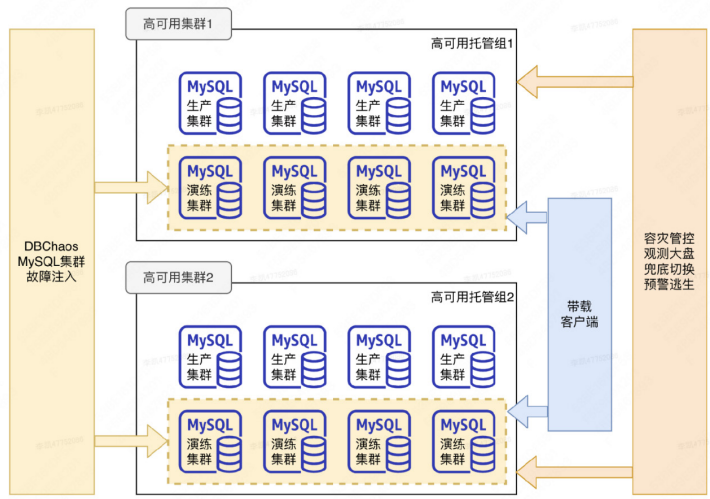
3.5.1 隔离环境演练

隔离环境演练顾名思义，它是一套和生产机房完全隔离的演练环境，有自己独立的 TOR、机柜，风险能做到完全隔离，可以开展独立断网或断电操作。参与演练的 PaaS 组件和业务服务要在该环境独立部署。在隔离环境除了会定期开展各种容灾演练发现容灾问题外，还可以验证各 PaaS 的独立部署能力，为国际化业务支撑提供基础。

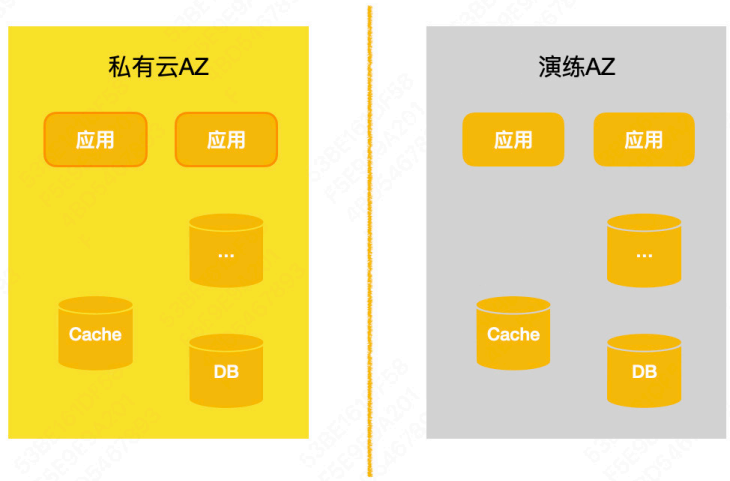
3.5.2 生产环境演练

- **常态化、大规模故障演练**：此类演练是日常持续开展的，通过演练平台对数据库集群注入故障，高可用进行故障切换。通过不同的演练规模来验证高可用的并发切换能力。此外，在容灾管控平台上，可以验证逃生能力、止损预案、及大规模故障的观测等。总而言之，它是利用“攻”和“防”相结合的形式，实现能力的验证验收和优化提升。

这类演练主要特点：一是参演集群都是由生产环境的高可用分组进行托管，就是说演练验证的都是生产环境的高可用的能力；二是参演的大规模集群是非业务集群，是每次演练前新创建的专门用于演练的集群，规模可以做到很大，目前可以常态化的进行 1500+ 集群同时进行演练；三是有一定的仿真效果，为使演练更为真实并对 RTO 做精准评估，对演练集群增加了带载能力。



- **真实专区演练**：上文介绍的隔离环境演练、大规模演练都是偏模拟性质的，和真实的故障场景有比较大的区别。为了弥补和真实故障主键的 GAP，我们基于公有云构建了一个专用演练 AZ，可以理解为就是一个独立的机房。参演业务和组 PaaS 件将部分承载业务流量的服务节点部署到演练 AZ 中，实际演练的时候会进行真实的断网，业务和组件可以在断网的时候观测和评估自己的容灾情况。这种通过真实机房、真实组件集群、真实的业务流量来验证组件和业务的实际容灾情况，会更加真实。



- **Game Day**：此外我们还在评估论证在真实机房开展演练的可行性，随着隔离环境演练、专区演练的常态化开展，各个组件的基础容灾能力会越来越强，在真

实机房进行常态化机房演练的终极目标也会随之达成。

4. 未来思考

经过两年多的建设，虽然在高可用自动切换、容灾能力运营治理、大规模故障观测、故障止损预案、容灾恢复等方面取得了一定的成果。但是还有很多能力短板需要建设补齐，业务新的发展也带来了新的需求和挑战。未来我们会补齐短板、迭代技术架构两个方向上进行持续的提升。

4.1 补齐短板

- **超大规模逃生能力、止损能力不足**：随着我们自建数据中心的落地，我们自建的AZ 规模会更大，这对能力的要求会更高，我们主要通过平台迭代和演练验证逐步提升能力。
- **跨域专线故障导致 Region 级高可用失效**：接下来我们会探索单元化方案或者独立部署方案，实现 Region 级或者更细粒度的闭环管理。
- **业务出海新挑战**：出海会给容灾架构带来一些新需求和挑战，是采用“长臂管辖”还是独立部署，是复用现有技术体系还是打造一套全新架构，这些问题都还需要进一步的探索和论证。
- **容灾效率问题**：平台基础功能已经相对完善，不过容灾决策以及处理协同等还需要人工进行，效率相对较低，未来会把容灾管控、应急止损等能力逐步向自动化演进；多环境演练成本比较高，也要逐步做自动化演练，把核心的演练场景逐步纳到长稳环境，通过定时或一定的策略让它自动去跑故障场景，我们只需要关注核心指标运营即可。

4.2 迭代架构

数据库相关技术发展很快，比如 Database Mesh、Serverless 等新技术形态会逐步落地，届时中间件、高可用、内核等会有比较大的变化，新型客户端 HA 方案的建设成熟及新 Proxy 架构，存计分离产品的引入都会使容灾的能力发生比较大的变化。容灾能力建设会随着这些确定的产品演进进行迭代。

容灾建设是一件非常有挑战的事，也是所有公司业务发展壮大后必须面对的一件事。欢迎大家在文末留言，跟我们一起交流。

MySQL 自治平台建设的内核原理及实践（上）

作者：裕锋

| **B 站视频**：美团数据库自治服务平台建设

1. 背景 & 目标

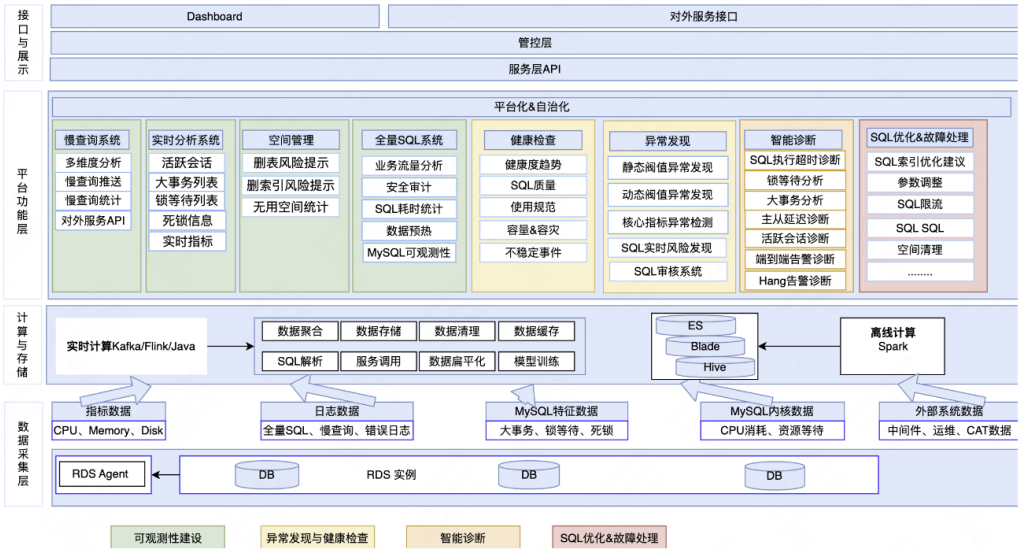
MySQL 的故障与 SQL 的性能，是 DBA 跟研发同学每天都需要关注的两个重要问题，它们直接影响着数据库跟业务应用程序的稳定性。而当故障或者 SQL 性能问题发生时，如何快速发现、分析以及处理这些问题，使得数据库或者业务系统快速恢复，是一项比较大的挑战。

针对此问题，美团数据库自治平台经过多轮的迭代建设，在多个场景下已经实现了异常的发现、分析以及处理的端到端能力。本文将跟大家分享一下我们平台建设的心路历程，同时提供一些经验、教训供同行参考，希望能够起到“抛砖引玉”的作用。本文主要介绍以下主题：

- **异常发现**：基于数理统计方式的动态阀值策略，来发现数据库系统的指标异常。
- **故障分析**：丰富完善数据库关键信息，来做精确的数据库异常根因分析；深入挖掘内核价值，来解决根因诊断方面的疑难杂症。
- **故障处理**：依据异常根因分析的不同结果，通过自助化或自动化的方式来进行故障的恢复处理。
- **内核可观测性建设**：如何跟数据库内核团队合作，从内核的角度来分析 SQL 性能问题，通过内核团队大量的内核代码改造，力求将数据库的可观测性跟诊断做到极致。
- **单 SQL 优化建议**：通过改造 MySQL 存储引擎，同时结合查询优化来打造基于 Cost 模式的索引优化建议。
- **基于 workload 索引优化建议**：基于整个 DB 或者实例的 Workload 策略的索引优化建议，为实现数据库的索引自维护提供前置条件。
- **基于 SQL 生命周期的治理**：实现从 SQL 上线前、执行过程中、执行完毕后几个环节，以期实现端到端的慢 SQL 治理。

2. 平台演进策略

美团数据库自治平台从下到上总体分为四层，分别为接口与展示、平台功能层，计算与存储、数据采集层，平台的总体架构以及每一层的作用如下：



- **数据库采集层**：要进行数据库的诊断与分析，需要依靠关键的指标以及 SQL 文本数据，当前在每个数据库实例上部署一个数据采集程序（rds-agent）统一负责采集、上报关键数值指标以及 SQL 文本数据。
- **数据计算与存储层**：数据采集层上报上来的数据，依托 Kafka、Flink&Spark 作为数据缓冲，对关键组件进行相关的数据处理，如 SQL 解析、SQL 模版化、数据聚合等操作，再把处理的结果存入 ES、Blade（美团自研的分布式数据库）、Hive 等分布式数据库或者大数据平台，提供给上层的平台功能层使用。
- **平台功能层**：此层是整个系统最为重要的部分，由于平台同时服务于 DBA 运维团队及研发团队，所以平台的建设分成了两条路：1）主要面向 DBA 用户，按照可观测性建设、异常发现、故障根因分析、故障处理几个阶段来进行建设；2）主要面向研发同学，按照 SQL 优化建议、风险 SQL 的发现、分析与 SQL 治理等跟 SQL 相关的几个阶段来建设。当然，两者并没有严格界限，这些功能所有的用户都可以同时使用。
- **接口与展示**：平台功能层提供的核心功能会通过 Portal 来展示，同时为了让平台

提供的功能更好地集成在用户自己的系统中，我们也通过 OpenAPI 的方式对外提供服务。

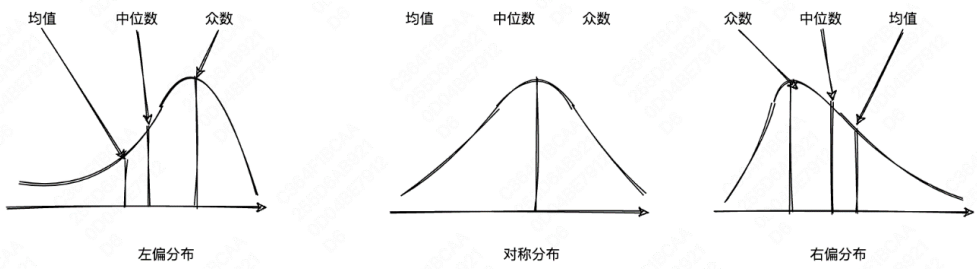
3. 异常发现

数据库产生异常时需要尽早地发现，才能防止异常一进步放大，避免造成真正的故障。异常发现的主要方式是对数据库或者 OS 的关键数值指标进行监控，相关指标包括 seconds_behind_master、slow_queries、thread_running、system load、Threads_connected 等等，也可以是业务端研发关注的关键指标，如“应用程序访问数据库的报错数量”、“SQL 执行平均耗时”等指标来进行监控。如果这些指标短时间内发生比较大的波动，那么数据库很可能出现了一些异常，这就需要及时进行处理。

这些异常如何才能被发现呢？业界一般有基于静态阈值以及动态阈值的两种异常发现策略。前者很简单，如根据专家经验，人工设定 seconds_behind_master 或者 Threads_connected 的告警阈值，超过阈值就认为发生了异常。此方式虽然简单易用，但 OLTP、OLAP 等不同的业务场景，对于相同指标的敏感度是不一样的，如果所有场景都使用统一的静态阈值来做异常发现，难免会有很多误告。而如果每个场景都去手工去调整，既不灵活，成本又太高，解决方案是基于不同场景的历史时序数据，使用数理统计的方式来分别建模，通过拟合出各自场景的模型来作为异常发现的策略。

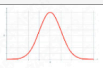
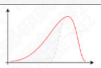
3.1 数据分布规律与算法选择

基于数理统计方法的异常发现，需要根据具体的场景使用特定的模型。一般来说，模型的选择跟时序数据的分布形态有很大的关系，时序数据的分布并不总是都像正态分布一样都是对称的，而是有些是左偏的，有些是右偏的，当然也有对称分布的。下图就展示典型的三种不同的时序数据分布形态。



针对上面的三种不同时序数据的分布形态，以及每种异常检测算法自身的特性，我们分别采用不同的异常检测算法。

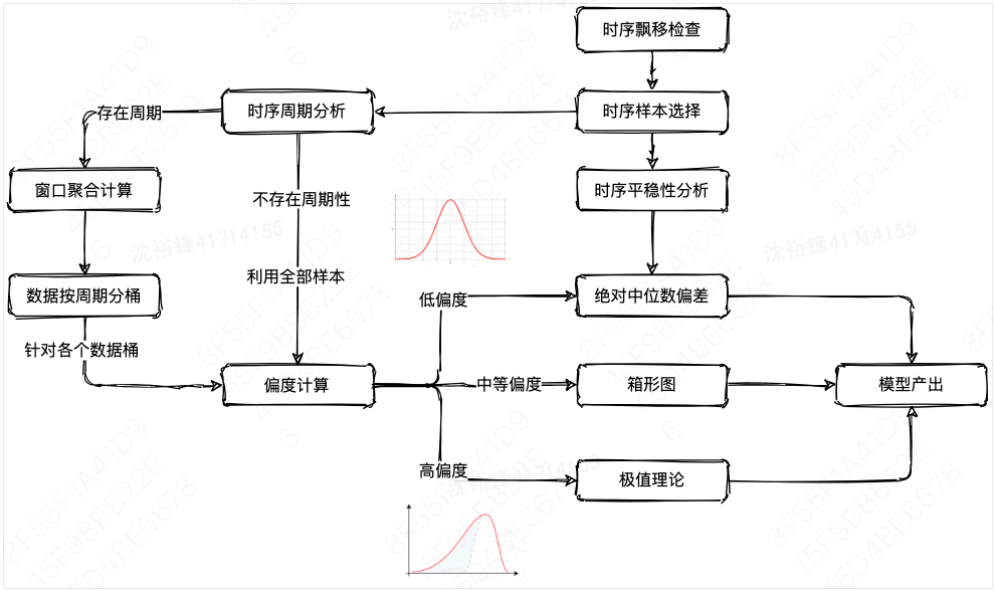
对于低偏态高对称分布选择“绝对中位差（MAD）”，中等偏态分布选择“箱形图（Boxplot）”，高偏态分布选择“极值理论（EVT）”。没有选择 3Sigma 的主要原因是：它对异常容忍度较低（建模的时候，如果有噪音等异常点也不会对模型的形态产生很大的影响，则说明异常容忍度很高），而绝对中位差（MAD）从理论上而言具有更好的异常容忍度，所以在数据呈现高对称分布时，通过绝对中位差替代 3Sigma 来进行检测。

算法\场景	对称分布下适用性	偏态分布下适用性	正态性要求	异常容忍度
图示			/	/
3Sigma	高	低	高	低
绝对中位差（MAD）	高	低	高	高
箱形图（Boxplot）	高	中	中	高
极值理论（EVT）	中	高	低	低

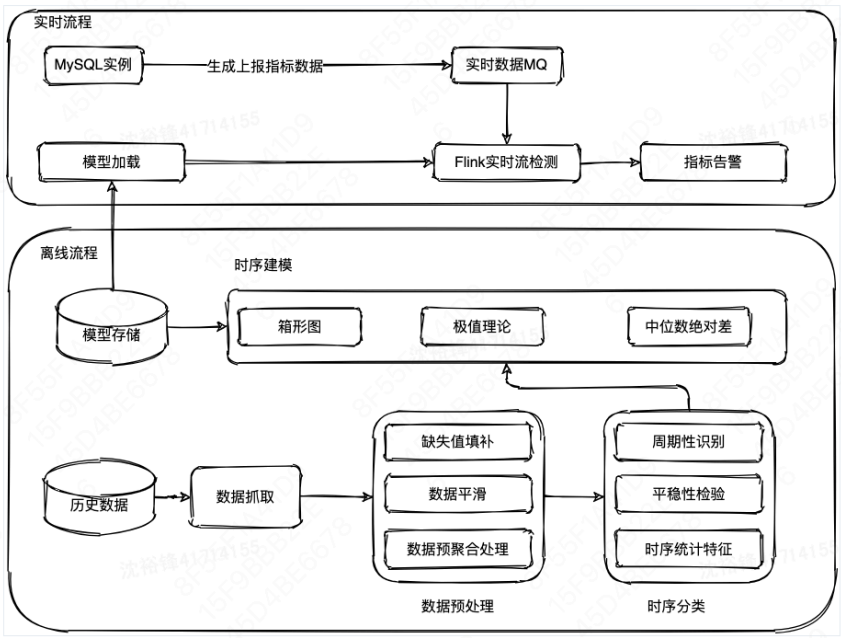
3.2 模型选择

数据分布跟算法适用场景的分析之后，对内部的时序数据进行检查，发现数据的规律主要呈现漂移、周期和平稳三种状态，对样本先进行时序的漂移（如果检测存在漂移的场景，则需要根据检测获得的漂移点 t 来切割输入时序，使用漂移点后的时序样本作为后续建模流程的输入）。

之后同时进行平稳性分析（如果输入时序 S 满足平稳性检验，则直接通过箱形图或绝对中位差的方式来进行建模）以及周期分析（存在周期性的情况下，将周期跨度记为 T，将输入时序 S 根据跨度 T 进行切割，针对各个时间索引 j ∈ {0,1, ,T 1} 所组成的数据桶进行建模流程。不存在周期性的情况下，针对全部输入时序 S 作为数据桶进行建模流程），再对时序数据分布特性进行偏度的计算，最后再根据不同的偏度特性选择不同的算法模型，具体如下：

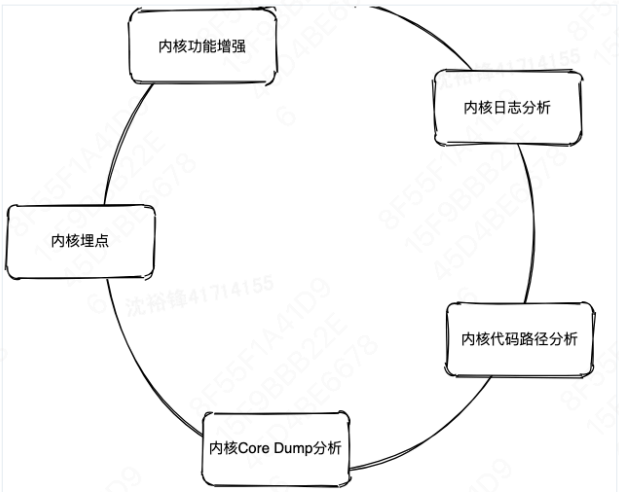


在算法确定之后，先在离线环境针对不同的场景使用历史指标来训练模型，模型训练完毕之后会存入数据库，在生产环境运行过程中，对于不同场景下的数值指标根据其特性来加载不同的模型，并且结合 Flink 实时计算框架来实时的发现指标的异常并进行告警。



4. 异常诊断

发现指标异常后，需要快速的给出异常的根因，我们可以根据具体的根因来选择不同的处理策略，然后进行自动或者手动的恢复工作。根因分析可以基于专家经验，也可以严格地按照内核代码的逻辑来进行分析。



不同类型的数据库内核思维诊断范式

本文重点讲述后者，强调如何使用“内核思维”来解决专家经验很难或者无法解决的诊断问题。本文将列举“内核代码路径分析”、”内核日志分析”、“内核功能增强“、“内核 Core Dump 分析”以及“内核埋点”等几种不同的范式，来说明数据库根因诊断的思路。

4.1 主从延迟（内核代码路径分析）

这里先介绍“内核代码路径分析”这个方式来诊断根因。对于数据一致性要求比较高的应用程序，seconds_behind_master 是一个十分重要的指标，如果其值过大就需要诊断其根因，防止应用程序读取到不一致的数据。根据专家经验，其值过大可能由“QPS 突增”、“大事务”、“大表 DDL”、“锁阻塞”、“表缺少主键或者唯一健”、“低效执行计划”、“硬件资源不足”等因数造成，把这些专家经验总结成规则列表，当异常产生时逐个迭代去验证是不是符合某个规则，据此来诊断根因，然而此方式存在如下两大问题：

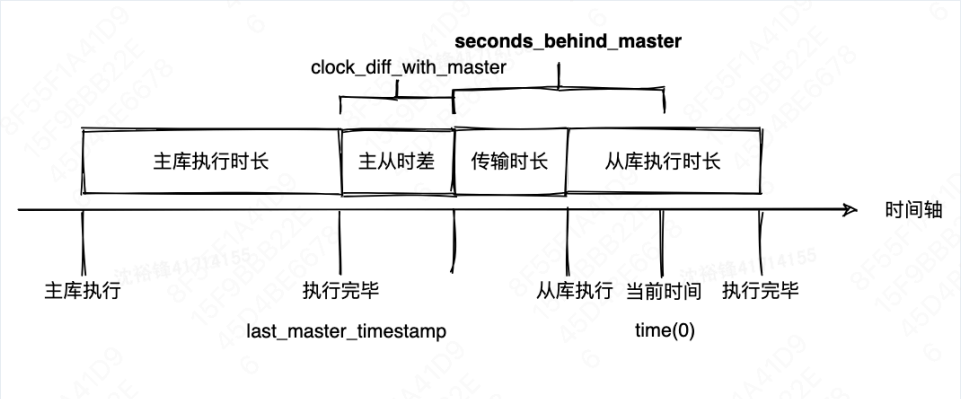
1. **无法枚举所有根因**：经验由于其固有的局限性不可能考虑到所有的故障场景，如何完整的给出造成 seconds_behind_master 值异常的所有规则是一个挑战；另外，如果需要对一个全新的指标进行诊断，而在没有任何的专家经验情况下，如何能快速地整理出完整的规则列表?
2. **缺乏对根因的深层次理解**：“QPS 突增”、“大事务”、“大表 DDL”、“锁阻塞”、“低效执行计划”、“硬件资源不足” 等因素会造成 seconds_behind_master 指标值的异常，但是为什么这些因数会造成指标值的异常呢? 如果不从内核源码角度来了解这些因素跟 seconds_behind_master 之间的逻辑计算关系，那么是无法了解真正原因的。

4.1.1 内核代码路径分析

针对上面两个问题，具体策略是直接看 seconds_behind_master 这个变量在内核层面是如何计算的，只有这样才能完整的枚举出所有影响 seconds_behind_master 值计算的因数。

从源码角度看，seconds_behind_master 的值由① time(0)、② mi->rli->last_master_timestamp 和③ mi->clock_diff_with_master 这三个变量来决定(代码层面 seconds_behind_master 的计算逻辑是：seconds_behind_master=((long)(time(0) – mi->rli->last_master_timestamp)– mi->clock_diff_with_master)，其中 time(0) 是系统当前时间(用秒表示)，clock_diff_with_master 这个值的计算很复杂、又很关键，会放到下一节详细进行说明。

而针对 mi->clock_diff_with_master 的计算，这个变量从源码层面看就是主、从实例之间的时间差；根据当前的信息就可以看出来，从库的当前时间以及主从库之间的时间差都会影响 seconds_behind_master 值的计算。seconds_behind_master 的计算和事务在主从库执行的情况如下：



last_master_timestamp 计算逻辑

从上面分析可以知道，last_master_timestamp 值是影响 seconds_behind_master 值计算的关键变量，所以很有必要从源码角度分析影响 last_master_timestamp 值的因数有哪些（从而间接获取了影响 seconds_behind_master 值的因素）。

last_master_timestamp 的值来自于一个叫 rli->gaq->head_queue() 的成员变量 new_ts（此处的 rli->gaq->head_queue() 是指代某个最新的已经完成 replay 的事务对应的 event group，event group 是指一个事务在 binlog 文件里生成一组 event 来表示某个事务，这个 event group 里的 event 从主库传输到从库后进行 replay 操作来还原主库的事务）。new_ts 值来源于 rli->gaq->head_queue())->ts，而 rli->gaq->head_queue())->ts 的值是通过 ptr_group->ts= common_header->when.tv_sec + (time_t) exec_time 计算获取的。

再看一下 when.tv_sec 跟 exec_time 的含义，前者指代 SQL 在主库上的 SQL 执行的开始时间，后者指代 SQL 在主库上的“执行时长”，“执行时长”又跟“锁阻塞”、“低效执行计划”、“硬件资源不足”等因素息息相关。

值得注意的是，前面提到的 rli->gaq->head_queue())->ts 的计算跟 slave_checkpoint_period 以及 sql_delay 两个变量也有关系，按照这个思路层层迭代下去找出所有影响 seconds_behind_master 值的因素，这些因素都是潜在的主从延迟异常的根源，这样也解决了前面说的“无法枚举所有根因”跟“缺乏对根因的深层次理解”两大问题。

为了便于理解上诉的逻辑，放出关键的源代码：获取 last_master_timestamp 值的来源 rli->gaq->head_queue() 的成员变量 new_ts。

```
bool mts_checkpoint_routine(Relay_log_info *rli, ulonglong period,
                            bool force, bool need_data_lock)
{
 do
  {
    cnt= rli->gaq->move_queue_head(&rli->workers);
  }
 ......................
  ts= rli->gaq->empty()
    ? 0
    : reinterpret_cast<Slave_job_group*>(rli->gaq->head_queue())->ts; //其中的
ts 来自下面的 get_
slave_worker 函数;
  rli->reset_notified_checkpoint(cnt, ts, need_data_lock, true);
  //  社区版本的代码 rli->reset_notified_checkpoint(cnt, rli->gaq->lwm.ts,
need_data_lock);
  /* end-of "Coordinator::"commit_positions" */
 ......................
}
```

获取 Master 实例上执行的 SQL 的开始跟执行时长信息 tv_sec 跟 exec_time。

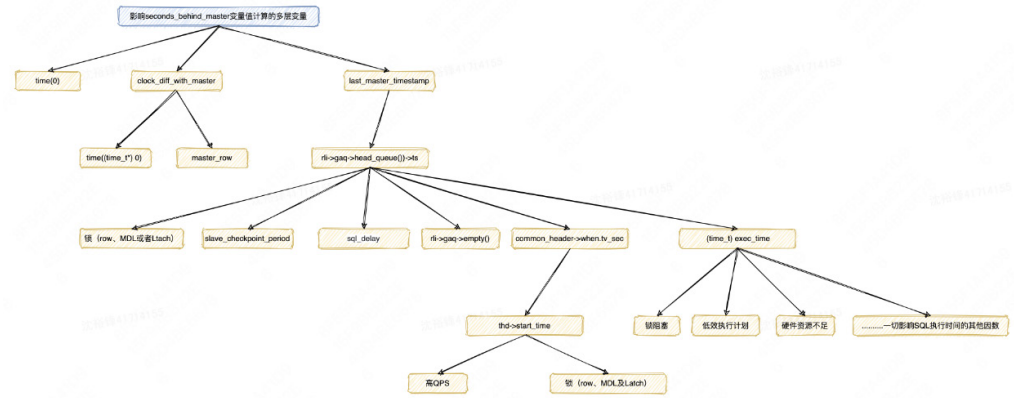
```
Slave_worker *Log_event::get_slave_worker(Relay_log_info *rli)
{
if (ends_group() || (!rli->curr_group_seen_begin && (get_type_code() == binary_
log::QUERY_EVENT
|| !rli->curr_group_seen_gtid)))
  {
  ..............
    ptr_group->checkpoint_seqno= rli->checkpoint_seqno;
    ptr_group->ts= common_header->when.tv_sec + (time_t) exec_time; //
Seconds_behind_master related
    rli->checkpoint_seqno++;
  }
}
```

根因层叠图

如果进一步分析内核代码，可以发现影响 seconds_behind_master 变量计算的因素还有很多，但是找出这些因素的思路是相同的。这个思路的好处是：无论之前有没有相关专家经验，理论上这种分析方式都能尽可能地枚举出所有的根因。

除了 seconds_behind_master，其他的像 thread_running、Threads_connected，slow_queries 等指标异常的分析也都可以套用这种思路。下面为按照上述思路整理出

来的影响 seconds_behind_master 值的部分因素的层次结构图：

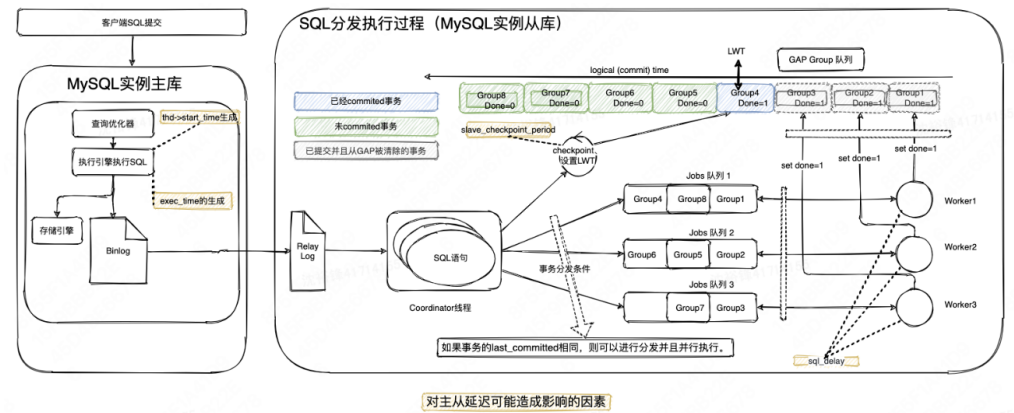


影响seconds_behind_master变量值的部分因素

4.1.2 流程分析

把影响 seconds_behind_master 值的相关因素确认后，可以画一个流程图把这些因素都体现在流程图中的具体位置。这样既能比较形象地理解影响 seconds_behind_master 的因素在整个主从复制流程中的所处的位置，又便于对整体规则进行查漏补缺。

下面我们使用一个具体的例子，来说明一下上面分析的因素是如何影响 seconds_behind_master 的。从下图可以看出在执行 SQL 的过程中影响 seconds_behind_master 计算的两个变量 thd->start_time 跟 exec_time 的计算在 master 实例。假设 start_time 的值为 2023-07-03 00:00:00，SQL 执行了 60 秒，所以 exec_time 为 60，2023-07-03 00:01:00，SQL 在主库上执行完毕，在从库上 replay 这个 SQL，可以看到 seconds_behind_master 值会从 0 开始并且逐渐增加 60 秒，然后再返回 0。



对主从延迟可能造成影响的因素

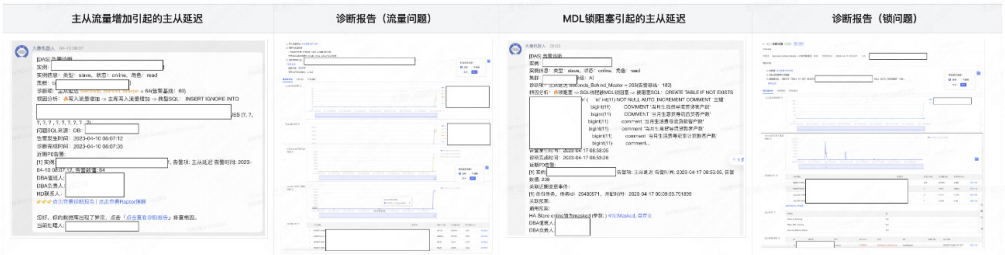
具体原因是：假设我们忽略 binlog 日志的传输时间，那么从库开始执行 replay 这个 SQL 的开始时间也是 2023-07-03 00:01:00，所以根据 seconds_behind_master=((long)(time(0) - mi->rli->last_master_timestamp)- mi->clock_diff_with_master）=2023-07-03 00:01:00 - 2023-07-03 00:00:00-60s，结果就是 0，然后 SQL 的执行时间是 60 秒，并且 (long)(time(0)（当前时间）的时间一秒一秒的在增加，所以 seconds_behind_master 值会从 0 开始逐渐增加至 60 秒。

再看一下其他的因数，协调器（Coordinator）会把 Group 放入一个叫做 GAP Group 的队列中，Coordinator 会以 slave_checkpoint_period 值为周期来扫描 GAP Group 中的元素并且更新 rli->gaq->head_queue())->ts 值，所以果 slave_checkpoint_period 的值被设置的很大，那么 rli->gaq->head_queue())->ts 的值因为没有及时更新而变得比较旧，从而引起 seconds_behind_master 值变大。

另外，每个 Worker 读取自己队列的 Group 元素，进行 repaly 操作，需要注意的是 sql_delay 这个变量，如果当前时间还没有达到 sql_delay 规定的时间（假设 sql_delay 被设置为 100 秒，那么 SQL 对应的 binlog 日志到达从库后需要等待 100 秒再执行），那么 worker 就不会进行 repaly 工作，这间接导致影响计算 seconds_behind_master 变量 thd->start_time 值比正常情况下小了 100 秒，所以当 worker 进行 replay 的时候，seconds_behind_master 的值就会相应的增加 100 秒。

4.1.3 产品展示

下面的产品展示了因为流量突增跟 MDL 锁造成的主从延迟的诊断分析报告的产品页面。我们可以看到，流量突增的具体 SQL 以及 MDL 锁的持有者，方便用户进行限流或者 Kill 掉阻塞者 SQL。



4.2 大事务诊断分析（内核功能增强）

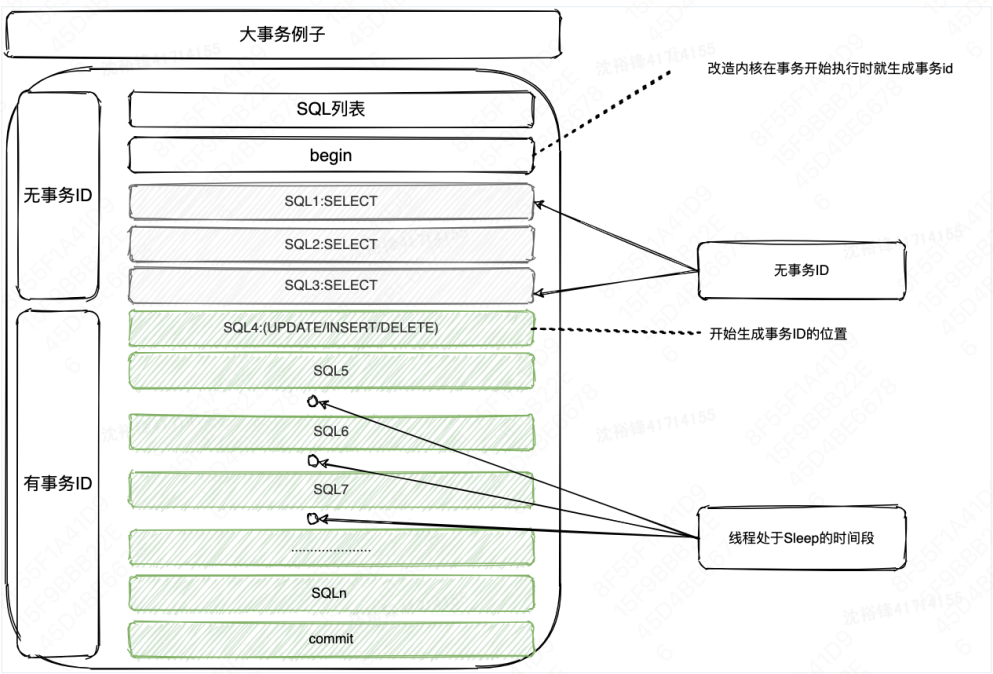
大事务的存在，对整个数据库系统的稳定性与总体 SQL 的性能都会产生很大的挑战，如大事务长时间持有某个锁会造成大面积阻塞，或者更改过多的行数造成整个实例硬件资源的不足。我们需要及时发现这些场景，并且将其信息发送给用户治理，但在实践过程中，往往面临如下的两大挑战：

第一个挑战：无法得到大事务所包含的完整的 SQL 列表，造成用户不清楚大事务的全貌是什么，用户也就无法识别需要优化的大事务。

- **解决方案**：每个事务来 MySQL 会在内核层面生成一个唯一的事务 ID：trx_id，如果事务包含的每条 SQL，都给其附加一个事务 ID trx_id 字段，并且把这些 SQL 连同 trx_id 一起输出（通过全量 SQL 输出），问题就可以解决；不过这里还有一个挑战，这个事务 ID 到底是何时产生的呢？如何大家熟悉内核内部事务的执行过程，就会知道事务 ID 的只有在事务修改数据的时候才会通过 trx_assign_id_for_rw 这个方法被获取，这意味着就下面这个图上展示的事务而言，是无法获取 SQL4 之前执行的读 SQL 语句列表，所以获取到的事务的 SQL 列表还是残缺的，那么如何获取到完整的 SQL 列表呢？解决方案也很简单，可以把事务 ID 的生成逻辑提前到在事务刚开始执行的时候生成就可以了。

第二个挑战：大事务的耗时组成不明确。数据库规定执行时长大于某个阈值的事务被定义为大事务，但是不清楚耗时到底是 SQL 本身的执行时间还是 SQL 执行之外的耗时，SQL 执行之外的耗时可能是在执行上下两个 SQL 之间，业务端在处理一些跟数据库无关的业务逻辑，也可能是网络延迟造成的。

- **解决方案**：上述问题可以通过在数据库内核内对 SQL 执行开始时、结束时分别埋点时间戳来解决，这样整个大事务执行总时间中有多少时间是在执行 SQL，有多少时间是在 Sleep 就一目了然；当然，这一块还可以做的更加的细致，比如两条 SQL 之间的 Sleep 时间到底是网络延迟还是应用程序端的延迟等，可以进一步细分大事务造成的原因到底是在 MySQL 端、网络端还是用户自己的应用程序造成的等待；关于计算网络端的延迟计算，可以参考 MySQL 内部的 mysql_socket_send_time 跟 vio_socket_io_wait_time 这 2 个关键指标的实现思路，下图是一个大事务的 SQL 列表以及耗时组成列表。



4.2.1 产品展示

内核团队通过内核改造，对事务中所包含的 SQL 都提供了 trx_id 后就可以根据 trx_id 把整个事务所有的 SQL 串起来。根据 SQL 执行的开始跟结束时间，我们也提供了所有 SQL 之间的 Sleep 时间，成功解决了上面的两个挑战，产品效果图如下：



4.3 MySQL Crash 分析（内核 Core Dump 分析）

MySQL 实例突然 Crash 了怎么进行根因诊断？进程 Crash 的根因分析，也是数据

库故障中最难分析的问题之一。本节提供一些思路尝试去分析各种场景下的 MySQL Crash 的根因。

4.3.1 Crash 的触发方式

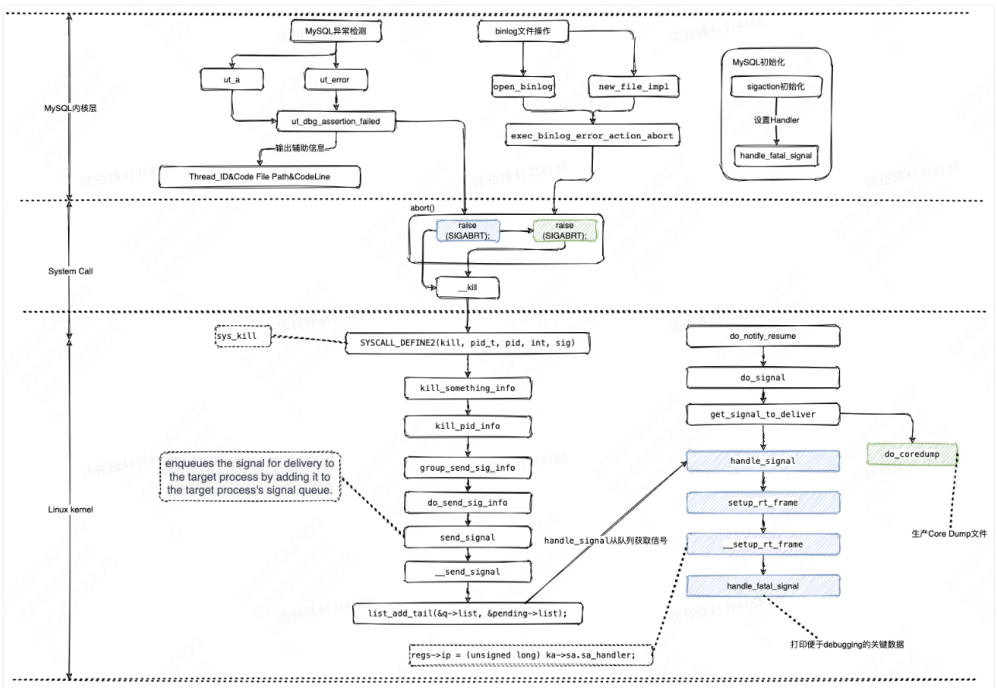
在分析 Crash 的根因之前，我们可以先了解一下 MySQL 进程是如何被 Crash 的整个过程十分有必要。一般来说，触发 Crash 的原因分成两类：① MySQL 进程自己触发了 Crash（这里称之为 MySQL 自杀）；② MySQL 进程被 OS 杀死。

针对前者，比如 MySQL 发现某个关键数据发生了 Data Corruption、磁盘空间不足、硬件错误、等待内核锁时间过长、MySQL 内核 Bug 等场景，都可能导致 MySQL 自杀。尤其是检查到 MySQL 内核里有些数据的状态不符合预期时，是必要要让那个实例 Crash 也不能继续执行，否则可能会产生更加严重的数据不一致性问题；而后者，OS 如果发现系统内存严重不足或者空指针引用等情况，就会把包括 MySQL 在内相关的进程杀掉。

分析一下 MySQL 自身触发 Crash 这个场景，在 MySQL 内部有很多地方通过 ut_a（如果是 ut_error 的话，则直接触发 Crash 断言对程序的内部数据状态进行异常检查，如果发现数据状态不符合预期，那么势必发生了 Data Corruption，这个时候程序会调用 ut_dbg_assertion_failed 在进程 Crash 之前做一些关键信息（如 thread id、发生 Crash 的文件名字跟 code line 等）的记录后，会继续调用 abort() 向进程发送 SIGABRT 信号量让进程 Crash 掉。

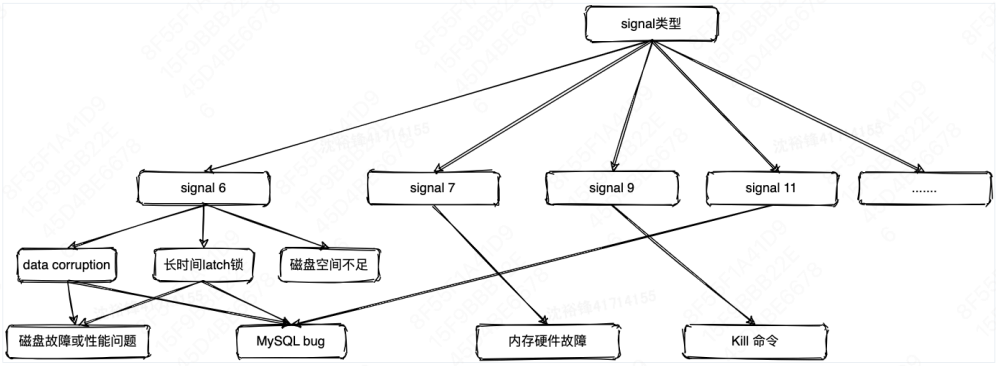
需要注意的是，从 abort() 的源码可知，这里调用了两次 raise (SIGABRT)，第一次调用 raise (SIGABRT) 触发处理函数 handle_fatal_signal（此函数在 MySQL 实例初始化时通过 sigaction 的 sa_handler 注册）被调用，这个函数主要是打印一些重要的调试信息，便于做 Core Dump 分析；第二次调用 raise (SIGABRT) 的目的，是为了让 OS 生成 Core Dump 文件（core Dump 文件非常重要，因为所有引起 MySQL Crash 的现场信息都包含在 Core Dump 里面）；如果是在 MySQL 自杀的情况下发生了 Crash ，一般会在 errorlog 里会产生如下的一段跟 Crash 相关的现场信息，其中的“signal”、“触发 Crash 的线程堆栈”、“正在执行的 SQL”等信息都是分析 Crash 根因的关键信息。

下图为 MySQL 通过 ut_a 断言检查异常问题后再到 OS 触发进程 Crash 的整体流程图。总体来说，MySQL 通过 raise 来发送 SIGABRT 信号到一个队列，OS 内核处理这个队列的信号并调用 MySQL 的处理程序 handle_fatal_signal 来保留一些关键的现场信息。这里需要注意到的是，OS 内核在 __setup_rt_frame 中执行“regs->ip = (unsigned long) ka->sa.sa_handler;”，这个步骤正是让 MySQL 的 handle_fatal_signal 方法被顺利的调用的原因。



4.3.2 根据 Signal 类型做根因分析

分析 Crash 根因的第一步就是看触发了什么类型的 signal，常见类型有“signal 6”、“signal 7”、“signal 11”几种类型，当知道了 Signal 类型后就有一个根因分析的大方向。根据经验，我们将常见的 signal 类型以及可能引起的原因大致如下图所示，下面对于常见的 signal 类型以及引起的根因做一个简单的分析。



(1) 如果是 signal 6，一般是实例的磁盘空间不足或者磁盘只读，MySQL 的数据发生了 data corruption，内核层面 latch 锁的长时间的锁等待造成。不过这里的 data corruption、长时间的锁等待可能是硬盘损坏或者 MySQL 的 Bug 造成的，判断逻辑如下：

1. 磁盘空间不足或者磁盘只读

- 磁盘写数据时，如果磁盘没有剩余空间或者数据库被设置为 read only 就会造成实例的 Crash，日志中有“Either disk is full or file system is read only while opening the binlog”的字样。

2. data corruption

- MySQL 在运行过程中如果断言（比如这里 ut_a(table != NULL) 返回 false，很可能是数据发生了 corruption 并且 MySQL 就会自行 Crash 掉；发生 data corruption 时一般在 MySQL 的 error log 中有“Database page corruption on disk or a failed file read of tablespace”的字样，所以查看日志来判断否有硬盘故障问题，如果没有硬件故障信息，则可能是 MySQL Bug 造成的 data corruption，具体分析看下面“MySQL Bug”那部分。

3. 长时间无法获取 Latch 锁

- 如果 MySQL 长时间没有办法获取到 Latch 锁，那么 MySQL 认为系统可能是 Hang 住了也会引起实例的 Crash，并且日志中打印“We intentionally crash the server because it appears to be hung”字样，一般是硬件故障造成的性能问题或者 MySQL 自身的设计缺陷形成的性能问题造成的，这次场景根因分析比较有挑战。

4. MySQL Bug

- 如果不属于上面任何一种情况，那么有可能是 MySQL 自身的 Bug 造成的 Crash，比如 MySQL 对一些 SQL 解析或者执行时会发生 Crash；这种场景一般先看一下 Crash 发生时正在执行的 SQL 是什么，这个 SQL 可能存在于 Crash 日志中（这个 Crash 日志中有个例子），可以先把 SQL 提取出来再次执行查看能否复现问题；如果在 Crash 日志中看不到 SQL 语句，就需要从 core dump 文件中提取 SQL 了，提取方式是 MySQL 每个链接对应的 THD 的成员变量 m_query_string 就包含了 SQL 文本，只需要打开 Core Dump 文件切换到某个包含 THD 实例的方法内，通过命令“p this->thd->m_query_string.str ”来打印，这里有个例子。
- 再举个 MySQL 的 Bug 造成 data corruption 的例子，从 Crash 日志里“InnoDB: Assertion failure in thread 139605476095744 in file rem0rec.cc line 578”看出，从 rec_get_offsets_func 函数中触发 ut_error 而导致的 Crash，之所以触发这个 Crash 是因为 rec_get_offsets_func 中的 rec_get_status(rec) 获取到的 MySQL 的记录类型不符合预期（因为记录类型只有固定的 REC_STATUS_ORDINARY、REC_STATUS_NODE_PTR，REC_STATUS_INFIMUM，REC_STATUS_SUPREMUM 这 4 种类型），如果内核发现一个记录类型不属于这 4 种类型的任何一种，那么就是发生了 data corruption，所以必须要把自己 Crash 掉。为了验证刚才的结论，看一下 Crash 发生时的 rec 的类型是什么，从源码可知 rec 的类型是通过 rec_get_status 获取，并且其通过调用的 rec_get_bit_field_1 跟 mach_read_from_1 两个函数可以知道 rec 的类型其实就是 rec 这个指针往前三个 byte（通过 #define REC_NEW_STATUS_MASK 0x7UL 可知）代表的值。
- 通过 gdb 加载 core dump 文件后，切换到抛出 exception 的线程，因为异常是在 rec_get_offsets_func 里抛出的，切换到 rec_get_offsets_func 对应的 frame 7 来验证 rec 的类型，看到 rec 的指针地址为 0x7f14b7f5685d（相关分析数据可以看此链接）。前面说过，rec 的类型值在 rec 指针往前三个 byte 里，也就是指针 0x7f14b7f5685a（0x7f14b7f5685d-3）那个位置的值，发现是 0x1f，执行与计算（11111(1f)&00111(0x7UL)=00111=7 ）得到的类型是 7，而记录类型的范围是（0 ~ 3），很明显这个指针指向的记录类型值信息发生了 data corruption（分析

过程查看此链接），这里做了一个 rec 的类型在正常情况下跟本例异常情况下的类型值计算的对比表，发现正常情况下，rec 的类型值就应该是 3。

	Record status	正常情况下的Record Type	异常情况下的Record Type
Record的Type	/* Record status values */ #define REC_STATUS_ORDINARY 0 #define REC_STATUS_NODE_PTR 1 #define REC_STATUS_INFIMUM 2 #define REC_STATUS_SUPREMUM 3 // 正常值的范围为0~3	rec = (const rec_t *) 0x10f5b4070 "supremum\U00000001" (0x10f5b4070-3)=0x10f5b406d (lldb) x/1xb 0x10f5b406d 0x10f5b406d: 0x0b 01011(0b)&00111(0x7UL)=00011=3	(gdb) x/1xb 0x7f14b7f5685d 0x7f14b7f5685d: 0xce (gdb) x/1xb 0x7f14b7f5685a / /rec-3 0x7f14b7f5685a: 0x1f 11111(1f)&00111(0x7UL)=00111=7 //7异常

- 这里有个重要问题是，为什么 rec 的类型是无效的呢？很可能是 MySQL 搜索满足条件的记录的时候，rec 指向的记录很可能被 page_cleaner 在后台被清理掉了，所以 rec 指针指向的记录就是无效了。官方有个 bugfix，解决方案就是把 prev_rec 设置为 NULL（这里的 prev_rec 是 persistent cursor 指向的记录，这里说一下 persistent cursor，它是 MySQL 从 InnoDB 层获得记录后进入 SQL 层前在 B-tree 上的 cursor 会被暂时存储到 row_prebuilt_t::pcur 中，当再次从 InnoDB 层拿数据时，如果对应的 buf_block_t 没有发生修改，则可以继续沿用之前存储的 cursor，否则需要重新定位，如果没有 persistent cursor 则每次都需要重新定位则对性能有影响），这样 prev_rec != NULL 这个条件不满足，也就没有机会进入 rec_get_offsets_g_func 里去检查 rec 的类型值而引发 Crash 了。

（2）如果为 signal 7，那么大概率是内存硬件错误，并且日志里一般有“mce: [Hardware Error]: Machine check events logged，mce: Uncorrected hardware memory error in user-access at xxx MCE xxx: Killing mysqld:xxx due to hardware memory corruption”等字样。

（3）如果为 signal 9，表示这个进程被 Kill 命令杀掉了。

（4）如果为 signal 11，表示是由 MySQL 的 Bug 造成的，这类问题较难分析特别是 MySQL Crash 现场（通过 core dump 打印出来的堆栈信息）往往还不是第一现场，由于篇幅关系具体的例子分析不在本文中给出，但是分析的思路跟上面的“MySQL Bug”是类似的。

5. 本文作者

裕锋，来自美团基础研发平台 – 基础技术部，负责美团数据库自治平台的相关工作。

6. 参考

- https://github.com/shenyufengdb/sql
- https://github.com/percona/percona-server/blob/release-5.7.41-44
- An Efficient, Cost-Driven Index Selection Tool for Microsoft SQL Server
- plan-stitch-harnessing-the-best-of-many-plans-2
- Random Sampling for Histogram Construction: How Much is Enough?
- AutoAdmin “what-if” index analysis utility
- What is a Self-Driving Database Management System?
- https://www.microsoft.com/en-us/research/publication/self-tuning-database-systems-a-decade-of-progress/
- Automatic Database Management System Tuning Through Large-scale Machine Learning
- Query-based Workload Forecasting for Self-Driving Database Management Systems
- The TSA Method
- https://blog.langchain.dev/langchain-chat/
- https://github.com/hwchase17/langchain
- REAC T: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS
- Evaluating the Text-to-SQL Capabilities of Large Language Models
- SQL-PALM: IMPROVED LARGE LANGUAGE MODEL ADAPTATION FOR TEXT-TO-SQL

MySQL 自治平台建设的内核原理及实践（下）

作者：裕锋

| **B 站视频**：美团数据库自治服务平台建设

0. 前文回顾

在《MySQL 自治平台建设的内核原理及实践（上）》一文中，我们主要介绍了数据库的异常发现与诊断方面的内容，在诊断方面经常会碰到一些难以找出根因的 Case。针对这些疑难杂症 Case，通过本篇可以了解到，如何通过内核可观测性以及全量 SQL 来解决这些问题。除此之外，在得出根因后，如何处理异常，如何对 SQL 进行优化，以及如何进行 SQL 治理等相关方面问题，也将在本篇中给予解答。

1. 内核可观测性建设

1.1 内核可观测性建设

1.1.1 性能诊断挑战

在自治性能诊断平台的建设过程中，我们发现如下两大挑战：

1. 很多 SQL 性能抖动的问题找不出根因，比如 SQL 的执行时长莫名其妙的突然变大，其执行计划良好、扫描跟返回的行数都很少，也没有行锁、MDL 锁相关锁阻塞；查看慢查询日志，也没有哪个字段的耗时比较高，但是 SQL 的执行时长就是突然变长，有时候达到几十秒长，而平时往往是几毫秒，各种分析后找不出原因。
2. 有时候在诊断一些指标异常的根因时，凭借的是不太严格的经验，而不是量化分析，比如 thread_running 或者 slow_queries 值突然升高的时候，可能会通过表 information_schema.processlist 查看当前的活跃会话中线程的状态，看一下状态是不是有行锁或者 MDL 锁之类的阻塞，或者通过慢查询日志里的相关数据来诊断根因。这里的挑战是：我们看到的是整个 SQL 某个时间点的瞬时状态，或者只是整个 SQL 执行过程中的部分数据，而不是整体的数据，所以得出

的根因诊断可能是片面的，也许一瞬间看到的是行锁，但是大部分时间被 MDL 锁阻塞。

1.1.2 解决思路

如果使用的是社区版本的 MySQL，基本上都会面临上面两大问题。我们先从内核的角度分析一下这两个挑战，对于第一个挑战，主要是对 MySQL 在内核层面执行细节不够了解，比如一条 SQL 执行了 10s，而从内核层面来看的话，这十秒的时间可能会有几百个步骤组成，检查后可能发现 row 或者 MDL 锁等待时间加起来只有 1 秒，那么其他的 9 秒的耗时在哪里呢？可能发生在这几百个步骤中的任何一个或者多个，所以如果没有这几百个步骤的明细数据，就无法诊断出突然出现的性能问题的根因。

第二个问题跟第一个问题从本质上来说是一样的。由于采集的数据是某个时间点的快照数据（通过活跃会话），或者只是部分指标的数据（通过慢查询日志），所以我们看到的只是片面的信息，而没有办法获取到整个 SQL 的完整的耗时分布信息。

1.1.3 Wait 耗时量化分析法

在分析完原因之后，我们参考了 TSA 的思想，同时结合 MySQL 自身的特点来做基于 Wait 的内核可观测性的建设。从 TSA 可以看出，SQL 执行的总耗时无非就是由 Off-CPU 跟 ON-CPU 两大部分组成，如果 SQL 有耗时长的问题，首先要确认是在 OnCPU 还是在 OffCPU 上耗时多。如果是前者，那么说明 SQL 本身有问题，比如消耗资源太多（比如无效索引造成扫描行数过多）；如果是后者，那么说明 SQL 本身没有问题，而是受到干扰或者系统资源不足，进而造成 OffCPU 层面耗时过多。

无论是何种情况，我们都需要把 OnCPU 跟 OffCPU 的耗时做进一步的拆分，从而来查看耗时瓶颈点到底在哪里，并且对耗时比较多的关键代码路径交由内核团队来进行埋点；相对 MySQL performance_schema 库中统计的 Wait 信息，实现非常轻量，所以对总体的性能的影响很小，并且耗时指标都是 SQL 明细粒度的，而不是类似 performance_schema 中的 SQL 模版级别的聚合数据。

耗时在 OnCPU

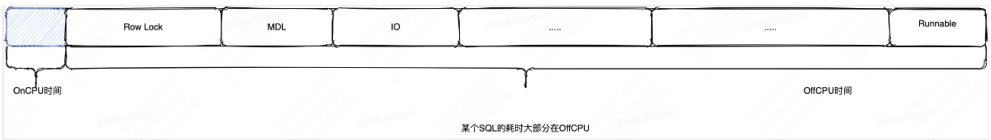
在分析一条 SQL 执行耗时，如果发现像下图一样，耗时分布大部分都在 OnCPU，这说明 SQL 本身有严重的性能问题（全表扫描、过滤效果不佳或者查询优化器的 bug

等原因），我们可以把诊断的方向集中在如何优化 SQL 本身就可以了，而不需要去关注其他的方面，如锁阻塞、磁盘或者网络延迟等 OffCPU 方面的问题。通过使用 getrusage 方法来获取某条 SQL 的 OnCPU 耗时时长，比如在 SQL 执行前，获取当前线程的 ru_utime、ru_stime 时间，然后在 SQL 执行完毕时再次获取 ru_utime、ru_stime 值，就可以得到 SQL 执行的 CPU Time。



耗时在 OffCPU

如果发现 OnCPU 的耗时占总耗时的比率比较低，通过 OffCPU 的相关指标发现其占总耗时的比率比较高，说明 SQL 本身没有问题，可能是被锁住了、硬件资源不足或者是被内核层面某个内核锁给卡住了。我们就需要查看到底是 OffCPU 相关的哪个，或者哪些指标耗时比较高，需要在内核层面对执行过程中的可能的、耗时比较长的执行代码路径进行埋点，只有获取到了跟性能相关的关键的数据，才能做出准确的判断。



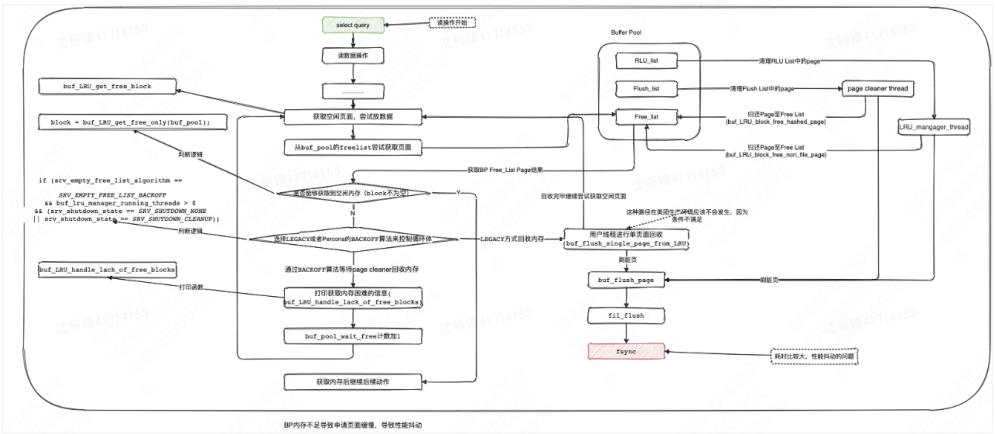
如何选择合适而全面的 OffCPU 相关的代码路径进行埋点？经过探索，这里提供了如下几个方式：

1. 分析 setup_instruments 表中包含的关键埋点信息，大致知道有哪些关键的指标可以埋点，比如 wait/IO、Mutex 类等，再结合自身的经验来判断哪些指标可能有性能瓶颈的问题。
2. 根据实际疑难 case 来判断选择哪些 Wait 指标，比如看到”Thread xxx has waited at trx0trx.cc line 1193 for 241.00 seconds the semaphore: xxxxx mutex REDO_RSEG created trx0rseg.cc:211, lock var 1”，则对 mutex REDO_RSEG 进行埋点，看到”Thread xxx has waited at dict0dict.cc

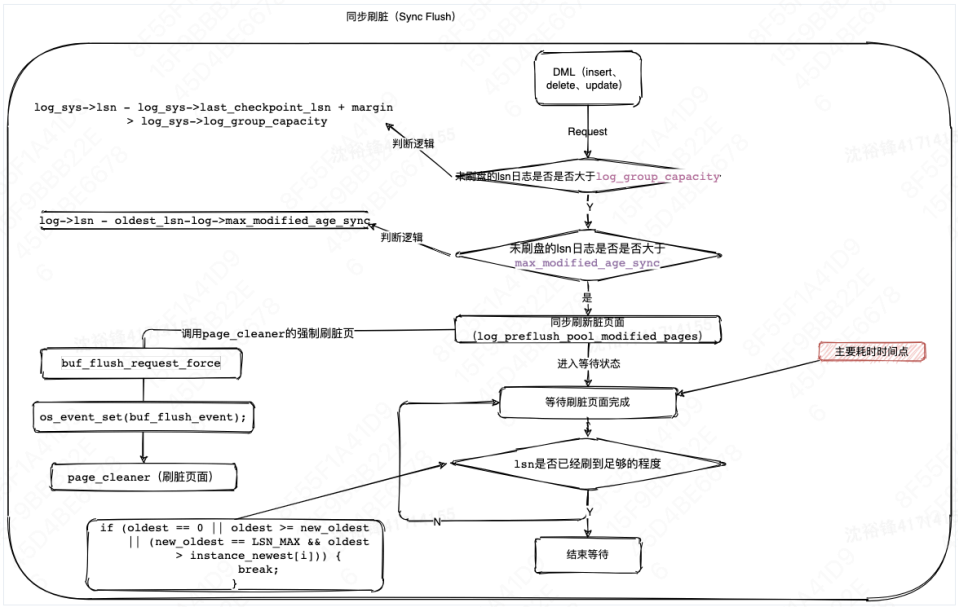
line 1239 for 245.00 seconds the semaphore: xxxxxx Mutex DICT_SYS created dict0dict.cc:1173, lock var 1”则对 Mutex DICT_SYS 进行埋点。

3. 源码分析，在内核层面对 SQL 的执行过程进行逐步的 Debug，根据经验分析可能的瓶颈点来埋点；下面举两个例子分别就纯源码的角度来分析，SQL 在读、写过程中哪些地方可能会造成性能抖动并做埋点。

例子 1，在 SQL 读操作的执行过程中，需要从 buffer pool 中请求内存资源，如果一直满足不了内存资源的请求则一直会循环尝试获取，如果在 2000ms 内还是没有从 Buffer pool 中请求到需要的内存，会打印日志“Difficult to find free blocks in the buffer pool……”，来表明内存很紧张；这种长时间等待内存资源的获取会生产了性能抖动，这个时候就需要在对应的内核代码处进行埋点获取 BP 内存资源等待时间。

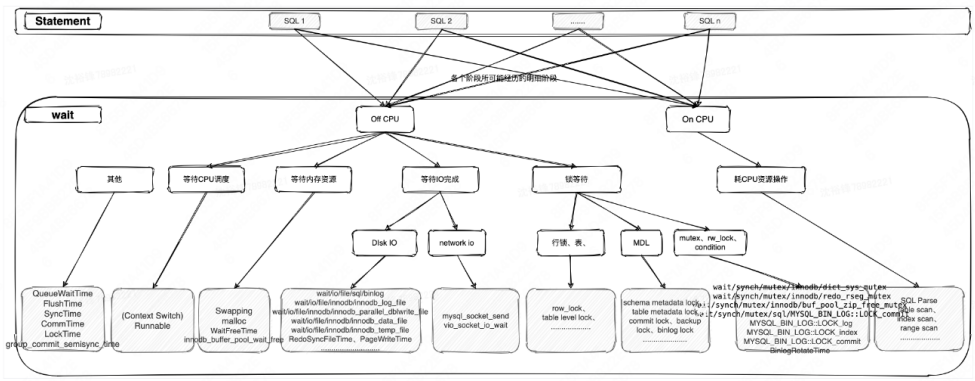


例子 2，在 SQL 写操作的执行过程中是需要写 redo log 的，如果 redo log 空间不够，则需要刷盘 redo log 跟 BP 中的脏页，而刷脏页可能是个很耗时的操作，并会引起包括活跃会话突增、慢查询等性能抖动问题。这个场景跟上面的类似，也需要对源码进行分析，并且在关键的位置进行埋点来统计刷脏的耗时。



1.1.4 Wait 指标层次图

通过上面的三种方式我们不断地迭代累计的关键耗时指标，整理成如下的 OffCPU 跟 OnCPU 两大类的分层架构图。截止目前为止，我们内核团队在内核层面埋点了 100 多个关键指标来辅助诊断，指标分为 Statement 跟 Wait 两大层级。



Statement

SQL 语句级别的指标，如 QUERY_TIME、Row_time 行锁时间、ROWS_EXAMINED、ROWS_SENT、Bytes_sent、NO_INDEX_USED、Full_SCAN 等，但是

这个层面的数据不足于判断出 SQL 性能问题的根因，比如 QUERY_TIME 很长，但是 ROWS_EXAMINED、ROWS_SENT 都很小，就需要进入下一个 Wait 级别的指标进行进一步的分析。

Wait

Wait 层面主要是 MySQL 内核层面的指标，比如 Latch（Mutex、rw_lock、sx_lock、cond）的指标，这里需要说明的是上面的分层图跟 MySQL 自带的 performance_schema 中的 wait type 分层图看上去很像，那为什么不直接使用而需要自研呢？这里主要从如下几个原因且内核团队都很好的解决了：

1. performance_schema 相关的统计信息是自实例启动后总体的 wait 时间的聚合数据，而更需要的是任何时间段、单 SQL 例子的明细数据，因为诊断的往往是某个时间点或者时间段的异常 SQL 问题。
2. performance_schema 中其有不少 Bug，除此之外，它只支持 OffCPU 类型的指标，OnCPU 耗时的统计不支持，甚至有些非常重要 OffCPU 的指标，比如 row lock 的 wait 统计信息都不支持。
3. 开启 performance_schema 相关的 wait 后，对总体的性能影响比较大，而基于内核埋点的实现方式很轻量级，对数据库整体性能影响很小。

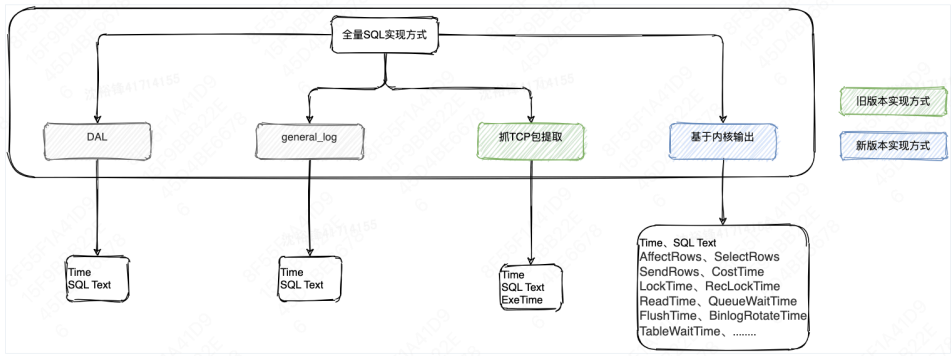
2. 全量 SQL

全量 SQL，指的是把应用程序或者用户访问 DB 的所有 SQL 集合，需要捕获到这些 SQL 并且发至后端进行分析。在内核可观测性建设之前，原来采用的方式是根据 MySQL 的协议，来解析 TCP 报文来提取出 SQL。此种实现方式的缺陷也很明显，就是除了 SQL 文本外，其他的能获取到的关键 SQL 指标信息非常少，这样做数据库的故障诊断跟 SQL 性能分析，会因为缺少关键指标而带来挑战。

针对此挑战，我们的解决方案是：改造为直接从 MySQL 内核吐数据，来作为全量 SQL 的数据来源（前面的内核可观测性分析介绍可知，单条 SQL 维度上能从内核层面输出 100 多个关键指标，包含故障诊断跟 SQL 性能分析需要的重要指标）。

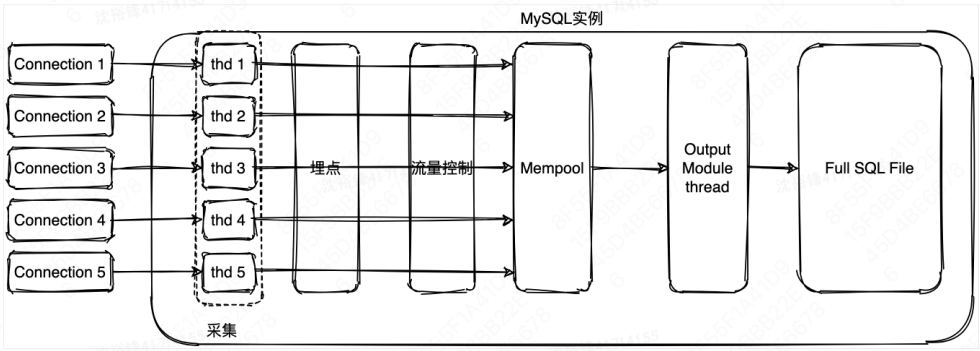
2.1 实现方式

全量 SQL 其实有好几个实现方式，初版使用了抓 TCP 包的方式，现在逐步从 TCP 抓包过渡到了基于内核输出 SQL 文本跟关键指标的方案。



2.2 内核实现方式

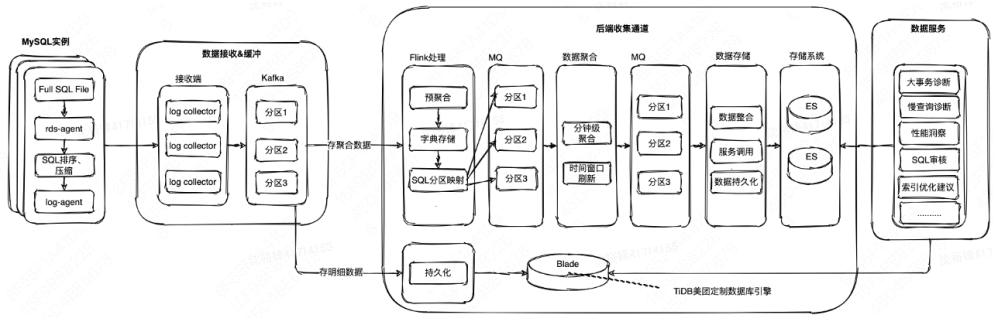
从内核层面来说，MySQL 对于用户的链接将创建独立的 thd 结构体，所以采集的全量 SQL 对应的 SQL 文本跟关键指标数据，可以存于 thd 结构体中。然后用户线程将 thd 保存的指标数据，复制到一个自定义内存池中（无锁队列）；输出模块有一个线程从内存池中依次获取保存的信息后，写入全量 SQL 文件。具体流程如下图所示：



2.3 全量 SQL 总体架构

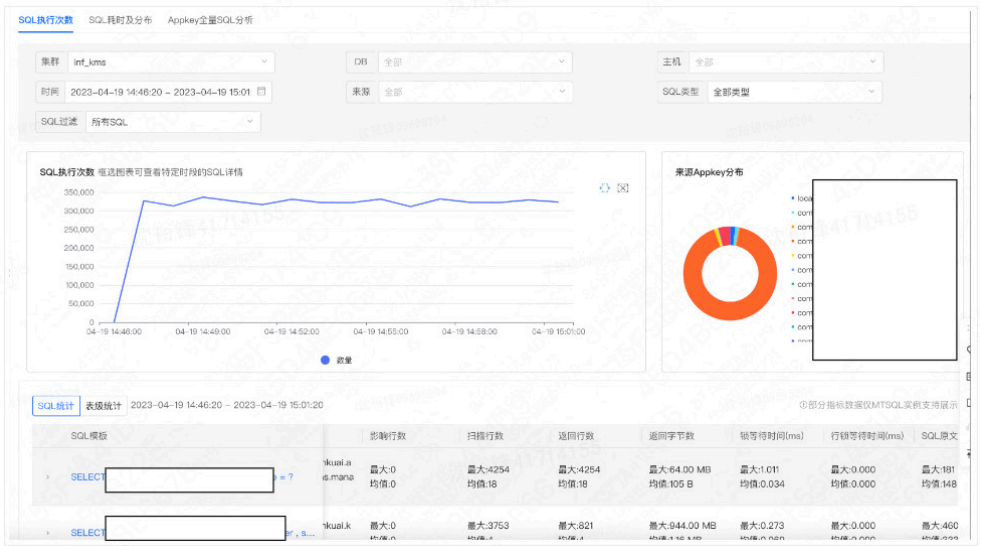
如何把海量信息（日均 PB 级别）上报到后端系统，来赋能故障诊断、SQL 审核、索引优化建议等场景是一个较大的挑战。我们首先通过数据采集器（rds-agent）读取 Full SQL File 文件内的内容，再往后端传输。

但是由于数据量特别大，为了节省存储资源，分析后选用 Snappy 压缩算法进行压缩，默认情况下压缩效果只有 1 ~ 2 倍，为此分析了 Snappy 的源码，发现如果让相似的 SQL 文本聚合在一起，那么压缩效果会有很大的提高，所以按照 SQL 文本的前 N 个字符（N 取 50，可以根据 SQL 文本的实际情况做调整）进行了排序后再压缩，发现压缩比提高到了 7 ~ 8 倍。后端的 SQL 存储分成了两种形式，一种是存明细 SQL，另一种是存聚合后的 SQL 模版信息。



产品展示

采集到的全量 SQL 既可以模版化展示，点击模版后也可以看到明细的 SQL。



3. 异常处理

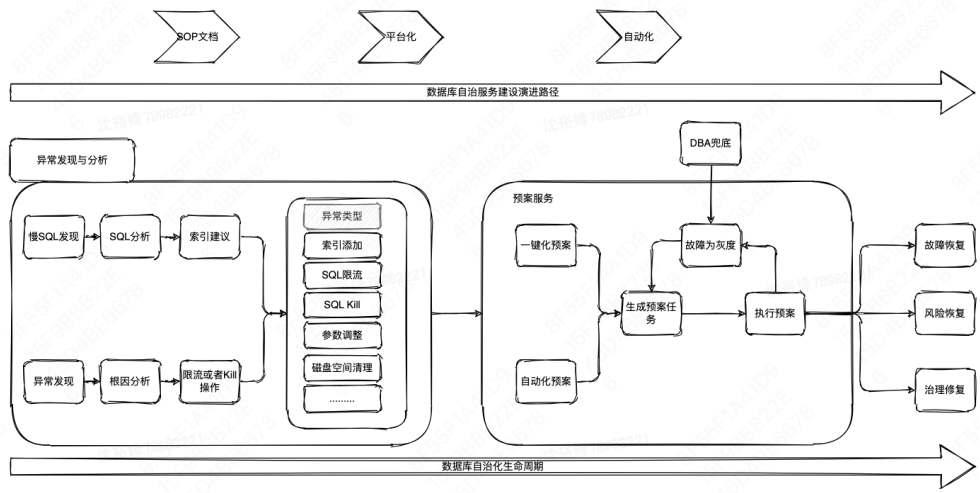
根因分析后，就需要根据具体根因来进行相应的处理，但是如何安全、可靠的进行处理其实是一个很大的挑战。具体的策略是根据其操作是“无损”还是“有损”来采取不同的处理策略。

如果是对业务“无损”的操作，比如磁盘空间清理、参数值调整、缺失索引添加等，目标是让操作尽量的自动化。如果是对业务是“有损”的操作，比如需要 Kill 或者限流，把相关操作进行包装，发送相关的聊天群让 DBA 或者用户确认后再进行相关的操作。

对于 MySQL Hang 的情况，高可用团队会进行周期性探测，如果发现主库 MySQL Hang 了，则会自动进行主从切换，如果发现从库 MySQL Hang 了则会进行相应的 MySQL 实例下线替换动作。

3.1 系统架构

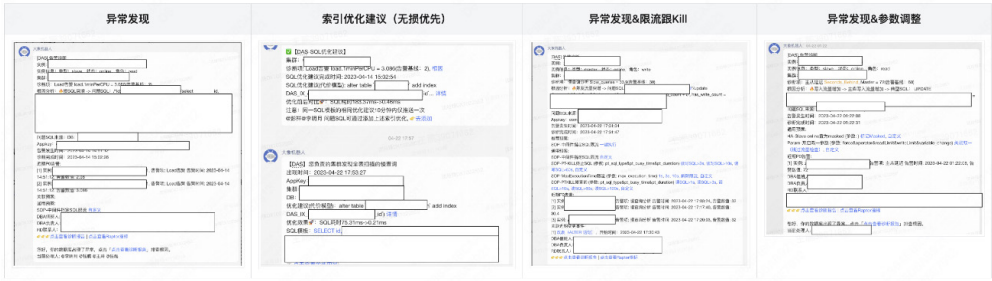
整个异常处理的系统架构图如下，由“异常发现与分析”系统以及 DBA 团队开发的“预案服务”系统组成，前者提供精确的故障根因，后者根据具体的根因来安全、可靠的执行对应的预案，使故障快速恢复。



3.2 产品展示

下图是两个异常发现、分析、给出建议以及处理的例子：1）异常发现、确认根因后，发现缺索引，系统会自动调用索引优化建议系统给出建议，用户可一键添加索引；2）对于

慢查询或者主从延迟问题，分析根因后提供用户进行限流、Kill、参数调优等操作的建议，避免异常被进一步放大。



4. 索引优化建议与治理

众所周知，很大一部分的数据库的异常都是跟 SQL 性能有关。很显然，日常的 SQL 性能问题的治理就很有必要，一个低成本但效果显著的 SQL 性能优化方案是提供索引优化建议，并且可以将这种索引优化建议的能力应用于 SQL 生命周期的三个阶段，这三个阶段下文会进一步阐述。

1. 应用程序“发布前”的性能问题 SQL 审核（事前）；
2. SQL 执行过程中的实时性能问题 SQL 发现（事中）；
3. SQL“执行完毕”后批量的 SQL 治理（事后）。

4.1 索引优化建议

建立适合的索引对 SQL 的性能提升效果很明显，如何添加适合的索引是一个有挑战性的任务。索引的添加不能光凭经验，因为索引是否被用到不但跟表的数据的分布有关，还跟 MySQL 查询优化器的 Cost 逻辑有关系，所以“人肉”进行索引建议的评估并不是一个好的方案。

解决方式是尽量要基于 MySQL 的 Cost 模型，利用 MySQL 自身提供的查询优化器的能力，来给出最佳的索引；索引优化建议的建设一般分为三个阶段：1）单 SQL 维度的优化建议；2）基于 workload 的整体优化建议；3）索引自维护。这几个阶段的建设需要按顺序来进行，并且不可跳跃。下面我们将分别进行阐述。

4.2 单 SQL 索引优化建议

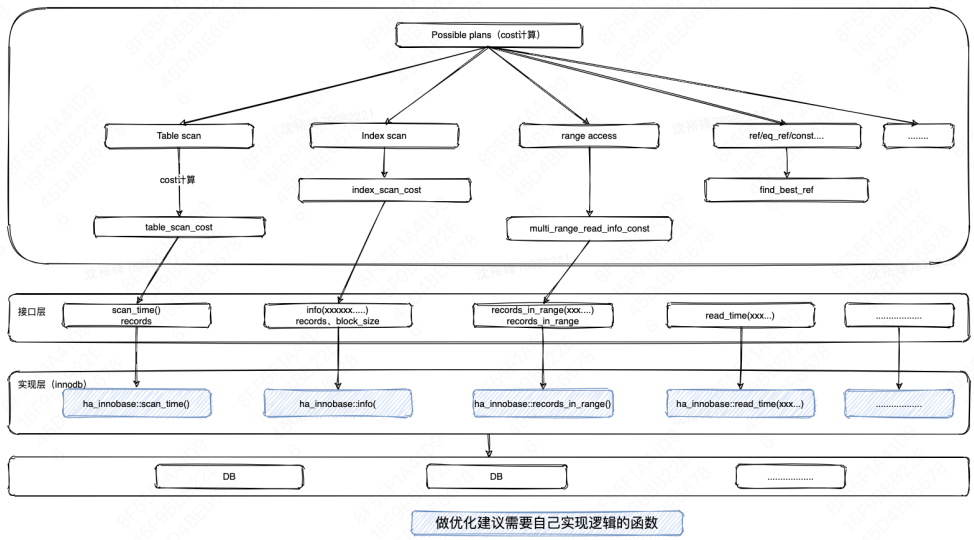
4.2.1 实现思路

单 SQL 索引优化建议，就是指输入一条 SQL 语句后，优化建议系统给出一个索引优化建议的过程。想要利用查询优化器本身的能力来实现索引建议的目标，就必须先要了解 MySQL 的查询优化器是如何工作的。

这里举个简单的单表 SQL 查询的例子，帮助我们来初步了解其工作原理。我们先了解查询优化器是如何从众多的候选执行计划中选择最终的执行计划的；在有了这个基础之后，再来讲是如何利用查询优化器自身的能力来做索引优化建议。

比如有 SQL 语句 "select * from test_db.table1 where c2=3 and c3=4 and c4<'3'"，分别在字段 id、c2、c3 以及 c4 上面建立了主健索引 Primary 以及三个二级索引 idx_c2、idx_c3 及 dx_c4。假设最终查询优化器从多个候选执行计划中选择了“执行计划 2”（通过索引 idx_c2 做 index seek 操作来执行 SQL），问题是此 SQL 的候选执行计划有多个，为什么最后选中了通过索引 idx_c2 而不是其他的索引做 index seek 操作的执行计划呢？总的来说，选择 idx_c2 为索引的执行计划，是因为相比于选择其他索引生成的执行计划的 Cost 更小。

具体来说，针对 SQL 的两类表访问方式：1）基于全表扫描的访问方式；2）非全表扫描的方式（比如 Index_scan、ref 类型或者 range_access 等），下图是这些访问方式 Cost 计算过程中涉及到的查询优化器跟存储引擎的关键函数，目标是假设某个索引存在的情况下，修改这些函数来模拟 Cost 的计算过程，来发现哪个索引存在的情况下对应的 Cost 值最小。



流程中的各种数据访问方式，对其 Cost 计算方式逐个分析。

1）Table scan 访问方式

从全表扫描 cost 计算公式可以知道 Table scan 的 Cost 分为 IO 的 Cost 跟 CPU Cost 两个部分之和，大致的公式为：IO-cost:#pages in table * IO_BLOCK_READ_COST + CPU cost:#records * ROW_EVALUATE_COST，其中 IO-cost 是通过 table_scan_cost 来计算的。

这里有两个关键的变量 records 跟 pages in table，分别表示这个表有多少行跟多少 page，这两个变量的值是怎么来的呢？如果使用的是 innodb，那么前者来自于 ha_innobase::info_low（ha_innobase::info），后者来自于 ha_innobase::scan_time()（而这两个函数都是 SQL 层 handler.h 定义的虚函数，不同的存储引擎分别实现之），知道了这两个变量的值，就知道具体的 Cost 值了。

2）Index scan 访问方式

覆盖索引的扫描，从 index_scan_cost 得知 Index scan 的 Cost 为：O-cost:#(records/keys_per_block) * IO_BLOCK_READ_COST+ CPU cost:#records * ROW_EVALUATE_COST，由于 IO_BLOCK_READ_COST 跟 ROW_EVALUATE_COST 都是常量，所以需要关注是 keys_per_block（keys_per_block 的计算跟 block_size 等有关）跟 records 这两个变量，也就是说如果知道了这两个变量的值，

就知道了具体的 Cost 值了。

3）range access 访问方式

从 multi_range_read_info_const 得知 range access 的 Cost 为：IO-cost:#records_in_range * IO_BLOCK_READ_COST + CPU cost:2*#records_in_range* ROW_EVALUATE_COST，由于 IO_BLOCK_READ_COST 跟 ROW_EVALUATE_COST 都是常量，所以只需要知道 records_in_range 这个变量的值，就能计算其 Cost 了。

4）ref 访问方式

从 find_best_ref 得知 ref 访问方式的 Cost 大致为：IO-cost:#prefix_rowcount * IO_BLOCK_READ_COST + CPU cost:#prefix_rowcount * cur_fanout * ROW_EVALUATE_COST，需要实现 cur_fanout 这个变量，而这个变量跟 info 中的 rec_per_key（rec_per_key 的意思是，比如对于 select * from test_db.table1 where c2=3，table1 中有多少行数据满足 c2=3）有关系，所以实现 rec_per_key 算法后就可以知道其 Cost 值了。

4.2.2 What_If 索引建议策略

如何把上面的 Cost 计算方式的理论分析应用到索引优化建议中？假设表 table1 除了主键外没有任何索引，索引优化建议工具最后怎么能给用户推荐出索引 idx_2 呢？关于这个问题可以用到一种称之为 What_If 的策略（在微软的 AutoAdmin "what-if" index analysis utility 中提出），思路就是假设索引 idx_2 存在并且能计算出在索引 idx_2 存在的情况下，此执行计划的 Cost 值，如果发现其 Cost 只要比当前存在的其他执行计划的 Cost 值更小，就推荐这个索引。

但是 MySQL 计算 Cost 的时候，要求索引是真实存在的，为了给出索引优化建议在生产环境实际去创建各种索引是不可能的。解决的策略就是在非生产环境的数据库实例上创建一个空表，并且添加所有可能被用到的索引（只是一个元数据的 create index 操作而不去真正的向表内添加数据），最后让优化器去自己选择最佳的索引来生成执行计划，这就是索引优化建议总体思路。

那么问题来了，在非生产环境下创建的索引只是一个元数据，不包含任何数据，如果就这样让查询优化器去计算 Cost 肯定是行不通。主要原因是缺乏上面的分析的各个访问方式的关键变量值，所以最关键的如何获取到那些影响 cost 计算的关键的变量信息。

从上面的“实现思路”分析可知，在计算 Table scan、Index scan、range access 或者 ref 的 cost 的时候，主要是从 scan_time()、records_in_range()、info() 这些在 innodb 存储引擎里实现的方法获取相关统计信息，来计算 Cost 大小。而索引没有真实的数据，所以必须要修改存储引擎里 scan_time()、records_in_range()、info() 这些跟计算 Cost 相关的方法的实现代码，让其在索引数据不存在的情况下，也能比较真实地获取到跟计算 Cost 值相关的统计数据。

比如，上面提到的 records、keys_per_block、pages in table、records_in_range、cur_fanout 等这些变量，就需要自己去计算；比如 records 这个变量，发现其是在 Innodb 中 info() 的实现，通过 n_rows（records）赋值给 records，所以只需要在 info() 中从生产环境的表里获取到真实的行数，赋值给 records 就可以了，其实就是一个元数据的查询而已。

再看一下 ref 类型访问方式的 Cost 计算过程中用到的 cur_fanout 变量，而这个值是来自 info() 中 innodb_rec_per_key 的值，它表示有多少个值满足等值条件，只需要在 info() 中实现 innodb_rec_per_key 的计算就可以了。

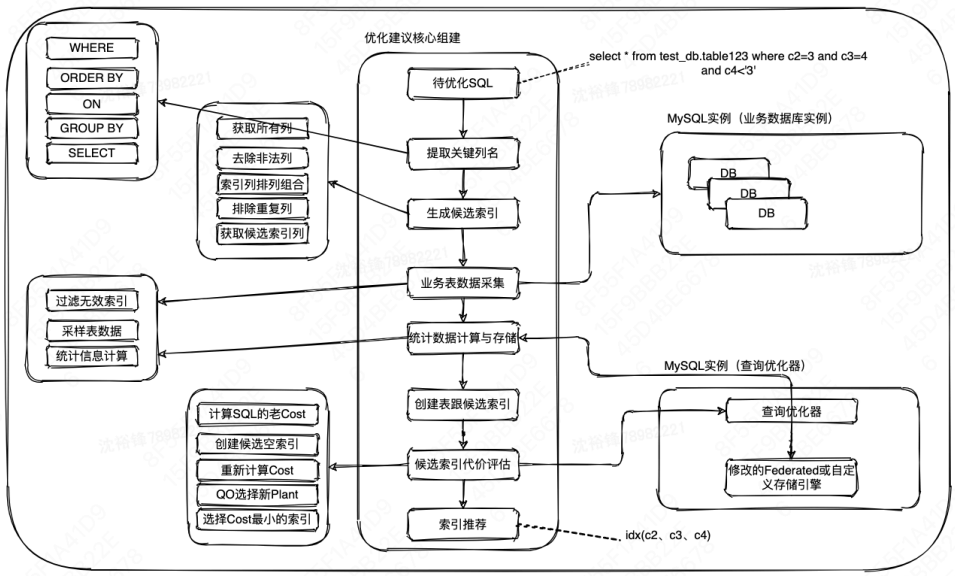
那么 innodb_rec_per_key 又怎么计算呢？我们可以对表进行了采样（采样的方案可以参考这篇《Random Sampling for Histogram Construction: How Much is Enough?》论文），采样后的数据对某列统计有多少个不同的值，之后使用采集到的总行数 / 不同值的个数来获取到 innodb_rec_per_key 值。

这里还有一个非常重要的问题，上面的方案是需要修改对应存储引擎的 scan_time()、records_in_range()、info() 这些关键的方法，但是 innodb 存储引擎的代码太复杂了，修改的难度比较大，一个比较好的方案是去修改 federated 存储引擎里的 scan_time()、records_in_range()、info() 这些方法，去获取上面不同访问方式中需要用到的关键变量，因为其实现代码比较简单。

4.2.3 索引建议整体流程图

把之前的步骤整理后的索引优化建议的流程图如下：

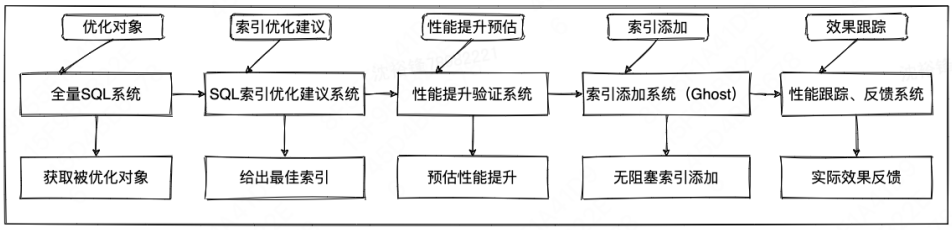
1. 从待优化的 SQL 的关键位置，比如 Where、On、Ordr By 等位置提取关键字段信息，利用提取到的字段信息生成候选索引；
2. 通过对涉及到的表的数据进行采样来获取查询优化器计算 Cost 时需要用到的统计信息，之后创建不包含任何数据的索引；
3. 修改存储引擎的代码重新实现的 scan_time()、records_in_range()、info() 这些函数，使得查询优化器能利用数据采样后计算出来的统计信息来比较真实的获取 Cost 值；
4. 选择一个 Cost 代价最小的执行计划，而此执行计划背后的索引就是优化建议工具推荐的索引，整体思路就是这样。



4.2.4 验证与跟踪

通过全量 SQL 等系统获取待优化的 SQL 语句后，输入索引优化建议系统，在给真实的生产环境添加这些索引之前，需要在非生产环境进行验证，看一下添加了相关的索引后性能是否真的有提升。如果有提升，那么把索引在生产环境添加后（比如通过开源的改表工具 Ghost 等），需要通过跟踪被影响 SQL 的性能，来查看其执行时间性能是否

真的有所提高。如果有性能有回退，需要及时进行告警。



4.2.5 产品展示

下面的三个图分别展示了提供给用户的索引建议，以及索引添加前后的执行时间效果数据对比图。

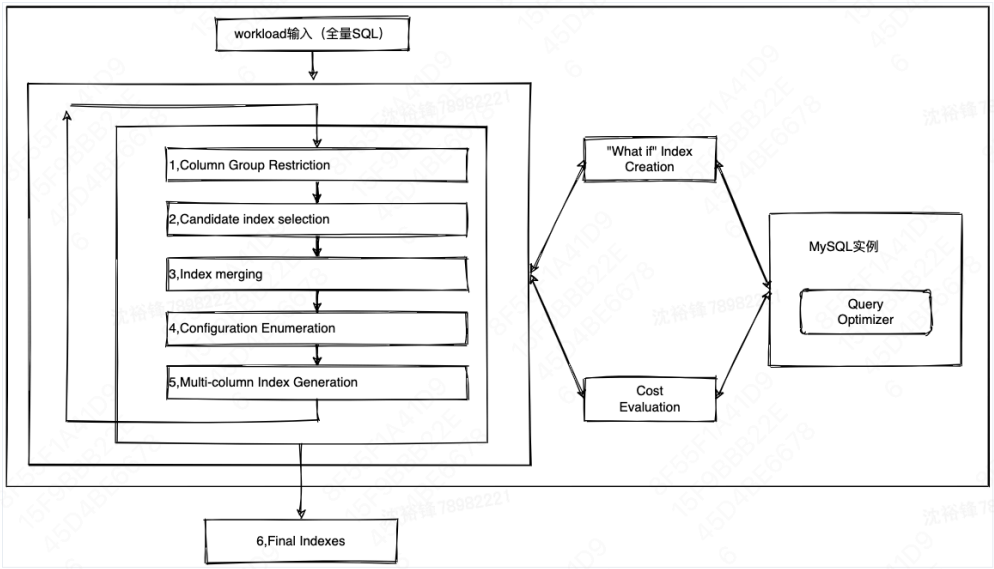


4.3 基于 Workload 的索引优化建议

具备了单 SQL 的索引优化建议能力后，那么在有限的存储空间的限制下（比如某个实例最多只能添加 100G 空间的索引），如果想给某个 DB 或者实例添加索引，添加哪些索引才能使整个数据库的性能提升达到最大化呢？思路主要是参考了微软在基于 Workload 索引优化建议方面的一序列相关工作。

4.3.1 总体架构

基于 Workload 的索引优化总体架构如下，总共有 6 大步骤组成，每个步骤对应着一个核心组件。



1）COLUMN GROUP RESTRICTION 组件

如果一个 workload（典型业务周期内包含的所有 SQL 总和称之为 workload）包含大量的 SQL 模版，那么潜在的索引组合的数量是极其庞大的，从中挑选出让系统性能提升最大的索引组合是一个非常困难并且耗时的过程。所以，我们需要一种称为列组合限制 (COLUMN GROUP RESTRICTION) 的技术，它的目的是消除大量可能组成索引的列组，因为这些列组最多只能对最终推荐的质量产生很小的边际影响（就是说就算推荐了索引，最后对数据库整体的性能的提升很有限，比如某个 SQL 只执行了一次，那么给之对应的表加索引就没意义了），所以本步骤的输出是 workload 的一组“有趣”的、对整体性能提升可能有最大提升的列组。

那么，怎么样选出这个“有趣”列组呢？我们需要定义一个叫 CG-Cost (g) 的函数，其值为引用列组合 g 的 workload 中所有查询 SQL 的 Cost 的某个分数值，查询 SQL 的 Cost 可以通过优化器估计的成本（MySQL 中 explain 给出的 Cost）来获取。如果 CG-Cost(g) ⩾ f ，列组 g 需要保留下来（其中的 0 ⩽ f ⩽ 1 是预先确定的阈值），否则可以丢失这个列组 g。看一个具体的例子：考虑下面的表 T (A,B,C,D) 的 workload 由“Q1，… Q10”等 10 个 SQL 组成。如果查询引用该列，则上述矩阵中的单元格包含 1，否则包含 0。为了简单起见，假设所有查询的成本均为 1 个单位。

假设指定的阈值 f = 0.2，那么工作负载的有趣列组是 {A}、{B}、{C}、{A,B}、{A,C}、

{B,C} 和 {A,B,C}，以及各组的 CG-Cost(g) 为 1.0、0.3、0.9、0.3、0.9、0.2、0.2，而 D 相关的列因为 CG-Cost (g) 值小于 0.2 而不被考虑，这样就可以在庞大的列组合的情况下排除掉大部分列组合，加快整个索引选择的迭代优化的过程。

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
A	1	1	1	1	1	1	1	1	1	1
B	1	1	1	0	0	0	0	0	0	0
C	0	1	1	1	1	1	1	1	1	1
D	0	0	1	0	0	0	0	0	0	0

2）Candidate index selection 组件

单列索引（假设索引）的建立，如果考虑 workload 的每一个可能的索引集，那么就会有太多的 what_if 索引需要考虑，这样整体的优化过程会非常缓慢。

一个比较好的思路是为每个 Query 独立确定最佳的索引集，并将属于这些最佳的索引集中的一个或多个索引视为候选索引集。该算法背后的直觉是，如果对于 workload 中的单个查询，该索引不是最佳的索引集的一部分，也不可能是整个 workload 的最佳的索引集的一部分。

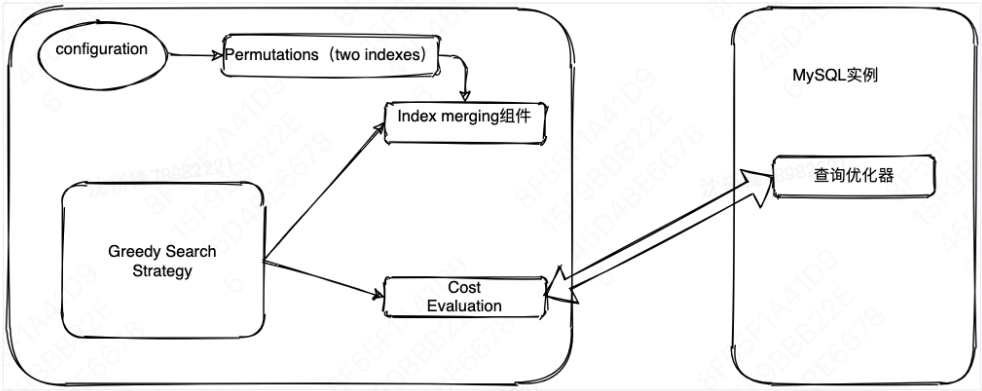
这个算法的挑战在于如何为 workload 中的每个查询 Q 确定最佳的索引集呢？其实针对单个查询来确定最佳的索引集这个问题与根据整体的基于 workload 来选择最佳的索引集这个问题，两者没有本质的不同。所以可以设计一个特殊的 workload，而这个 workload 只包含一个查询，之后可以通过上面已经实现的索引优化建议工具本身，来为单个 Query 获取最佳的索引集。

单查询获取最佳的索引集，如何为单个 SQL（SQL 模版）获取最佳的索引集呢？一个方法是单个 SQL 模版进行解析，在 SQL 关键的位置比如 where、on、order by、group by、select 等的取出关键的列先创建单个列作为候选索引列，再结合下面的从“Multi-column Index Generation 组件”那节创建出来的多列候选索引，让查询优化器自己去选择最好的索引。同时这个步骤依靠优化器选出来的索引，作为下一轮算法（索引合并）的输入。

3）Index merging 组件

上面的“CANDIDATE SELECTION”组建选出来的索引是专门针对单 SQL 优化的索引集，但是对于整体 workload 来说，这个方法会导致较大的存储和索引维护成本。比如，针对单 SQL 选出的索引都是最佳的，但是这些索引加起来后的存储大小已经超出了允许的范围，需要进行一些索引的合并。

如果两个索引的列存在部分重叠，则可能可以合并，比如包括关系的索引 [a,b,c] 跟 [a,b] 以及有部分重叠的索引 [a,b,c] 跟 [c,d}] 的合并，当然 [a,b] 跟 [c,d}] 是不能合并的。索引合并的主要思想就是采用现有的一组索引生成一组具有更低存储和维护开销的新索引，同时索引被合并的情况下，SQL 查询的性能能保持不变或者影响比较小，索引合并的架构如下：



4）Configuration Enumeration 组件

如果基于 workload 的索引集有 n 个候选索引，但是因为存储大小限制，要求 workload 工具最多只能选择 k 个索引（一般是受到了存储大小跟 Cost 大小的限制），那么应该如何选择这个 k 个索引呢？一般来说，原始的暴力枚举算法将枚举大小为 k 或更小的候选索引的所有子集，并选择总成本最低的一个索引集，这个枚举算法确实保证了最优解。

然而，这在工程上不切实际的。因为对于 n=40 和 k=10 的情况，枚举的索引集的数量实在太庞大，无法进行详尽的枚举，所以需要一个性能更好的算法的“贪心算法”，其思想是假设有 n 个 SQL（一般是实施的时候，是 n 个 SQL 模版来替代 n 个 SQL 实

例），整个 workload W 表示为 {Q1, .., Qn}。

方案是利用暴力跟贪心算法的组合，运行 Greedy(m, k) 算法枚举索引集合（m 为通过暴力算法选择的 m 个最佳索引，k 为选择的索引的总大小），选择一组索引，直到选出所需数量的索引或总 Cost 无法进一步降低，该算法选择大小为 m（其中 m <= k）的最佳索引集作为“种子”，然后以贪心算法扩展这个种子，直到选择了所有 k 个索引，探索算法就是从所有的候选索引中选择一个索引，使得加了这个索引后，整体 SQL 的性能提升效果最大化，虽然从全局角度看未必是提升最大化的。

5）Multi-column Index Generation 组件

多列索引的选择有一个比较大的挑战，就是候选索引实在太多，比如对于表中给定的一组 k 列，最多可能有 k! 多列索引，考虑所有排列组合将会显著增加 Configuration（索引）空间，所以需要减少列组合的搜索空间。采用迭代方法来考虑多列索引的宽度（列的个数）。在第一次迭代中，只考虑单列索引，基于在第一次迭代中选择的单列索引，在往后的迭代中选择一组可接受的双列索引。这组“两列索引”与单列索引一起成为第二次迭代的输入。

那么，具体是如何在单列的基础上选择其他的列组成多列索引呢？这里有多种算法，先使用符号 M (a, b) 来表示列 a 和 b 上的双列索引，其中 a 是双列索引的前导列。多列索引选择一般有 MC_LEAD、MC_ALL、MC_BASIC 三种算法，这几种算法在实现的时候一般会采用 MC_LEAD 算法 ，因为被证明效果最好。

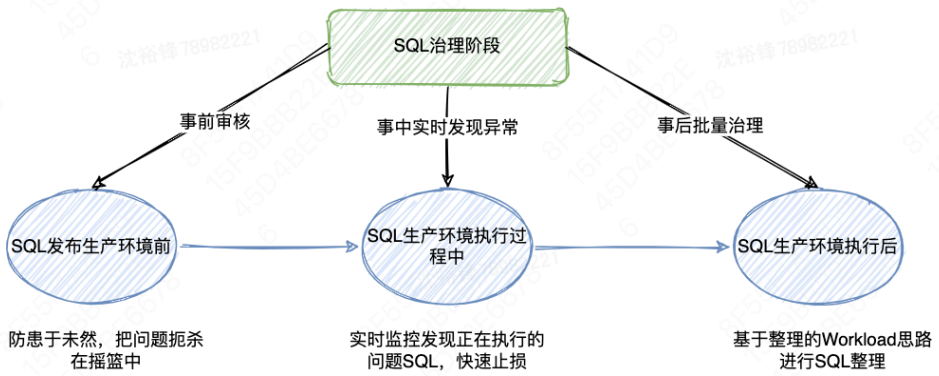
- **MC_LEAD**：从 CANDIDATE SELECTION 被优化器选择的索引，再加上某个“indexable column”列，这个“indexable column”列对应的索引不一定在第一轮迭代中被选择；
- **MC_ALL**：从 CANDIDATE SELECTION 被优化器选择的索引，再加上某个“indexable column”列，这个“indexable column”列对应的索引一定在第一轮迭代中被选择，这种条件比较苛刻，业界相关的测试表明效果明确不如 MC_LEAD 跟 MC_BASIC，虽然多列索引的搜索空间明显减少，提升了总体性能；
- **MC_BASIC**：则是把单列跟多列索引混在一个迭代周期里建立，而不是先根据单列建议一个索引，然后再通过迭代的方式在单列的基础上逐渐增加索引的宽度。

6）Final Indexes 组件

经过几轮迭代，当同时满足 Cost 跟 Storage Size 的要求后退出迭代优化过程，输出一组最优的索引优化建议集合。

4.4 SQL 治理

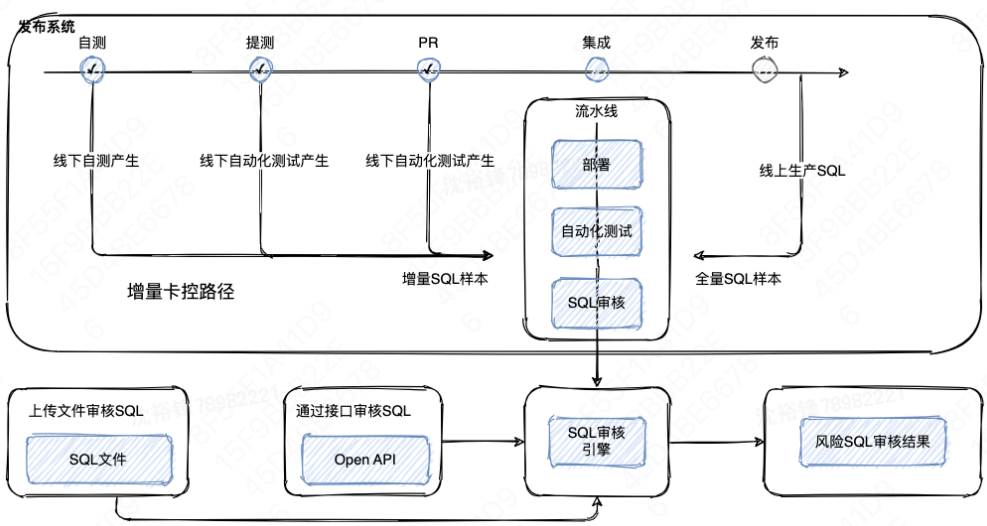
具备了 SQL 索引优化建议的能力后，就可以对有性能问题的 SQL 进行三个阶段的治理，1）SQL 发布生产环境前，起到防患于未然、把问题扼杀在摇篮中的作用；2）SQL 生产环境执行过程中，起到实时监控发现正在执行的问题 SQL，快速止损，比如误删索引的作用；3）SQL 生产环境执行过后，对于执行过的 SQL，基于整体的 Workload 优化策略，通过离线的方式进行批量的风险 SQL 治理。如下图所示：



4.4.1 风险 SQL 审核（事前）

此阶段是程序发布前对潜在的风险 SQL 进行卡点，在公司的 CI/CD 平台集成流水线里增设 SQL 审核卡控点，尽量防止风险 SQL 被带到生产环境引发故障，起防患于未然的作用。

具体来说，指定了一序列的规则，发现风险 SQL 后可以暂停发布，并且给出相应的索引优化建议来优化 SQL。



4.4.2 产品展示

根据审核规则，展示审核结果与建议，包括风险提示跟 SQL 索引优化建议等内容，审核的规则包括是否全表扫描、扫描行数过多等内容。

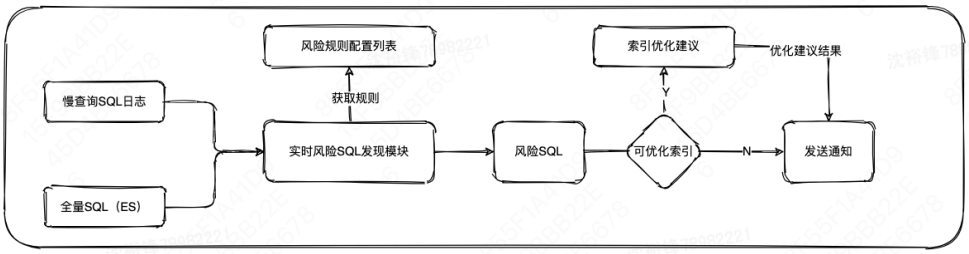


4.4.3 性能问题 SQL 实时发现（事中）

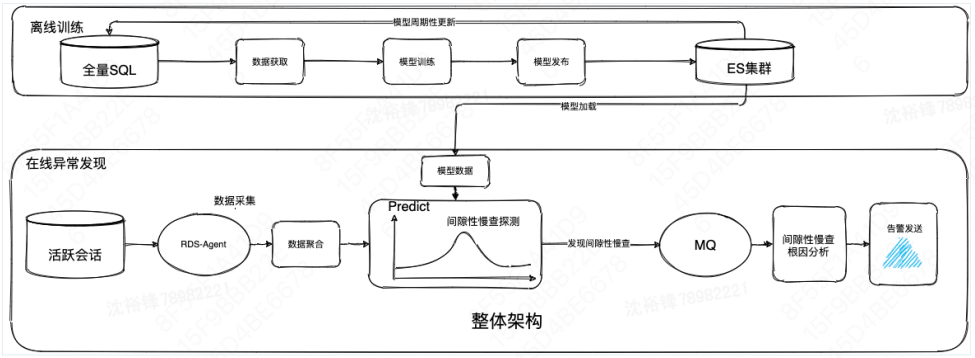
此阶段是风险 SQL 的实时发现功能，主要针对在 SQL 运行过程中因为表结构的更改（如索引误删除）、事前审计阶段遗漏掉的性能问题 SQL 等场景，实时地发现有性能问

题的 SQL，并且给出优化建议。对于实时的性能问题 SQL 发现使用了两种发现策略，一种是基于规则的发现策略，另一种通过数据建模的方式来发现策略。

基于规则的风险发现策略，其中的规则包括单位时间内的慢查询数量、SQL 执行时间、扫描行数等内容。

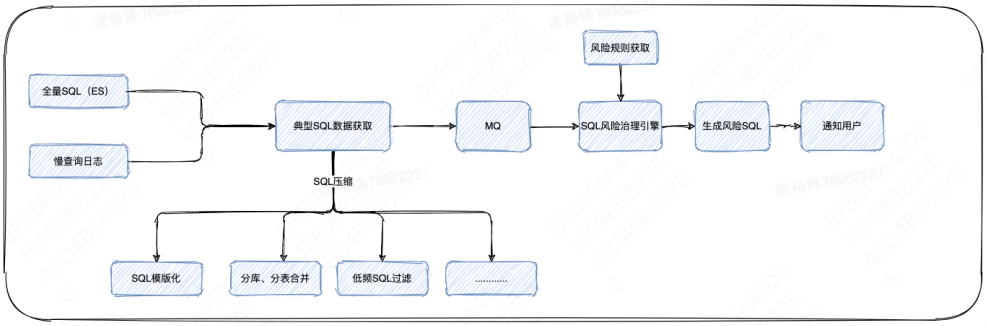


基于数据建模的风险发现策略，是通过全量 SQL 获取 SQL 的历史执行时长来建模，再通过 Process List 等数据源，把当前正在执行 SQL 的执行时间输入模型，从执行时间的角度来判断是否有异常。



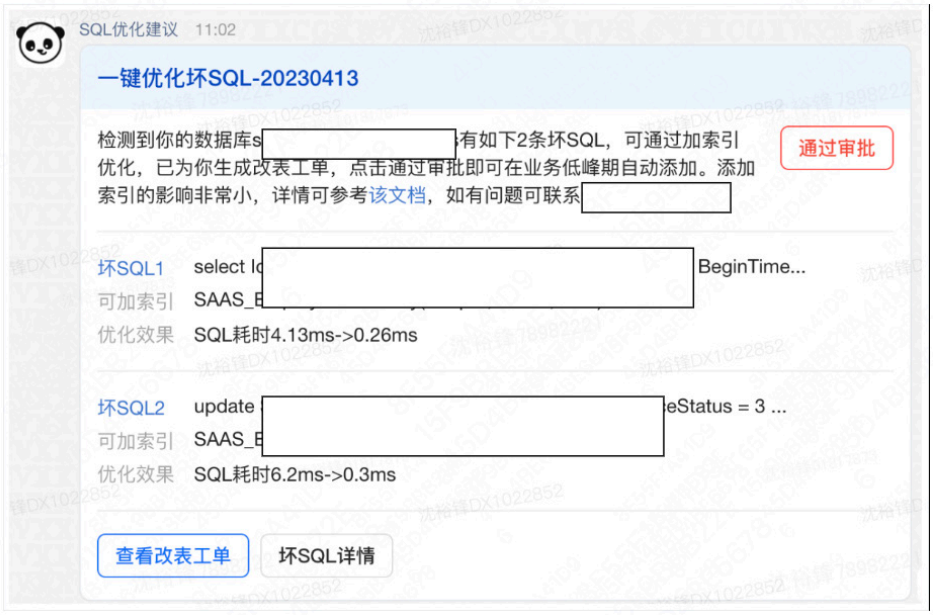
4.4.4 批量 SQL 治理（事后）

此阶段是通过对 SQL 执行的历史记录进行批量分析，从全局的角度、利用 workload 索引优化建议的思想提供最佳 SQL 索引优化建议，批量推送优化建议给用户进行风险 SQL 的治理。



4.4.5 产品展示

对于批量的 SQL 优化建议，这里提供了一键审批索引添加的功能，可以更快的让被建议的索引添加到生产环境上。



5. 本文作者

裕锋，来自美团基础研发平台 – 基础技术部，负责美团数据库自治平台的相关工作。

6. 参考

- https://github.com/shenyufengdb/sql
- https://github.com/percona/percona-server/blob/release-5.7.41-44
- An Efficient, Cost-Driven Index Selection Tool for Microsoft SQL Server

- plan-stitch-harnessing-the-best-of-many-plans-2
- Random Sampling for Histogram Construction: How Much is Enough?
- AutoAdmin “what-if” index analysis utility
- What is a Self-Driving Database Management System?
- https://www.microsoft.com/en-us/research/publication/self-tuning-database-systems-a-decade-of-progress/
- Automatic Database Management System Tuning Through Large-scale Machine Learning
- Query-based Workload Forecasting for Self-Driving Database Management Systems
- The TSA Method
- https://blog.langchain.dev/langchain-chat/
- https://github.com/hwchase17/langchain
- REAC T: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS
- Evaluating the Text-to-SQL Capabilities of Large Language Models
- SQL-PALM: IMPROVED LARGE LANGUAGE MODEL ADAPTATION FOR TEXT-TO-SQL

AIOps 在美团的探索与实践——事件管理篇

作者：政东 迎港 俊峰

0. 写在最前

文中所提及的事件并不仅限于故障，还包括运维工作中的告警、异常等。

“An incident is an unplanned interruption to an IT Service or a reduction in the Quality of an IT Service.” Source: Incident Management –ITIL

1. 背景

在《AIOps 在美团的探索与实践——故障发现篇》一文中，我们探讨了 AIOps 在异常检测的实践。Horae（美团 AIOps 平台）在单时序异常检测方面已有较多积累，智能告警功能作为底层能力支撑了监控系统和异常检测场景。服务运维团队在此基础上开展 AIOps 在事件管理领域的相关工作，本文主要分享过去两年的探索与实践，希望能对大家有所帮助或启发。

事件管理的复杂性体现在两个方面：

1. 数据繁多：

- **数据多样化**：运维工作需要各种类型的数据来识别、诊断、处理问题，包括告警、链路、指标、日志、变更（含发版）等。
- **数据实时性强、关系复杂**：运维数据通常需要实时采集和处理。这些数据之间的关系错综复杂，如链路数据与告警数据、指标数据与日志数据等可能存在密切的关联，需要精细的统一处理。
- **领域知识强**：运维领域涉及的知识广泛，包括网络、硬件、系统、数据库、应用等多个层面，业务运维更需要不同的领域知识，这对运维人员和运维工具提出了较高的要求。

2. 流程复杂：

事件管理的时间线如下，每个环节都需要提效才能达成事件管理的效率提升。

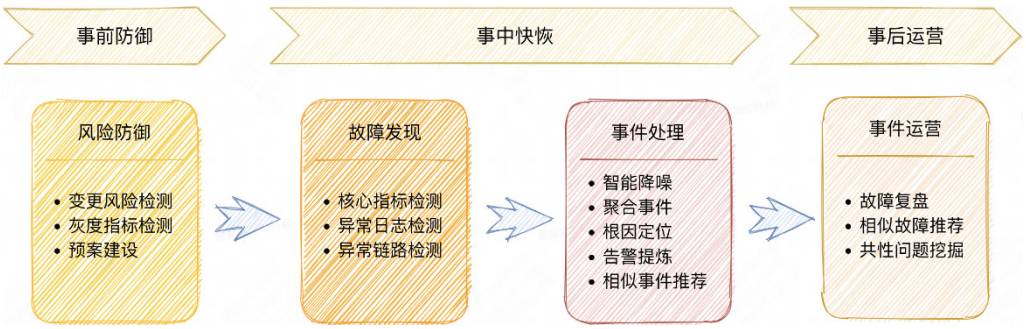


图 1　事件管理时间线

面对上述挑战，美团运维团队在过去几年建设了丰富的工具体系，基于专家经验、规则配置、流程管控等方式进行事件管理。本文聚焦的 AIOps 实践，是对上述工作的赋能，可拆解为四个模块：

- **风险预防——变更风险智能检测**：以用户和实体为对象，结合规则以及机器学习模型，对用户行为进行分析和异常检测。
- **故障发现——智能识别指标异常**：基于统计算法和机器学习算法识别指标的异常模式，帮助用户快速发现故障。
- **事件处理——诊断和预案推荐**：通过多模态数据和算法规则引擎来帮助用户快速定位故障，推荐止损预案。
- **事件运营——相似故障推荐**：基于 NLP 技术推荐相似故障复盘，挖掘共性问题。

2. 事件管理中 AI 能力总览

AIOps 在事件管理领域中的能力框架如下：

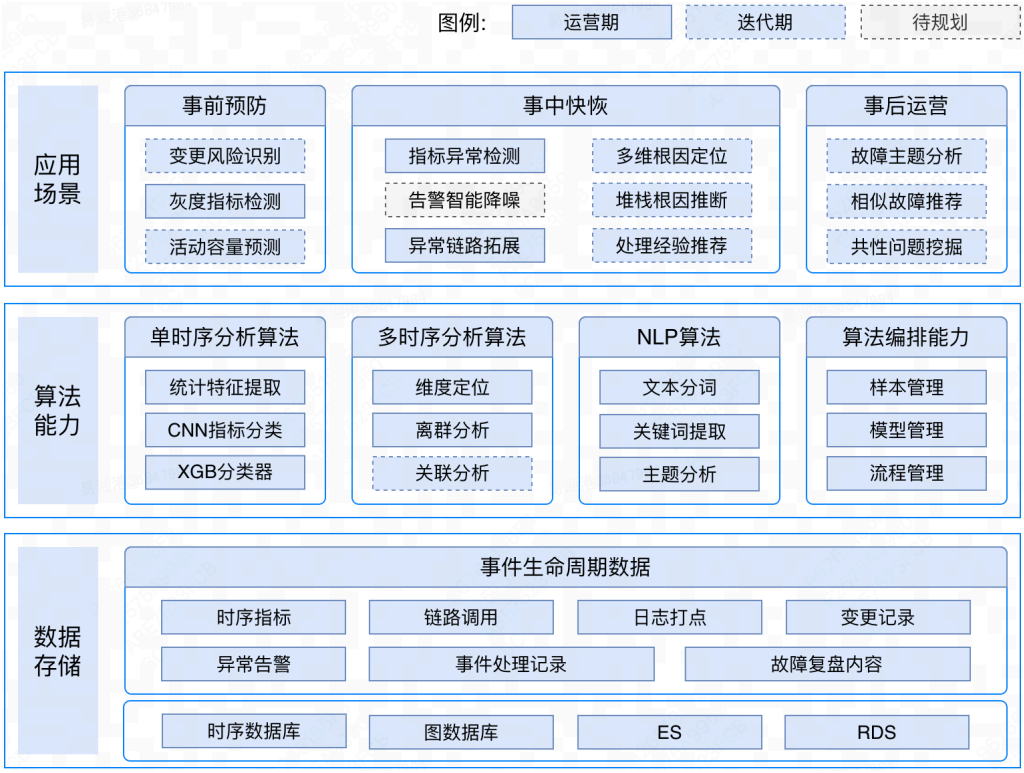


图 2　AIOps 事件管理领域能力框架

3. AIOps 之事件管理场景

3.1 事前预防

3.1.1 风险识别

变更检测分成前、中、后三个阶段。变更前风险预警的收益相对较高，因为它能够拦截异常的发生。但由于变更动作尚未发生，变更前检查所能获取到的参考信息少，检测难度比较大。变更中、后检测可以参考灰度组的变化情况以及是否有异常指标的出现，检测的参考信息更多，准确度更高。我们和 MCM- 线上变更管理平台（美团变更管控系统，后文简称 MCM）合作，共同探索了对变更前、变更中和变更后的一些异常进行检测与识别。

变更前

配置变更风险的检测和识别。当用户进行配置变更时，我们会进行配置项变更风险检

查。我们根据该配置项的历史合法变更数据挖掘出该配置项的约束规则，对当前变更值进行风险检测。约束项包括结构文本合法性、分隔符合法性、前后结构一致性等风险规则。

变更中 / 后

当灰度变更时部分系统指标会变化，比如集群中灰度机器的 QPS、4XX、5XX 指标可能会因为变更发生变化。系统需要识别出因为错误变更而导致的异常，屏蔽灰度变更影响导致的异常。

我们需要注意，如果直接采用全量未变更分组进行参考组对待检测组进行异常识别，会有一些干扰和噪声。同一个集群的机器会由于自身配置、承载流量任务等差异，其机器指标会产生出不同的分布和聚类情况。将实际差异较大的机器指标作为参考组，会干扰异常检测的结果。因此，我们需要筛选出和待检测指标相近的数据作为参考组，再在类内距离较近的多指标数据中识别出该指标是否符合正常模式。

以灰度变更组的数据为待检测数据点，以未变更组时序数据、变更历史时序数据作为参考组进行异常识别。算法思路如下：

1. **剔除参考数据的离群序列**：找到和检测数据相近的参考组，再做异常检测。我们使用优化后的自适应 DBSCAN[1] 进行聚类，排除参考组的离群时序序列。
2. **检测待检测数据是否异常**：识别待检测的时序数据在参考组中是否存在异常情况，包括点异常、上下文异常、子序列模式异常等异常特征。

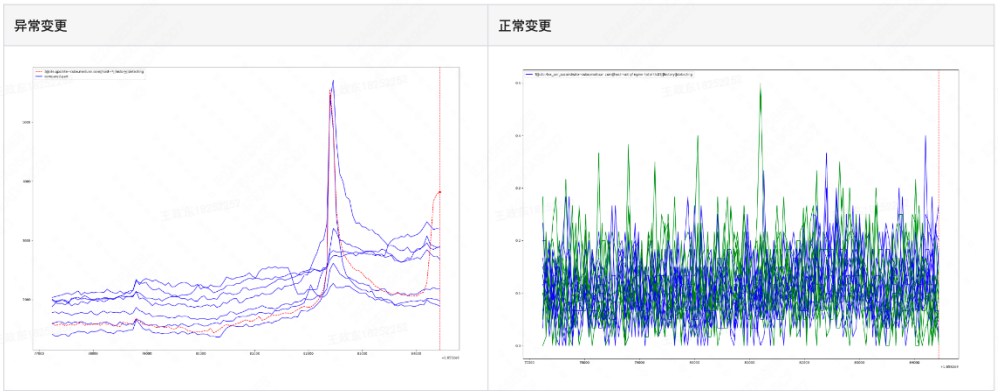


表 1　异常集群变更 vs 正常集群变更

该功能已经上线到 MCM，用于某核心平台系统集群变化后的变更复检。检测效果效果如下：



图 3　多指标复检效果图

当用户进行集群变更时，会触发集群维度和机器维度的变更检查，识别核心指标（QPS、4XX、5XX）是否有一些异常。当指标中存在异常时，指标数据详细展示区会有额外展示：

- **异常点标记线**：检测异常的时间点上会有红色的竖线标记。
- **异常项详细展示区**：详细展示出检测到异常的服务器主机名、时间点、指标值、与比对基线的偏离情况。
- **标记误报按钮**：如果发现异常为误报，可以点击按钮进行标记，便于后期算法复盘优化。

3.2 事中快恢

当故障事件发生后，需要尽可能降低服务的异常对其他用户的影响，提升服务的可用性。可以从 MTTD（平均检测时间）、MTTT（最短定位时间）、MTTR（平均修复时间）这三个指标入手。

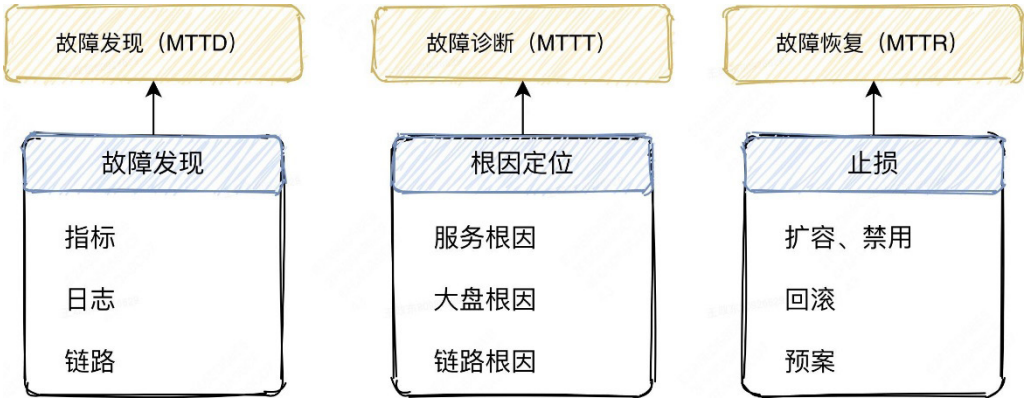


图 4　事件处理手段

3.2.1 异常发现

故障发现需要快速、准确。为避免误报，服务运维团队开发了一种基于历史上邻近的点分布相似（时序特征相似）思想的智能异常检测算法。如果当前待检测点相较其他历史参考点相对异常（存在点异常或者模式异常），检测流程会将异常点识别出来，并告知用户待测指标出现异常现象。

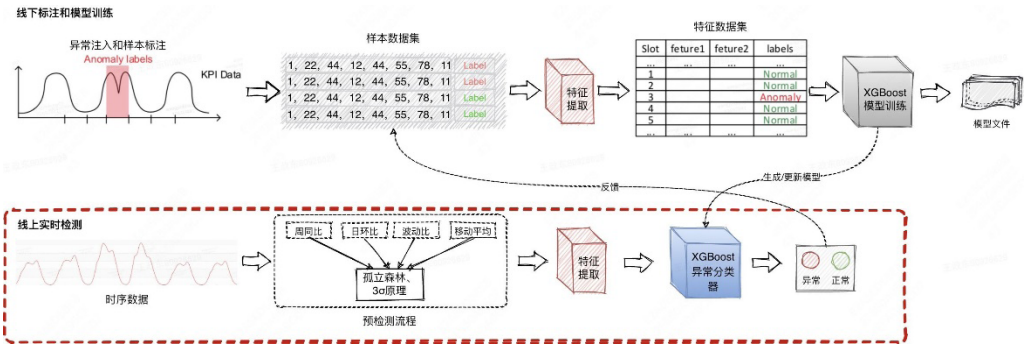


图 5　异常发现能力流程图

在进行实时检测流程中，待检测点会先进入预检测流程。预检测组件会拦截绝大多数正常点，而当预检测异常时，才会执行特征提取阶段，进入模型异常分类；同时分类结果通过反馈机制可以增加到样本集，提高模型泛化能力和精召率。整个算法流程训练、检测、反馈闭环。

该项能力为美团监控系统提供无阈值的时序检测能力。目前检测流程中的分类器在真实线上样本的精确率和召回率均在 98% 以上。团队会每周定时抽样核心指标并对检测结

果进行复盘，核心指标的异常检出准确率在 90% 左右。

3.2.2 根因诊断

在事件处理阶段，事件根因的自动定位可以大幅度降低定位时间（MTTT），帮助用户快速处理事件。由于美团现有系统规模极其庞大且复杂，因此不能用一种简单的定位方式来涵盖所有的错误根因。我们从多个方面来定位故障根因，包括链路异常、日志堆栈异常、服务异常等故障场景。这里我们将从两个方面来探讨根因定位的探索和实践。

异常链路拓展

雷达系统是美团的统一的事件管理平台，用于高效处理告警、事件和故障，同时也提供对公司内微服务系统的根因定位能力（后文简称为雷达）。识别微服务系统中的故障链路是重要且具有挑战性的工作，根据服务的调用情况构建服务调用图，并通过异常检测进行扩展和剪枝，以获得准确的异常链路图。拓扑图过大或过小都不利于根因定位。链路异常检测工作有如下要求：

1. **实时性高**：由于服务调用实时变化，异常计算工作不能过于耗时，过长的滞后结果将会导致拓扑图更新的延迟，异常检测没有价值。
2. **计算量大**：美团每分钟产生几十万级别的链路数据，并且每一种链路数据都包含调用次数、TP 耗时等关键指标。
3. **精召率高**：我们需要准确识别出当前链路是否存在异常，精准识别可以防止拓扑爆炸或者根因节点缺失。

为了解决以上问题，我们和数据库平台研发组合作，研发了一套基于预训练的大容量异常检测服务来进行链路异常检测，其具有大容量，低时延，准确率高的特点。算法使用了历史长时序来挖掘时序特征参数，并在实时检测中参考临近 120 分钟进行了波动过滤，可以在较短的时间内快速识别指标的异常程度，实现了每分钟百万级别的异常检测。整体检测的平均流程耗时在 1.5–3ms，检测的异常点精确率在 81%，异常点召回率在 82%，F1 值为 81%。

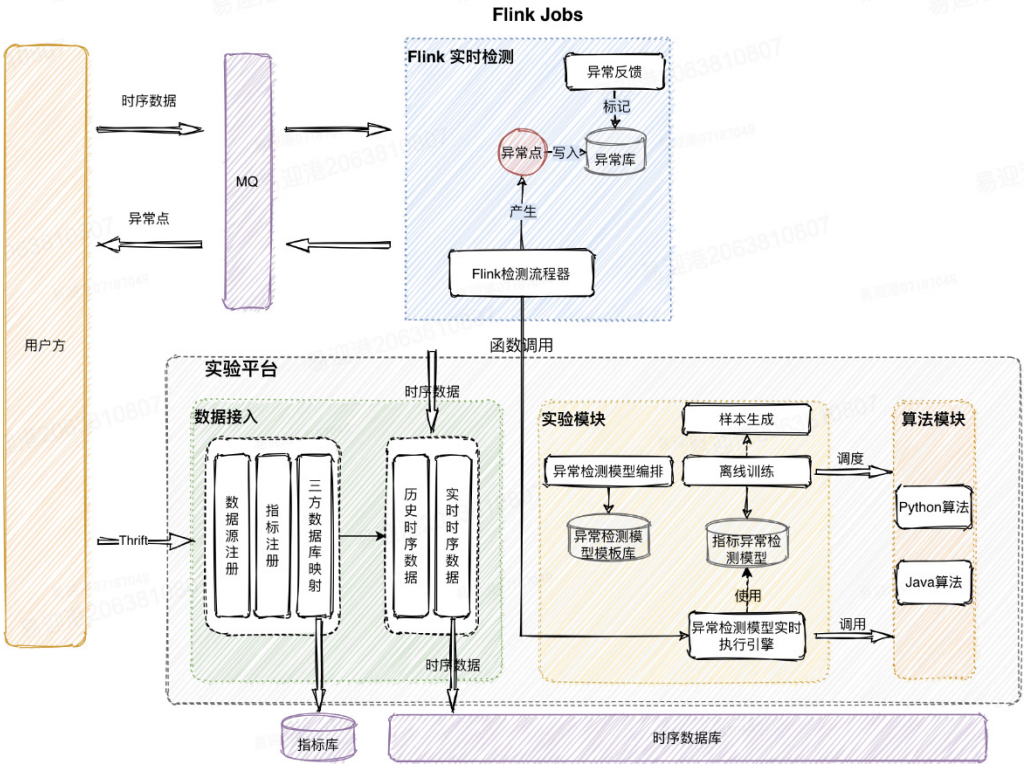


图 6　百万级别异常检测能力框架图

我们使用编排好的训练流程对单指标进行单模型参数建模，存放到离线模型数据库中。在实时检测过程中从数据库中加载对应的预训练参数，根据检测流程进行实时监测。

该工作的核心思想是：将大量的复杂的计算异步化，在实时流检测的过程中，大幅度降低实时浮点计算量，提高整体的计算容量。该项工作对接了雷达链路中的流量和 TP 线的识别，支持雷达在故障诊断的过程中，获取异常节点与邻接节点之间的调用量、耗时的波动，来获取准确的故障全景图，并获取节点间调用异常的准确情况。

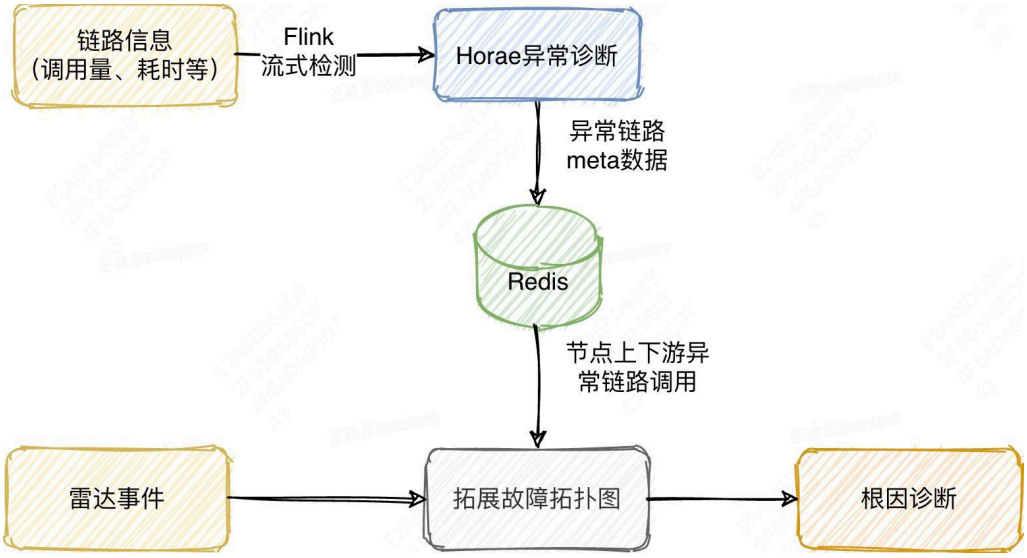


图 7　链路拓扑异常计算

异常拓扑图效果如下图所示，雷达链路的拓展根据流量异常、失败率上涨、耗时等多个方面进行拓展，可以有效地找到核心故障链路图：

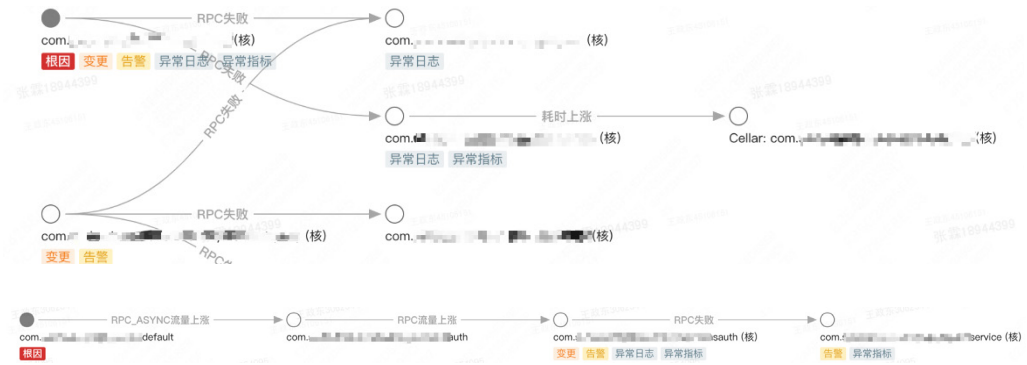


图 8　雷达异常链路

指标多维度根因定位

大盘有一部分是多维度的指标，由于其业务特性比较强，这些指标波动很难用通用系统指标进行异常定位。我们目前探索了从指标自身维度的异常特征来进行异常维度定位的工作。

总的 KPI 指标异常，需要处理人员人工下钻到不同维度分析。如果指标的维度较多，

人工分析成本巨大。该项工作的困难和挑战有以下两个方面。 – 首先，不同的组合的KPI 是相互依赖和影响的，真正的根因元素的 KPI 异常，可导致其他维度的 KPI 也发生变化，很难对 KPI 指标的根因做一个量化的判断。 – 其次，由于 KPI 拥有多维度的属性，因此随着维度的增加或粒度的细化，元素的数目往往呈现指数级增长的趋势，可能需要从成千上万的多维属性空间进行搜索；此外，对如此多的维度快速做预测也是一个挑战。

在算法的应用场景中，我们需要考虑算法的可落地性。什么时候触发、如何提升结果的准确度、有效性是我们需要解决的问题。

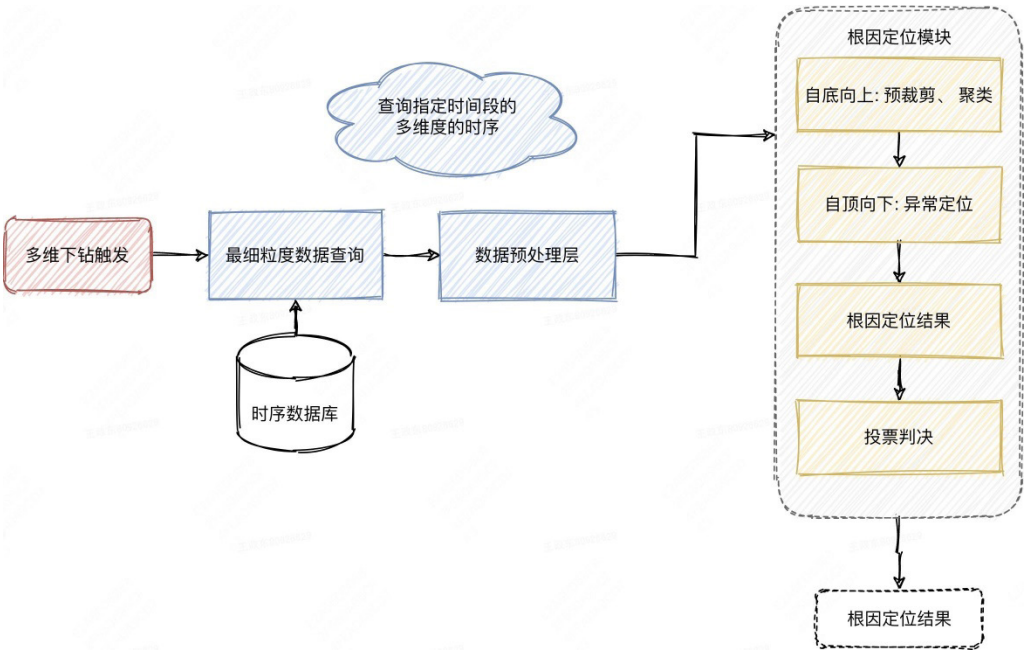


图 9　指标异常维度定位流程图

上图是我们执行异常维度定位的简易流程，我们在 Squeeze2 的基础上针对美团业务特性做了优化，其具有以下特点：

1. **自动化框定检测时间范围**：使用变点检测等算法自动框定时间区间，用户无需人工框定所需要的检测时间区间即可自动化的进行异常维度定位操作，并且该项工作提高了下钻的性能和准确度。
2. **多时间戳下钻，定位结果汇总**：为了减少因为单点抖动而产生的错误下钻，提高

算法的精确度，通过多时间戳的方式并行下钻分析，然后根据各个指标的不同特征，区分汇总结果，提高结果的可用性。

裁剪非关键根因，减少干扰维度的影响：对于最后的根因维度，会计算每一个子维度的整体重要占比，裁剪非重要根因，减少无意义维度带来的干扰。将下钻的根因编码解析，提高定位结果的可读性。

结果展示

当核心大盘指标出现异常时，系统就会自动触发下钻分析，将异常维度的分析结果推送到群里，帮助用户快速定位该指标的异常维度是什么。



表 2　告警与异常维度定位

3.2.3 相似事件推荐

雷达系统中经常出现相似事件，它们往往有着相似的根因，如同一个业务的促销活动、某个中间件故障等。如果我们能够根据当前事件的异常现象，找到历史上最相似的一些 Case 推荐给处理人，则能为事件的定位和处理提供参考，提高处理效率。我们实现了一套相似事件推荐算法，通过 NLP 技术和规则过滤，找到每个事件的 Top 历史相似事件并推荐给处理人。算法的整体流程如下：

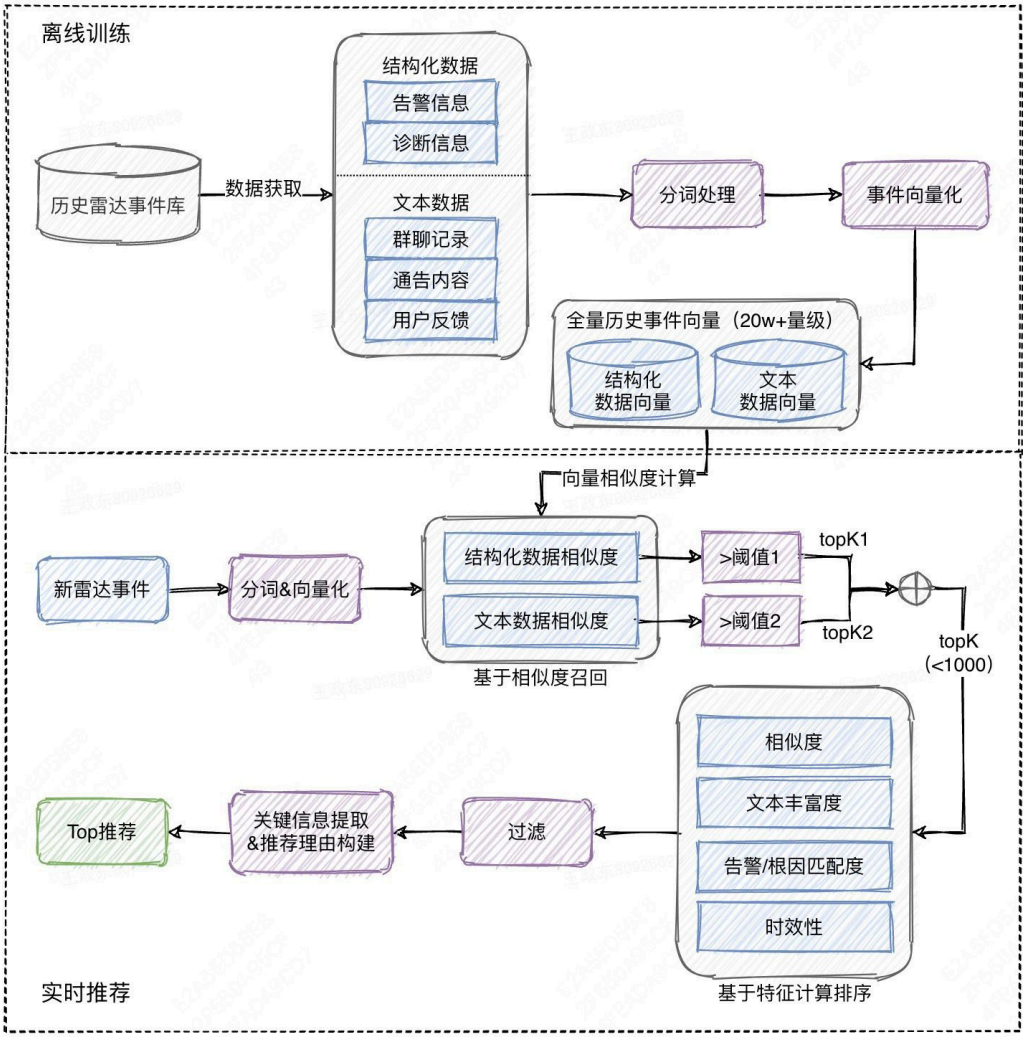


图 10　相似事件算法流程

整体而言，算法在离线阶段使用 NLP 技术，将每个历史雷达事件进行向量化并存储；在实时推荐时，算法将新的雷达事件进行向量化后，通过向量相似度搜索到最接近的历史事件，并通过一些规则计算的特征进行排序和过滤，得到最终可推荐给用户的 Top 相似事件。下面对一些实现细节进行介绍。

离线训练阶段

1）数据类型区分

一个雷达事件中包含的数据可以分为两种类型：结构化数据和文本数据。它们主要的区

别如下：

类型	来源	可重复度	作用
结构化数据	系统生成，包括：告警信息、诊断信息	高，经常重复出现在历史多个事件里面	告警是故障在系统层面的表征，体现故障的现象，可认为是故障的"What"维度
文本数据	用户生成，包括：群聊、通告、反馈等	低，不太可能在多个事件里面完全一致	用户在故障过程中的发言或反馈，更多是在讨论故障定位和处理，体现故障的原因、止损方案，可认为是故障的"Why"&"How"维度

表 3　事件数据结构

如上表所示，这两类数据的特点和作用有着很大的不同，且它们的可重复度也不一样。如果我们把两类数据都放到一个语料库中去训练一个向量化模型，会导致在后续的实时推荐阶段中，对于文本数据较多的事件产生“不公平”的现象：由于用户生成的文本数据的可重复度低，它们之间的相似度“天花板”会远低于结构化类型的数据，这就意味着一个事件的群聊越丰富，就越不容易找到相似的事件，这不符合我们的预期。

所以我们针对这两类数据分别构建语料库，通过文本向量化算法分别得到两个向量集，以便后续做更精细化的控制。

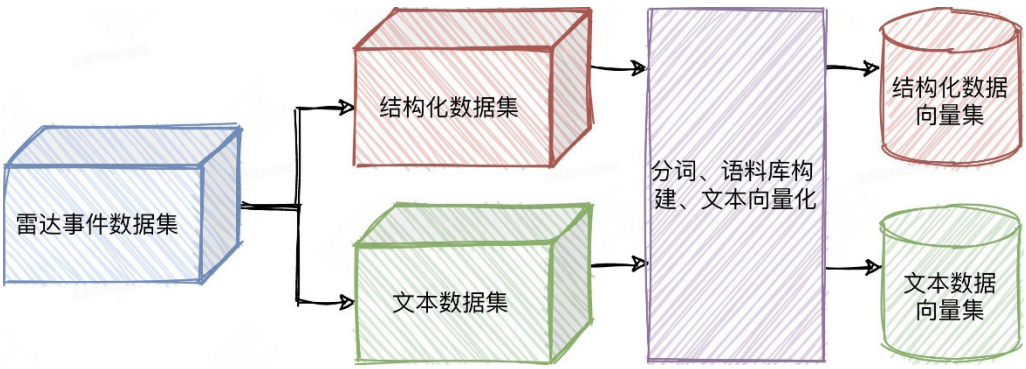


图 11　事件建模过程

2）分词 & 向量化

分词（tokenization）是将文档分解为以字词为单位的基本要素，方便后续处理。对于雷达事件中的文本类型数据，需要采用分词器进行分词，并去除停用词后得到 tokens（即分词后的词语列表）；对于结构化字段，则直接提取属性值或通过一些规则处理得到 tokens。

为提升文本分词效果，预先加载了 IT 领域词库、公司 Appkey 列表、公共服务名称作

为分词器的词库。经过分词后，对事件进行词频统计，再通过 Tfidf 算法计算各词语的权重。Tfidf 算法得到的向量长度等于词库的大小，向量第 i 个位置上的值表示词库中第 i 个词在事件中的权重，计算方式会综合考虑词语在当前事件的出现次数和在历史所有事件中的出现次数：

词 i 权重≈词 i 在当前事件中的出现次数词 i 在多少个历史事件中出现词 i 权重≈词 i 在当前事件中的出现次数词 i 在多少个历史事件中出现

一个词语在当前事件中出现得越多，且在越少的历史事件中出现过，说明它是一个比较关键的词语，Tfidf 算法将给予它比较大的权重。

注意，以上公式只是概念化表示，不是 Tfidf 算法的实际公式，对于 Tfidf 算法的具体实现感兴趣的读者可自行搜索了解。我们对事件的结构化数据和文本数据分别经过以上步骤进行处理，每个事件将被用两个 Tfidf 向量表征并存储起来，用于后续在实时推荐阶段计算事件相似度。

实时推荐阶段

1）基于向量相似度召回候选事件

在实时推荐阶段，我们同样将新的雷达事件分为结构化和文本类型数据，分别进行分词并向量化。然后，我们计算它们与历史事件向量的相似度，得到结构化和文本数据与所有历史事件的相似度。我们设定不同的阈值来召回候选的历史相似事件，在实践中，设定的文本数据相似度阈值要低于结构化数据的相似度。两类数据召回的相似事件取并集，作为候选的相似事件列表，它们是与当前事件具有一定相似度的历史事件，量级在 1000 个以下。

2）基于规则计算特征进行排序

一个历史事件是否值得推荐给用户，除了与当前事件的相似度之外，还需要考虑更多的维度，例如事件本身是否具有足够的文本内容可以参考，以及距离当前事件的时间是否比较接近。为了使值得推荐的事件被展现给用户，我们对每个候选事件通过规则计算一系列特征，用于后续的排序：

- **文本丰富度**：衡量历史事件的内容质量，其中群聊、通告、反馈等文本数据较丰

富的，能够提供更多关于处理、定位过程的信息，为当前事件的处理提供更多参考。

- **时效性**：衡量历史事件发生时间与当前事件的距离，越临近当前事件发生时间的历史事件，参考价值越大。
- **根因匹配度**：判断历史事件诊断根因是否与当前事件诊断根因一致，如果一致那么很有可能背后是同一个原因导致的问题。
- **告警匹配度**：衡量两个事件告警列表的相似程度，计算过程中会降低通用兜底类告警的权重，提高具有明确业务含义告警的权重。

以上特征，再加上召回阶段已经计算好的结构化数据、文本数据相似度，进行加权求和，得到最终的推荐得分，用于对候选事件进行排序。我们还需要对其进行过滤，去除不希望推荐给用户的历史事件。过滤的依据主要包括：对于系统发现事件，告警匹配度需要足够大，对于人为发现事件，推荐的历史事件的文本内容需要足够丰富。

3）关键信息提取 & 用户展示

为了使用户能够更快速地从历史事件中获取有效信息输入，我们从每个待推荐的历史事件提取出关键信息，包括由用户反馈的历史事件的根因、解决方案等信息，把这些附在推荐项上，在事件处理过程中推送到群里，辅助用户提高事件处理效率。同时，在 Web 端页面也会展示 Top 相似事件列表，以供用户参考。

案例展示：



表 4　相似事件能力结果展示

该功能上线后，有相似事件的 Case 覆盖率为 70% 左右，对于系统发现且有推荐历史

相似事件的 Case，其平均故障处理时长比无相似事件的 Case 缩短 28%。为了评估推荐的准确率，我们对有推荐相似事件处理经验且用户反馈了真实根因的 Case 进行了人工复盘，若推荐的历史事件与当前事件有类似的根因或解决方案，则标记为推荐准确，由此统计得到的推荐准确率约为 76%。

在复盘过程中，我们发现，推荐的准确率很大程度上取决于用户所配置的告警质量，对于粒度较粗的通用兜底类告警（如域名 5xx 告警）所产生的事件，推荐准确率会低于细粒度的、具有明确业务含义的告警所产生的事件。原因也是显然的：粗粒度的告警背后的根因可能千差万别，所以即使一个历史事件有着同样的告警，也会有很大可能不是同一个根因导致的故障；而细粒度的告警则与之相反。

在后续的优化中，对于相似事件的排序策略，我们可能需要根据事件中告警的粒度，来调整不同告警类型对于相似度的贡献大小，而告警的粒度如何衡量，则需要结合公司具体的现状和我们的经验判断。总而言之，在数据驱动的算法之外，需要结合一些专家知识来弥补数据的不足，从而使算法的输出更好地服务于用户。

3.3 事后运营

在故障事后，用户对于故障的运营和复盘有利于经验沉淀，避免相同的问题再次发生，对于稳定性的长期提升有着重要作用。

COE（Correction Of Error）是美团的故障复盘系统，以文本形式记录了公司大量的历史故障复盘内容。我们在 COE 系统中基于主题分析等 NLP 技术，实现了故障复盘的主题展示和相似推荐能力，旨在帮助用户找到更多相似的故障，挖掘出共性问题。目前该功能处于初期建设阶段，正在迭代和探索的过程中。

4. 总结和未来展望

本文简单介绍了团队在 AIOps 在事件管理这一领域的探索和实践。我们从三个关键的运维阶段——事前预防、事中处理以及事后运营，深入探讨 AIOps 在这些场景中的具体应用和优势。

之后我们也会从多个方面进一步探索 AIOps 在美团场景下的可能性。这里我们简单列举了几个可能且有价值的发展方向。

智能日志检测

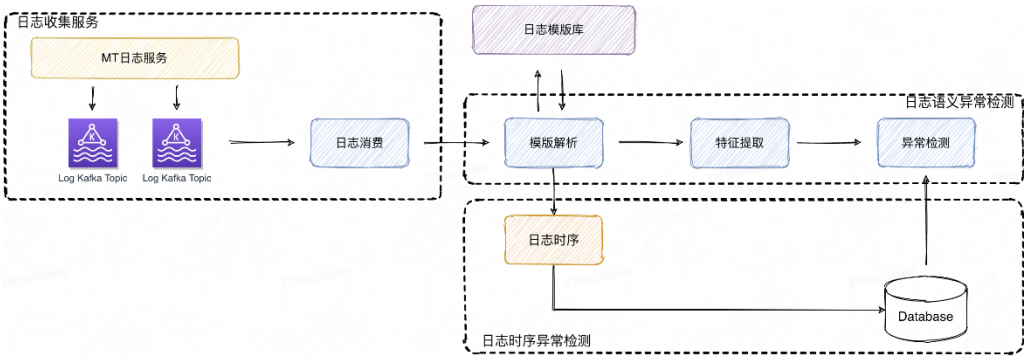


图 12　日志检测

日志异常检测通常由四个模块组成，包括日志收集上报、日志解析、特征提取以及异常检测器。我们期望通过提取日志的计数、序列、语义等特征，来动态识别当前服务的日志是否存在异常情况，

我们计划从两个方面进行探索：

- **日志模版时序异常**：通过 Drain3 等日志模版挖掘技术，将服务日志转化成日志模版时序。基于时序异常检测算法，可以识别日志模版徒增、异常日志出现等风险点。
- **日志模版语义异常**：通过解析日志模版的语义特征，结合机器学习和深度学习算法，识别当前日志是否存在语义异常。

智能化变更识别

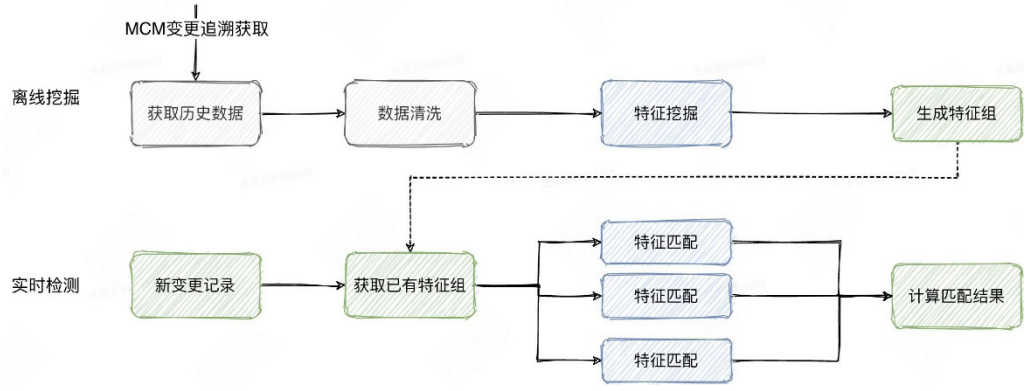


图 13　配置变更挖掘与识别

对于配置类型的变更（如：分布式内存配置变更、营销活动配置变更等），一旦变更人员疏忽填写错误或遗漏，会导致线上问题。同时，这类配置可能数量极其庞大，我们需要去动态分析识别每一配置对应的模型信息。

对于历史上正常完结没有回滚的配置变更，根据其变更值进行特征提取，学习该配置的Key、Value 变化规律，从统计学、数据建模等多个角度构造每个配置值的特征组。当新变更到来时，就转化成了特征相似匹配，从而发现人工填写错误的问题。

5. 本文作者

政东、迎港、张霖、俊峰等，均来自美团基础研发平台。

6. 参考文献

[1] Ester M, Kriegel H P, Sander J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *AAAI Press*. 1996.

[2] Li Z, Pei D, Luo C, et al. Generic and Robust Localization of Multi-dimensional Root Causes. *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), IEEE*. 2019.

[3] He P, Zhu J, Zheng Z, et al. Drain: An Online Log Parsing Approach with Fixed Depth Tree. *2017 IEEE International Conference on Web Services (ICWS), IEEE*. 2017.

[4] M Du, F Li. Spell: Streaming Parsing of System Event Logs. *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 2016.

[5] IBM. Drain3. https://github.com/IBM/Drain3

[6] Akiko A. An information-theoretic perspective of tf–idf measures. *Information Processing and Management*. 2003.

[7] David B, Andrew Ng, Michael J. Latent Dirichlet Allocation. *Journal of Machine Learning Research*. 2003.

[8] 曹臻，威远．基于AI算法的数据库异常监测系统的设计与实现．https://tech.meituan.com/2022/09/01/database-monitoring-based-on-ai.html

7. 招聘信息

基础技术部 – 服务运维部 – 故障管理开发组主要负责故障发现、故障定位、故障恢复、故障运营、告警中心、风险管理、数据仓库等工作。目前团队诚招高级工程师、技术专家。欢迎有兴趣的同学投送简历至 edp.itu.zhaopin@meituan.com。（邮件主题注明：运维工具）

2023精选论文

PIER- Permutation-Level Interest-Based End-to-End Re-ranking Framework in E-commerce

Xiaowen Shi*
Meituan
Beijing, China
shixiaowen03@meituan.com

Fan Yang*
Meituan
Beijing, China
yangfan129@meituan.com

Ze Wang†
Meituan
Beijing, China
wangze18@meituan.com

Xiaoxu Wu
Meituan
Beijing, China
wuxiaoxu04@meituan.com

Muzhi Guan
Meituan
Beijing, China
guanmuzhi@meituan.com

Guogang Liao
Meituan
Beijing, China
liaoguogang@meituan.com

Yongkang Wang
Meituan
Beijing, China
wangyongkang03@meituan.com

Xingxing Wang
Meituan
Beijing, China
wangxingxing04@meituan.com

Dong Wang
Meituan
Beijing, China
wangdong07@meituan.com

ABSTRACT

Re-ranking draws increased attention on both academics and industries, which rearranges the ranking list by modeling the mutual influence among items to better meet users' demands. Many existing re-ranking methods directly take the initial ranking list as input, and generate the optimal permutation through a well-designed context-wise model, which brings the evaluation-before-reranking problem. Meanwhile, evaluating all candidate permutations brings unacceptable computational costs in practice. Thus, to better balance efficiency and effectiveness, online systems usually use a two-stage architecture which uses some heuristic methods such as beam-search to generate a suitable amount of candidate permutations firstly, which are then fed into the evaluation model to get the optimal permutation. However, existing methods in both stages can be improved through the following aspects. As for generation stage, heuristic methods only use point-wise prediction scores and lack an effective judgment. As for evaluation stage, most existing context-wise evaluation models only consider the item context and lack more fine-grained feature context modeling.

This paper presents a novel end-to-end re-ranking framework named *PIER* to tackle the above challenges which still follows the two-stage architecture and contains two mainly modules named FPSM and OCPM. Inspired by long-time user behavior modeling methods, we apply SimHash in FPSM to select top-K candidates from the full permutation based on user's permutation-level interest in an efficient way. Then we design a novel omnidirectional attention mechanism in OCPM to better capture the context information in the permutation. Finally, we jointly train these two modules in an end-to-end way by introducing a comparative learning loss, which use the predict value of OCPM to guide the FPSM to generate better permutations. Offline experiment results demonstrate that PIER outperforms baseline models on both public and industrial datasets, and we have successfully deployed PIER on Meituan food delivery platform.

KEYWORDS

Re-ranking, End-to-End Learning, Recommender Systems

ACM Reference Format:
Xiaowen Shi, Fan Yang, Ze Wang, Xiaoxu Wu, Muzhi Guan, Guogang Liao, Yongkang Wang, Xingxing Wang, and Dong Wang. 2023. PIER: Permutation-Level Interest-Based End-to-End Re-ranking Framework in E-commerce. In *the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/xxxxxx.xxxxxx

1 INTRODUCTION

E-commerce applications such as JD.com and Meituan have a large number of items. To improve the user's decision-making efficiency, a slate which contains limited items is usually provided based on the user's interest. As shown in figure 1, a slate named Guess You Like which consists of three items, is displayed to users on Meituan food delivery platform.

Due to the rapid growth of deep learning techniques, many well-designed ranking models have been proposed to improve the recommendation performance, mainly focusing on feature interaction (e.g. Wide&Deep [9], DeepFM [14], xDeepFM [16]), user interest modeling (e.g. DIN [27], DIEN [26], ETA [8]), and so on. However, most existing ranking methods only model the CTR of the current item, but ignore the crucial mutual influence among contextual items. In order to model the influence of the arrangement of displayed items on user behaviors, the re-ranking stage is introduced to rearrange the initial list from the ranking stage.

*Equal contribution. Listing order is random.
†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 '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-xxxx-x/xx/xx…$15.00
https://doi.org/10.1145/xxxxxx.xxxxxx

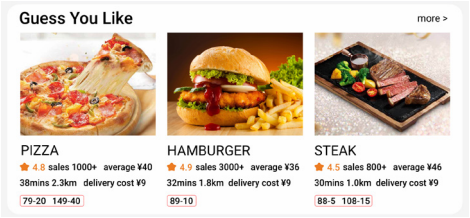


Figure 1: Guess You Like in Meituan.

Existing re-ranking methods can be divided into two categories [12]. The first category is the step-greedy re-ranking strategy [3, 12, 13, 29], which sequentially decides the display results of each position. Such methods only consider the preceding information but ignore the succeeding information, which is insufficient to obtain the optimal result. Different from the greedy strategy, another solution is the context-wise re-ranking strategy [1, 6, 11, 20, 25], which uses a context-wise evaluation model to capture the mutual influence among items and re-predict the CTR of each item. Methods like PRM [20], directly take the initial ranking list as input, and generates the optimal permutation based on the predict value given by context-wise model. These one-stage methods suffer an evaluation-before-reranking problem [25], that is, due to the order obtained after re-ranking is different from the initial order, inputting the re-ranked list will result in different prediction results. In order to resolve evaluation-before-reranking problem, a straightforward solution is to evaluate every possible permutation, which is global-optimal but is too complex to meet the strict inference time constraint in industrial system. Therefore, most existing re-ranking framework uses a two-stage architecture [11, 25] which consists of permutation generation and permutation evaluation. To be specific, they use some heuristic methods such as beam-search [21] to generate the suitable amount of candidate permutations firstly and then feed into the evaluation model to get the optimal permutation.

In order to improve the performance of the re-ranking stage under the two-stage architecture, on the one hand, the generated candidate permutations should contain the optimal permutation as much as possible. On the other hand, the context-wise evaluation model should predict as accurately as possible. However, existing methods in both stages can be improved through the following aspects:

- Generation stage. Some heuristic methods, such as the beam search algorithm [11], merely use point-wise prediction scores (i.e. item CTR) to generate candidate permutations, while ignoring the mutual influence between each item and its contexts in one permutation. In addition, since the generation stage is independent of the evaluation stage, the evaluation results cannot guide the generating process. Therefore, the quality of the generated candidate permutations lacks an effective judgment.
- Evaluation stage. Different from the point-wise item prediction, the evaluation of the permutations needs to use various types of contextual information to fully model the mutual influence among items. In addition to the influence of item context, there is also the fine-grained influence of feature context. These features form a variety of channels, and users may be interested in the features of a certain channel. For example, price-sensitive users will pay more attention to the comparison of price information in the context, which we call it as multi-feature channel competition problem.

To resolve the aforementioned issues, we propose a novel end-to-end re-ranking framework named Permutation-Level Interest-Based End-to-End Re-ranking (PIER). Our framework still follows the two-stage paradigm which contains two mainly modules named Fine-grained Permutation Selection Module (FPSM) and Omnidirectional Context-aware Prediction Module (OCPM). Inspired by long-time user behavior modeling methods [4, 8], we apply SimHash in FPSM to select top-K candidates from the full permutation based on user's permutation-level interest in an efficient way. Then in OCPM, we design a novel omnidirectional attention and context-aware predict mechanism to better capture the context information and predict the list-wise CTR of each item in the permutation. Finally, we integrate these two modules into one framework and training in an end-to-end way. We introduce a comparative learning loss, which use the predict value of OCPM to guide the FPSM to generate better permutations.

The main contributions of our work are summarized as follows:

- We propose a novel re-ranking framework named PIER, which integrates generation module and evaluation module into one model and can be trained in an end-to-end manner.
- We conduct extensive offline experiments on both public dataset and real-world industrial dataset from Meituan. Experimental results demonstrate the effectiveness of PIER. It is notable that PIER has been deployed in Meituan food delivery platform and has achieved significant improvement under various metrics.

2 RELATED WORK

An industrial recommender system typically consists of three stages[1]: matching [28], ranking [15, 24] and re-ranking [20]. Matching stage aims to recall thousands of relevant items from the whole item set. Ranking stage [9, 14, 27] point-wisely predict the click-through rate (or conversion rate, etc.) of recalled items. Re-ranking stage aims to find the best (e.g. maximization of the total clicks) permutation from the initial list given by the ranking model. In this paper, we mainly focus on re-ranking stage.

Typical re-ranking methods can be divided into two categories. The first category is the step-greedy re-ranking strategy [3, 12, 13, 29], which sequentially decides the display results of each position through recurrent neural network or approximation solution. Seq2slate [3] utilizes pointer-network and MIRNN utilizes [29] GRU to determine the item order one-by-one. Similarly, the client-side short video re-ranking framework of Kuaishou [13] combines a point-wise prediction model which takes ordered candidates list as input with beam-search to sequentially generate the final list. These methods ignore succeeding information, which is insufficient to obtain the optimal result.

[1]We do not discuss advertising-related stages (e.g. mix-ranking [7, 18], auctions [17]) in this paper.

Another category is context-wise re-ranking strategy [1, 6, 11, 20, 25], which uses a context-wise evaluation model to capture the mutual influence among items and re-predict the CTR/CVR of each item. Methods such as PRM [20] and DLCM [1] take the initial ranking list as input, use RNN or self-attention to model the context-wise signal and output the predict value of each item. The optimal permutation is sorted according to the predict value. Such methods bring an evaluation-before-reranking problem [25] and leads to sub-optimum. Similarly, methods such as EXTR [6] estimate pCTR of each candidate item on each candidate position, which are substantially point-wise models and thus limited in extracting exact context. In order to model the exact context of the permutation, a straightforward solution is to evaluate every possible permutation through a well-designed context-wise model. This is a global-optimal method but is too complex to meet the strict inference time constraint in industrial system. In order to reduce the complexity, PRS [11] adopts beam-search to generate few candidate permutations firstly, and score each permutation through a permutation-wise ranking model. Although heuristic methods are effective, they only use point-wise prediction scores and lack an effective judgment, which can be improved further.

In our scenario, we still adopt the two-stage architecture and focus on improving the overall performance by optimizing both two stages under the online time-consuming constraints.

3 PRELIMINARIES

Typically, an industrial recommender system consists of three consecutive stages: matching, ranking and re-ranking [11, 15, 20, 24, 25, 28]. Given a certain user involving his/her input ranking list $C = \{a_i\}_{i=1}^{N_o}$, the final displayed N_d items to user are formulated as the displayed list $\mathcal{P} = \{a_i\}_{i=1}^{N_d}$, where $N_d \leq N_o$.

If there is no re-ranking stages, the top-N_d items are selected as $\mathcal{G}$ from C for display. Therefore, the task of re-ranking stage is to learning a re-ranking strategy $\pi : C \rightarrow \mathcal{P}^*$, which aims to select and rearrange items from C, and subsequently recommends a better displayed list $\mathcal{P}^*$, with the aim of improving indicators such as CTR and GMV.

4 METHODOLOGY

We present the overview structure of PIER (**P**ermutation-level **I**nterest-based **E**nd-to-End **R**e-ranking) in Figure 2. Specifically, we take ranking list $C = \{a_i\}_{i=1}^{N_o}$ and user's permutation-level click behavior sequence $\mathcal{B} = \{b_i\}_{i=1}^{M}$ as the input of PIER. Then, we generate candidate permutations $\mathcal{G} = \{p_i\}_{i=1}^{T}$ on C through full permutation algorithm[2]. Next, we use the Fine-grained Permutation Selection Module (FPSM) to select top-K permutations from large number of candidate permutations. Finally, we use the Omnidirectional Context-aware Prediction Module (OCPM) to calculate scores of each permutation and select the best permutation p^* as the re-ranking list to display.

In FPSM, we propose the time-aware hamming distance calculated based on SimHash [5, 8, 19]. The top-K permutations are selected by sorting the distance between user's permutation-level click behaviors and candidate permutations. In OCPM, we design a novel omnidirectional attention unit to model the context information in each permutation and output the list-wise pCTR of each item in this permutation. The best permutation is selected based on the output score, i.e., the sum of list-wise pCTRs. The relationship between FPSM and OCPM is like the matching and ranking stages in recommender systems. We combine the two into one framework to generate the optimal re-ranking list. Next we will detail FPSM and OCPM separately.

[2]We set the number of items N_d in each permutation to 3 for illustration in figures.

4.1 Fine-grained Permutation Selection Module

For time-consuming considerations, some re-ranking methods utilize heuristic methods such as beam-search [11] to generate candidate permutations and use a well-designed prediction model to select the optimal permutation, but these heuristic methods are not consistent with the modeling target, leading to suboptimal performance. Inspired by long-term user behavior modeling methods such as ETA and SDIM [4], we propose FPSM to select top-K candidates through SimHash. Here, We use the user's historical click behaviors as target and then calculate the distance between it and all candidate permutations. If the distance is closer, we believe that the permutation can better match user's interest thus can bring greater revenue. In this manner, we can not only reduce the time complexity, but also can make consistent selection by training the

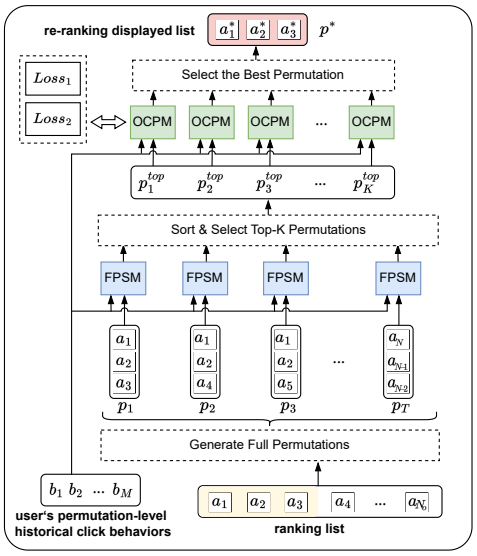


Figure 2: Overview of our framework PIER. PIER takes ranking list and user's permutation-level historical click behaviors as input, and outputs a re-ranking list to display with the help of FPSM and OCPM.

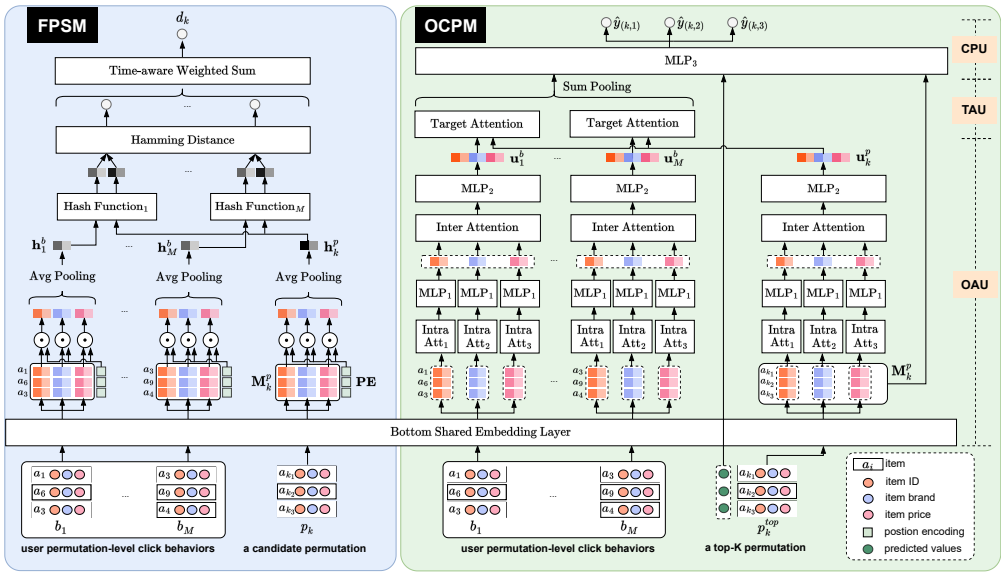


Figure 3: The structures of Fine-grained Permutation Selection Module (FPSM) and Omnidirectional Context-aware Prediction Module (OCPM). FPSM takes user permutation-level click behaviors and a candidate permutation as input and outputs the distance score of this permutation. OCPM takes user permutation-level click behaviors and a top-K permutation selected by FPSM as input and outputs the predicted list-wise pCTR of each item in this permutation. Best view in color.

FPSM and the prediction model in an end-to-end way. Next, we will introduce how to select the top-K permutations through FPSM.

As shown in the left part of Figure 3, we first use the bottom shared embedding layers to extract the embeddings from raw inputs. For permutation p_k, we denote the embedding matrix $\mathbf{M}_k^p$ as follows:

$$\begin{aligned}\mathbf{M}_k^p &= \begin{bmatrix} \mathbf{E}_{0;}^{p_k} \\ \mathbf{E}_{1;}^{p_k} \\ \cdots \\ \mathbf{E}_{N_d;}^{p_k} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{E}_{;0}^{p_k} & \mathbf{E}_{;1}^{p_k} & \cdots & \mathbf{E}_{;N_f}^{p_k} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{e}_{0;0}^{p_k} & \mathbf{e}_{0;1}^{p_k} & \cdots & \mathbf{e}_{0;N_f}^{p_k} \\ \mathbf{e}_{1;0}^{p_k} & \mathbf{e}_{1;1}^{p_k} & \cdots & \mathbf{e}_{1;N_f}^{p_k} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{e}_{N_d;0}^{p_k} & \mathbf{e}_{N_d;1}^{p_k} & \cdots & \mathbf{e}_{N_d;N_f}^{p_k} \end{bmatrix} \in \mathbb{R}^{N_d \times N_f \times D}, \end{aligned} \tag{1}$$

where N_d is the number of items in each permutation, N_f is the number of feature fields (i.e., ID, category, brand and so on) in each item, D is the dimension of the embedding-transformed feature field, $\mathbf{e}_{i;j}^{p_k} \in \mathbb{R}^D$ is the embedding of the i-th item's j-th feature field in permutation p_k, $\mathbf{E}_{i;}^{p_k} \in \mathbb{R}^{N_f \times D}$ is the embedding matrix of the i-th item in permutation p_k, and $\mathbf{E}_{;j}^{p_k} \in \mathbb{R}^{N_d \times D}$ is the embedding matrix of the j-th feature field in permutation p_k. Analogously, the embedding matrix of the m-th permutations in user's permutation-level history click behaviors is formulated as $\mathbf{M}_m^b \in \mathbb{R}^{N_d \times N_f \times D}$.

Next, we generate the position encoding matrix $\mathbf{PE} \in \mathbb{R}^{N_d \times D}$ for each permutation as follows:

$$\begin{aligned} \mathrm{PE}_{(i,2d)} &= \sin(i/10000^{2d/D}), \\ \mathrm{PE}_{(i,2d+1)} &= \cos(i/10000^{2d/D}), \\ \mathbf{PE} &= \begin{bmatrix} \mathrm{PE}_{(0,0)} & \mathrm{PE}_{(0,1)} & \cdots & \mathrm{PE}_{(0,D)} \\ \mathrm{PE}_{(1,0)} & \mathrm{PE}_{(1,1)} & \cdots & \mathrm{PE}_{(1,D)} \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{PE}_{(N_d,0)} & \mathrm{PE}_{(N_d,1)} & \cdots & \mathrm{PE}_{(N_d,D)} \end{bmatrix} \in \mathbb{R}^{N_d \times D}. \end{aligned} \tag{2}$$

Then, the embedding matrices of each feature field are multiplied by the position encoding matrix $\mathbf{PE}$ respectively and then merged into corresponding permutation representation $\mathbf{h}_k^p$ by average pooling, as follows:

$$\mathbf{h}_k^p = \frac{1}{N_f} \sum_{i=1}^{N_f} \text{Avg-Pool}\left(\mathbf{E}_{:,i}^{p_k} \odot \mathbf{PE}\right), \quad \forall k \in [N_o]. \tag{3}$$

Analogously, the representation of m-th permutation in user's permutation-level history click behaviors is formulated as $\mathbf{h}_m^b$.

In our scenario, users are more likely to click the permutations which are closer to the user's interest. We use user's permutation-level history click behaviors to represent user's interest and calculate the distance between user's interest and each candidate permutation. Specifically, we utilize the random projection schema (SimHash) [5, 8, 19] to calculate the similarity between the representations of user clicked permutations and the representations of candidate permutations. We first generate M different hash functions corresponding to M user's permutation-level behaviors. For each candidate permutation p_k, we hash its representations $\mathbf{h}_k^p$ with M different hash functions and calculate the hamming distance between it and each user's permutation-level behavior, as follows:

$$\mathrm{Sim}(p_k, b_m) = \mathrm{Hash}_m(\mathbf{h}_k^p, \mathbf{h}_m^b), \quad \forall m \in [M], \forall k \in [N_o]. \tag{4}$$

Meanwhile, the more recent behavior, the more it can reflect the user's current interest and will be given a higher weight in similarity calculation. So we weight these distance according to the occurrence time of each behavior to obtain the time-aware hamming distance, as follows:

$$d_k = \sum_{m=1}^{M} w_m \cdot \mathrm{Sim}(p_k, b_m), \quad \forall k \in [N_o]. \tag{5}$$

where w_m is the time-aware weight of m-th behavior.

Finally, we sort candidate permutations based on their distance scores and select top-K permutations $\mathcal{P}^{top-K} = \{p_i^{top}\}_{i=1}^{K}$ with the smallest distance scores as the output of FPSM.

Since FPSM shares bottom embedding layers with OCPM and fixes random vectors for hash function and position encoding, it dose not have its own independent parameters that need to be trained. In order to ensure the quality of the selected top-K permutations during training, we propose a contrastive loss to improve the performance of FPSM. The detail of contrastive loss is discussed in Section 4.3.2.

4.2 Omnidirectional Context-aware Prediction Module

For each candidate permutation selected by FPSM, we then use OCPM to predict the pCTR of each item through three consecutive unit: Omnidirectional Attention Unit (OAU), Target Attention Unit (TAU), Context-aware Prediction Unit (CPU). The architecture of OCPM is shown in the right part of Figure 3 and we will introduce each unit of OCPM separately.

4.2.1 Omnidirectional Attention Unit (OAU). When showing a permutation to users, on the one hand, they may pay attention to different displayed information such as price, delivery fee, rating, etc. On the other hand, the competitive relationship between the same feature of different items will also affect the user's behavior. For instance, placing expensive items ahead cheap items can stimulate user's desire to click on cheap one. From this point of view, in OCPM, we design an omnidirectional attention unit to effectively modeling the information of each permutation.

Specifically, OCPM first uses the same bottom shared embedding layers as FPSM to extract embeddings from raw inputs. Then we use N_f parameter-independent self-attention layers [22] to calculate the mutual influence of different items in each field separately and output corresponding matrix $\mathbf{H}_j^{p_k}$, as follows:

$$\mathbf{H}_j^{p_k} = \mathrm{soft\,max}\left(\frac{\mathbf{Q}_j^{p_k}\mathbf{K}_j^{p_k\top}}{\sqrt{D}}\right)\mathbf{V}_j^{p_k}, \quad \forall j \in [N_f], \forall k \in [K], \tag{6}$$

where $\mathbf{Q}_j^{p_k}, \mathbf{K}_j^{p_k}, \mathbf{V}_j^{p_k}$ represent query, key, and value for the j-th field in the k-th target permutation, respectively. D denotes feature dimension of each feature. Here, query, key and value are transformed linearly from $\mathbf{E}_{:j}^{p_k}$, as follows:

$$\mathbf{Q}_j^{p_k} = \mathbf{E}_{:j}^{p_k}\mathbf{W}_j^Q, \mathbf{K} = \mathbf{E}_{:j}^{p_k}\mathbf{W}_j^K, \mathbf{V} = \mathbf{E}_{:j}^{p_k}\mathbf{W}_j^V, \forall j \in [N_f], \forall k \in [K], \tag{7}$$

where $\mathbf{W}_j^Q, \mathbf{W}_j^K, \mathbf{W}_j^V \in \mathbb{R}^{D\times D}$. Then $\mathbf{H}_j^{p_k}$ are input into a Multi-Layer Perceptrons (MLP) layer to generate representation:

$$\begin{aligned} \mathbf{h}_j^{p_k} &= \mathrm{MLP}_1\left(\mathbf{H}_j^{p_k}\right), \quad \forall j \in [N_f], \forall k \in [K], \\ \mathbf{Z}^{p_k} &= \left[\mathbf{h}_1^{p_k} \quad \mathbf{h}_2^{p_k} \quad \dots \quad \mathbf{h}_{N_f}^{p_k}\right] \in \mathbb{R}^{N_f\times D}, \quad \forall k \in [K], \end{aligned} \tag{8}$$

After that, an inter-field self-attention layer is introduced to calculate the mutual influence between different fields in each permutation and output the final representation of the permutation $\mathbf{u}_k^p$ by an MLP layer, as follows:

$$\mathbf{u}_k^p = \mathrm{MLP}_2\left(\mathbf{H}_j^{p_k}\right) = \mathrm{MLP}_2\left(\mathrm{soft\,max}\left(\frac{\mathbf{Q}'_{p_k}\mathbf{K}'^{\top}_{p_k}}{\sqrt{D}}\right)\mathbf{V}'_{p_k}\right), \forall k \in [K], \tag{9}$$

where $\mathbf{Q}'_{p_k}, \mathbf{K}'_{p_k}, \mathbf{V}'_{p_k}$ represent query, key, and value for the k-th permutation and are transformed linearly from $\mathbf{Z}^{p_k}$, as follows:

$$\mathbf{Q}'_{p_k} = \mathbf{Z}^{p_k}\mathbf{W}^{Q'}, \mathbf{K}'_{p_k} = \mathbf{Z}^{p_k}\mathbf{W}^{K'}, \mathbf{V}'_{p_k} = \mathbf{Z}^{p_k}\mathbf{W}^{V'}, \forall k \in [K]. \tag{10}$$

Analogously, the final representation of m-th permutation in user's permutation-level history click sequence is formulated as $\mathbf{u}_m^b$. Through these two self-attention layers, we can effectively model the relationship between different items and different feature fields.

4.2.2 Target Attention Unit (TAU). Zhou et al. [27] have proved that historical behaviors, which are more relevant to the target, can provide more information for the model's predict. We use a permutation-level target attention unit to model the interactions between target permutation and each permutation in historical behaviors, as follows:

$$\begin{aligned} \mathbf{w}_{m;k} &= \mathbf{u}_m^b \cdot \mathrm{MLP}_{\mathrm{Att}}\left(\mathbf{u}_k^p || \mathbf{u}_m^b || (\mathbf{u}_k^p \odot \mathbf{u}_m^b) || (\mathbf{u}_k^p - \mathbf{u}_m^b)\right), \\ \mathbf{w}_k &= \text{Sum-Pool}\left(\left[\mathbf{w}_{1;k}, \dots, \mathbf{w}_{M;k}\right]\right), \forall m \in [M], \forall k \in [K]. \end{aligned} \tag{11}$$

where $||$ means concatenation. The output representation $\mathbf{w}_k$ of TAU is regarded as the representation of user's permutation-level interest on the target permutation and is used as input of next unit.

4.2.3 Context-aware Prediction Unit (CPU). In Context-aware Prediction Unit, we use a parameter-sharing MLP layer to predict the list-wise pCTR of each item in each permutation. Taking the k-th target permutation as an example, the inputs of CPU consist of four parts: representation of the k-th target permutation $\mathbf{u}_k^p$, user's permutation-level interest on the k-th target permutation $\mathbf{w}_k$, the original predicted values (i.e., point-wise pCTRs of each

item) $\{v_i^{pk}\}_{i=1}^{N_d}$ and embedding matrix of the k-th target permutation $\mathbf{M}_k^p$. Then the list-wise pCTR of the t-th item in the k-th target permutation is predicted as follows:

$$\hat{y}_{(k,t)} = \sigma\Big(\text{MLP}_3(\mathbf{u}_k^p||\mathbf{w}_k||\{v_i^{pk}\}_{i=1}^{N_d}||\mathbf{M}_k^p)\Big), \tag{12}$$

where σ is Sigmoid Function. In PIER, the score of each permutation is easily obtained by summing the output list-wise pCTR. However, It should be noticed that by using the framework of PIER, the score of each permutation can be conveniently adjusted according to business needs, such as using GMV instead of CTR.

4.3 Model Training

In training stage, We first use the permutation of real exposures to train the OCPM, then jointly train the two modules by adding a contrastive learning loss.

4.3.1 Pre-training of OCPM. In order to accurately evaluate the selected top-K permutations, we first pre-train the OCPM using the samples collected from online log. The inputs are the permutations which are real displayed and the labels are whether the items in each displayed permutation are clicked. Then the loss of the OCPM is calculated as follows:

$$Loss_1 = \sum_{t=1}^{N_d}\Big(-y_{(i,t)}\log(\hat{y}_{(k,t)}) - (1-y_{(i,t)})\log(1-\hat{y}_{(i,t)})\Big), \tag{13}$$

where subscript i is the index of samples and t is the index of displayed items.

4.3.2 Joint training of PIER. In the joint training phase, since we use fix random vector as 'hash function' and 'position encoding', and the embeddings are shared between FPSM and OCPM, FPSM does not need to update gradients. When the embedding is updated, the hash signature is updated correspondingly. So how to improve the quality of the selected top-K permutations during the training process. Here, we propose a contrastive loss to ensure this goal. The main idea is that the difference between the average pCTR of the selected top-K permutations and the average pCTR of any K permutations in unselected permutations is as large as possible. Therefore, the final loss in the joint training phase is:

$$Loss_2 = -\sum_{k=1,k'\notin[K]}^{K}\Big(\frac{1}{N_d}\sum_{t=1}^{N_d}\hat{y}_{(k,t)} - \frac{1}{N_d}\sum_{t=1}^{N_d}\hat{y}_{(k',t)}\Big)^2, \tag{14}$$

where subscript i is the index of samples and t is the index of displayed items.

Finally, we sample a batch of samples $\mathcal{B}$ from the dataset and update PIER using gradient back-propagation w.r.t. the loss:

$$L(B) = \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}}\Big(Loss_1 + \alpha \cdot Loss_2\Big), \tag{15}$$

where α is the coefficient to balance the two losses.

5 EXPERIMENTS

In this section, we conducted extensive offline experiments and online A/B test to evaluate the effectiveness of our framework[3]. For offline experiments, we will first compare OCPM with existing baselines and analyze the role of different designs in it. Given the same prediction model, we will next compare the performance of our FPSM with other generative method. Finally, we will verify how different hyper-parameter settings (e.g., K, α) affect the performance of our framework. For online experiments, we will compare our framework with the existing strategy deployed on the Meituan platform using an online A/B test.

[3]The code is publicly accessible at https://github.com/Lemonace/PIER_code.

5.1 Experimental Settings

5.1.1 Dataset. In order to verify the effectiveness of our framework, we conduct sufficient experiments on both public dataset and industrial dataset. For public dataset, we choose Avito dataset. For industrial dataset, we use real-world data collected from Meituan food delivery platform. Table 1 gives a brief introduction of the datasets.

- **Avito**. The public Avito dataset contains user search logs and metadata from avito.ru, which contains more then 36M ads, 1.3M users and 53M search requests. The full features include user search information(e.g., userid, searchid, and search date) and ad information(e.g., adid, categoryid, title and so on). Each search id corresponds to a search page with multiple ads. For each user, We rank his search pages in increasing order based on the search date, and use the first T-1 search pages as the behavior pages, and the ads in the T-th search page as the target ads to be predicted. Here we use the data from 20150428 to 20150514 as the training set and the data from 20150515 to 20150520 as the testing set to avoid data leakage.
- **Meituan**. The industrial Meituan dataset is collected on Meituan food delivery platform during April 2022, which contains user information(e.g., userid, gender, age), ad information(e.g., adid, categoryid, brandid and so on). According to the date of data collection, we divide the dataset into training and test sets with the proportion of 8:2.

5.1.2 Evaluation Metrics. For offline experiments, we use different metrics for different modules. Specifically, we use the widely adopted AUC metric to evaluate the effectiveness of the prediction module and use the following metrics to evaluate the whole framework:

- **HR(Hit Ratio)@1** [2]. For each data, HR is 1 only when the top-K permutations selected by generative methods contains the best permutation.
- **Cost**. Cost means the overall time-consuming for different re-ranking frameworks.

For online experiments, we compare our proposed framework with existing method through CTR, GMV and inference time.

5.1.3 Hyperparameters. The hidden layer sizes of MLP_1, MLP_2, MLP_3 are $(128, 64, 32)$, $(60, 32, 20)$, and $(50, 20)$, respectively. the learning rate is 10^{-3}, the optimizer is Adam and the batch size is 1,024. The α is 0.1, the embedding size is 8 and the length of user

Table 1: Statistics of the datasets.

Dataset	#Requests	#Users	#Ads
Avito	53,562,269	1,324,103	23,562,269
Meituan	230,525,531	3,201,922	98,525,531

Table 2: The experimental results about AUC and Logloss on two datasets.

Model	Avito		Meituan	
	AUC	LogLoss	AUC	LogLoss
DNN	0.6876	0.0483	0.6538	0.1917
DCN	0.6896	0.0483	0.6550	0.1916
PRM	0.7131	0.0481	0.6718	0.1899
EXTR	0.7114	0.0481	0.6704	0.1901
Edge-Rerank	0.7163	0.0479	0.6694	0.1909
OCPM	**0.7320**	**0.0471**	**0.6822**	**0.1891**

behavior sequence is 5. For Avito dataset, the length of ranking list and re-ranking list are both 5, thus the length of full permutation is 120 and **K** is set to 10. For Metuan dataset page, we select 3 items from the initial ranking list which contains 10 items, thus the length of full permutation is 720 and **K** is set to 100.

5.2 Offline Experiments For OCPM

5.2.1 Baselines. We compare OCPM with both point-wise and list-wise representative methods as baselines. We select DNN, DCN as point-wise baselines and PRM, EXTR and Edge-Rerank from Kuaishou as list-wise baselines. A brief introduction of these methods are as follows:

- **DNN** [10]. DNN is a basic deep learning method for CTR prediction, which applies MLP for high-order feature interaction.
- **DCN** [23]. DCN explicitly applies feature crossing at each layer, requires no manual feature engineering, and adds negligible extra complexity to the DNN model.
- **PRM** [20]. PRM adjusts an initial list by appling the self-attention mechanism to capture the mutual influence between items.
- **EXTR** [6]. EXternality TRansformer regards target ad with all slots as query and external items as key&value to model externalities in all exposure situations.
- **Edge-Rerank** [13]. Edge-Rerank combines an on-device ranking model and an adaptive beam search method to generate context-aware re-ranking result.

5.2.2 Performance Comparison. Table 2 summarizes the results of offline experiments. All experiments were repeated 5 times and the averaged results are reported. We have the following observations from the experimental results: i) All re-ranking listwise model (e.g. PRM, EXTR) makes great improvements over point-wise model (e.g. DNN, DCN) by modeling the mutual influence among contextual items, which verifies the impact of context on user clicks behavior. ii) Compare with transformer-based model(e.g. PRM, ETXR), Edge-Rerank also improve the CTR prediction because of they use the history sequence and previous item information. iii) Our proposed OCPM brings 0.0189/0.0104 absolute AUC on Avito/Metuan dataset gains over the state-of-the-art independent baseline which is a significant improvement in industrial recommendation system.

5.2.3 Ablation Study. To explore the effectiveness of different modules in OCPM, we conduct ablation studies on Avito and Meituan dataset. All experiments were repeated 5 times and the averaged AUC is reported:

Table 3: Result of ablation experiment on different parts in re-ranking model.

Model	Avito		Meituan	
	AUC	LogLoss	AUC	LogLoss
OCPM	**0.7320**	0.0483	**0.6822**	0.1917
- OAU	0.7198	0.0483	0.6774	0.1916
- TAU	0.7263	0.0481	0.6786	0.1899

- OCPM (-OAU) blocks the omnidirectional attention module. OAU aims to capture item context and feature context information, which could model the competitive relationship between the same attribute of different item. As shown in Table 3, AUC decreases by 0.0112/0.0048, suggesting that extracting context information is crucial and the proposed OAU meets this requirement.
- OCPM (-TAU) does not use the target attention unit. TAU aims to compute the interactions of target permutation and each permutation in historical behavior. As shown in Table 3, AUC decreases by 0.0057/0.0036 on Avito/Metuan dataset, suggesting that our TAU are capable to capture user history page-level interest.

5.3 Offline Experiments For PIER

5.3.1 Baselines. We compare our whole framework which combines FPSM and OCPM with some heuristic generative methods as baselines. All these methods use a fixed OCPM but the candidate permutations are generated in different ways. A brief introduction of these methods is as follows:

- **Random & OCPM**. We randomly select K permutations as candidates.
- **PRS(Beam Search & OCPM)** [11]. We use the beam search method to generate K candidate permutations based on the cumulative CTR.
- **Full Permutation & OCPM**. We directly feed all candidate permutations into the prediction model and select K permutations based on average CTR. This can be seen as the upper bound of our framework.

5.3.2 Performance Comparison. Table 4 summarizes the results of offline experiments and we have the following observations: i) Intuitively, PIER achieves great improvements over random and beam-search methods on finding the best permutation on both public dataset and industrial dataset with a small increase in time complexity. One reasonable explanation is that FPSM can select better permutations through the guidance of contrastive loss. ii) We still have much room for improvement compared with the full permutation method on finding the best permutation, but we greatly reduce the time cost of the whole framework.

5.3.3 Ablation Study. To verify the impact of different units (i.e., time-aware weighting, contrastive loss), we study two ablated variants of PIER framework:

- PIER (-time-aware weighting) does not use the time-aware weight on each behavior and treat them as equally important.
- PIER (-contrastive Loss) removes the contrastive loss for improving the quality of the selected top-K permutations, i.e., $\alpha = 0$.

Table 4: HR and Cost of different methods.

Model	Avito		Meituan	
	HR	Cost (ms)	HR	Cost (ms)
Random & OCPM	0.09	**10.8**	0.13	**70.5**
PRS (Beam-Search & OCPM)	0.62	16.8	0.72	78.7
Full Permutation & OCPM	1.00	100.5	1.00	430.5
PIER (FPSM & OCPM)	**0.78**	17.5	**0.84**	85.3
- time-aware weighting	0.74	17.3	0.70	84.8
- contrastive loss	0.52	17.5	0.61	85.1

Since removing a certain unit requires re-training the entire framework, we try to ensure the AUC of OCPM are as close as possible to guarantee the comparability of the ablation experiment. Judging from the experimental results, we have the following findings: i) The performance gap between w/ and w/o time-aware weighting on distance calculation has little impact on public dataset but the performance on industrial dataset is affected to some extent. This indicates that in the Meituan scenario, users' interests remain consistent in a short period of time. ii) The decline in the performance without contrastive loss is obvious on both dataset. This indicates that the auxiliary loss enables FPSM to select better permutations.

Table 5: Result of ablation experiment on different parts in PIER

Settings	Avito			Meituan		
	AUC	HR	Cost (ms)	AUC	HR	Cost (ms)
$\alpha = 0$	0.7337	0.52	17.5	0.6833	0.57	85.1
$\alpha = 0.01$	0.7336	0.63	17.4	0.6831	0.62	85.3
$\alpha = 0.05$	0.7329	0.71	17.6	0.6828	0.71	85.3
$\alpha = 0.1$	0.7320	0.78	17.5	0.6822	0.84	85.2
$\alpha = 0.3$	0.7108	0.81	17.4	0.6674	0.88	85.4
$\alpha = 0.5$	0.6927	0.88	17.6	0.6425	0.91	85.3
$K = 5$	0.7325	0.63	13.8	-	-	-
$K = 10$	0.7320	0.71	17.5	-	-	-
$K = 20$	0.7318	0.82	25.2	-	-	-
$K = 50$	0.7299	0.93	46.9	0.6827	0.78	49.2
$K = 100$	-	-	-	0.6822	0.84	85.3
$K = 200$	-	-	-	0.6813	0.92	157.3
$K = 300$	-	-	-	0.6799	0.95	223.3

5.4 Hyperparameter Analysis

We analyze the sensitivity of two hyperparameters: α, K. Specifically, α is the weight of the contrastive loss and K is number of permutations selected by FPSM. The result is shown in table 5, showing the same trend on public dataset and industrial dataset and we have the following findings :

i) As α increases within a certain range, the AUC of OCPM maintains a relatively good level, while HR is improved. When α exceeds a certain level, the influence of the contrastive loss on OCPM increases, and the AUC begins to decline, resulting in a decrease in the confidence of the HR indicator.

ii) Changing K mainly affects HR and average cost of the framework. Increasing K within a certain range can quickly improve the HR performance. When it exceeds a certain range, HR increases slowly. Meanwhile, the average cost increases with the increase of K. Therefore, the value of K should be set reasonably to balance the effect and efficiency in practice.

Table 6: The experimental results from Online A/B testing.

Model	CTR	GMV	Cost(ms)	Time-out
Base Ranking Model	0%	0%	**0**	0.0%
PRM	1.21%	0.92%	6.1	0.0%
Beam-Search & Evaluator	3.17%	2.54%	10.5	0.052%
Full Permutation & Evaluator	-	-	52.4	64.39%
PIER	**5.46%**	**5.83%**	11.1	**0.061%**

5.5 Online Results

We compare PIER with other models (e.g. base ranking model, PRM and so on) and all deployed on Meituan food delivery platform through online A/B test. Specifically, we conduct online A/B test with 1% of whole production traffic from April 09, 2022 to April 25, 2022 (one week). As a result, we find that PIER gets CTR and GMV increase by 5.46% and 5.83% respectively. Besides, we focus on time costs, which is an important indicator, determine whether it can be applied to a large scale of industrial scenarios. As show in Tabel 6, Full-Permutation could not deploy because of time-out ratio increase by 64.39%. Compared with beam-search, PIER has improved CTR by 2.29% and GMV by 2.31% while the time-out ratio effect increases little, which is acceptable to the system. Now, PIER has been deployed online and serves the main traffic, and contributes to significant business growth.

6 CONCLUSIONS

This paper presents a novel end-to-end re-ranking framework named PIER which contains two mainly modules named FPSM and OCPM. Inspired by long-time user behavior modeling methods, we apply SimHash in FPSM to select top-K candidates from the full permutation. For better capturing the context information in the permutation, we design a novel omnidirectional attention mechanism in OCPM. Finally, we jointly train these two modules in an end-to-end way by introducing a comparative learning loss to guide the FPSM to generate better permutations. Both offline experiment and online A/B test show that PIER significantly outperformed other existing re-ranking baselines, and we have deployed PIER on Meituan food delivery platform.

REFERENCES

[1] Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 135–144.

[2] Areej Alsini, Du Q Huynh, and Amitava Datta. 2020. Hit ratio: An evaluation metric for hashtag recommendation. *arXiv preprint arXiv:2010.01258* (2020).

[3] Irwan Bello, Sayali Kulkarni, Sagar Jain, Craig Boutilier, Ed Chi, Elad Eban, Xiyang Luo, Alan Mackey, and Ofer Meshi. 2018. Seq2slate: Re-ranking and slate optimization with rnns. *arXiv preprint arXiv:1810.02019* (2018).

[4] Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. 2022. Sampling Is All You Need on Modeling Long-Term User Behaviors for CTR Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2974–2983.

[5] Moses S Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th annual ACM symposium on Theory of computing*. 380–388.

[6] Chi Chen, Hui Chen, Kangzhi Zhao, Junsheng Zhou, Li He, Hongbo Deng, Jian Xu, Bo Zheng, Yong Zhang, and Chunxiao Xing. 2022. EXTR: Click-Through Rate Prediction with Externalities in E-Commerce Sponsored Search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2732–2740.

[7] Dagui Chen, Qi Yan, Chunjie Chen, Zhenzhe Zheng, Yangsu Liu, Zhenjia Ma, Chuan Yu, Jian Xu, and Bo Zheng. 2022. Hierarchically Constrained Adaptive Ad Exposure in Feeds. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3003–3012.

[8] Qiwei Chen, Changhua Pei, Shanshan Lv, Chao Li, Junfeng Ge, and Wenwu Ou. 2021. End-to-End User Behavior Retrieval in Click-Through Rate Prediction Model. *arXiv preprint arXiv:2108.04468* (2021).

[9] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[10] 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.

[11] Yufei Feng, Yu Gong, Fei Sun, Junfeng Ge, and Wenwu Ou. 2021. Revisit recommender system in the permutation prospective. *arXiv preprint arXiv:2102.12057* (2021).

[12] Yufei Feng, Binbin Hu, Yu Gong, Fei Sun, Qingwen Liu, and Wenwu Ou. 2021. GRN: Generative Rerank Network for Context-wise Recommendation. *arXiv preprint arXiv:2104.00860* (2021).

[13] Xudong Gong, Qinlin Feng, Yuan Zhang, Jiangling Qin, Weijie Ding, Biao Li, Peng Jiang, and Kun Gai. 2022. Real-time Short Video Recommendation on Mobile Devices. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3103–3112.

[14] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1725–1731.

[15] Xiang Li, Shuwei Chen, Jian Dong, Jin Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2023. Decision-Making Context Interaction Network for Click-Through Rate Prediction. *arXiv preprint arXiv:2301.12402* (2023).

[16] 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.

[17] Guogang Liao, Xuejian Li, Ze Wang, Fan Yang, Muzhi Guan, Bingqi Zhu, Yongkang Wang, Xingxing Wang, and Dong Wang. 2022. NMA: Neural Multi-slot Auctions with Externalities for Online Advertising. *arXiv preprint arXiv:2205.10018* (2022).

[18] Guogang Liao, Ze Wang, Xiaoxu Wu, Xiaowen Shi, Chuheng Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2022. Cross dqn: Cross deep q network for ads allocation in feed. In *Proceedings of the ACM Web Conference 2022*. 401–409.

[19] Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. 2007. Detecting Near-Duplicates for Web Crawling. In *Proceedings of the 16th International Conference on World Wide Web*. https://doi.org/10.1145/1242572.1242592

[20] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. 2019. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*. 3–11.

[21] D Raj Reddy et al. 1977. Speech understanding systems: A summary of results of the five-year research effort. *Department of Computer Science. Camegie-Mell University, Pittsburgh, PA* 17 (1977), 138.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[23] 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.

[24] Zhe Wang, Liqin Zhao, Biye Jiang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2020. Cold: Towards the next generation of pre-ranking system. *arXiv preprint arXiv:2007.16122* (2020).

[25] Yunjia Xi, Weiwen Liu, Xinyi Dai, Ruiming Tang, Weinan Zhang, Qing Liu, Xiuqiang He, and Yong Yu. 2021. Context-aware reranking with utility maximization for recommendation. *arXiv preprint arXiv:2110.09059* (2021).

[26] 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 *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[27] 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.

[28] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1079–1088.

[29] Tao Zhuang, Wenwu Ou, and Zhirong Wang. 2018. Globally optimized mutual influence aware ranking in e-commerce search. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3725–3731.

A Collaborative Transfer Learning Framework for Cross-domain Recommendation

Wei Zhang
Meituan
Beijing, China
zhangwei180@meituan.com

Pengye Zhang
Meituan
Beijing, China
zhangpengye@meituan.com

Bo Zhang
Meituan
Beijing, China
zhangbo126@meituan.com

Xingxing Wang
Meituan
Beijing, China
wangxingxing04@meituan.com

Dong Wang
Meituan
Beijing, China
wangdong07@meituan.com

ABSTRACT

In the recommendation systems, there are multiple business domains to meet the diverse interests and needs of users, and the click-through rate(CTR) of each domain can be quite different, which leads to the demand for CTR prediction modeling for different business domains. The industry solution is to use domain-specific models or transfer learning techniques for each domain. The disadvantage of the former is that the data from other domains is not utilized by a single domain model, while the latter leverage all the data from different domains, but the fine-tuned model of transfer learning may trap the model in a local optimum of the source domain, making it difficult to fit the target domain. Meanwhile, significant differences in data quantity and feature schemas between different domains, known as domain shift, may lead to negative transfer in the process of transferring. To overcome these challenges, we propose the Collaborative Cross-Domain Transfer Learning Framework (CCTL). CCTL evaluates the information gain of the source domain on the target domain using a symmetric companion network and adjusts the information transfer weight of each source domain sample using the information flow network. This approach enables full utilization of other domain data while avoiding negative migration. Additionally, a representation enhancement network is used as an auxiliary task to preserve domain-specific features. Comprehensive experiments on both public and real-world industrial datasets, CCTL achieved SOTA score on offline metrics. At the same time, the CCTL algorithm has been deployed in Meituan, bringing 4.37% CTR and 5.43% GMV lift, which is significant to the business.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Data mining**.

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.
KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599758

KEYWORDS

Recommendation Systems;Multi-Domain Recommendation;CTR Prediction

ACM Reference Format:
Wei Zhang, Pengye Zhang, Bo Zhang, Xingxing Wang, and Dong Wang. 2023. A Collaborative Transfer Learning Framework for Cross-domain Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599758

1 INTRODUCTION

In online recommendation systems, traditional CTR prediction models[2][23][6][34][24][35] focus on a specific domain, where the CTR model serves only a single business domain after being trained with samples collected from this domain. At large e-commercial companies like Alibaba and Amazon, there are often many business domains that need CTR prediction to enhance user satisfaction and improve business revenue. Since different business domains have overlapping user groups and items, there exist commonalities among these domains [37]. Enabling information sharing is beneficial for learning the CTR model of each domain. Therefore, how to use cross-domain transfer learning to leverage information from the rich domains to help the poor domains has become one of the main research directions in the industry.

The common cross-domain modeling scheme in the industry mainly falls into two paradigms[22][32][5][36][17][14][20]: 1) union and mix up the source and the target samples, and then perform multi-task learning techniques to boost the performance in all domains; 2) pre-train a model using the mixed or the data-abundant source domain data, and then finetune it in the data-deficient target domain to fit the new data distribution. In the first approach, the domain-specific and domain-invariant characteristics are learned through different types of network designs, where the domain indicator is commonly used to identify domains. In the fine-tuning paradigm, the approach mainly argues that the data in the target domain is not enough to make the parameters to be fully trained. Therefore, in such opinions, the pre-train is necessary to train the parameters first, and then make the model converge to the optimum through the target domain data. The two paradigms have been proven to be effective in some scenarios, however, they may still fall short on some occasions, as we will discuss these limitations in the following.

For the multi-task learning solutions, all the source domain data is mixed up with the target ones, and they make the assumption that the model architecture could definitely identify the difference and similarities. However, this may be too idealistic. The user behaviors and item groups may be different as the domain changes, and the amount of data among domains can vary. Therefore, the training process can be easily dominated by data-abundant domains, resulting in insufficient training in sparse domains (i.e., **seesaw effect**[33][3][29]). As a result, such approaches are not friendly to sparser target domains.

For the pre-train and fine-tune solutions, the fine-tuning process is expected to make use of the trained parameters and guide the optimization through target samples. However, the optimal solution trained in the source domain may be a **local minimum** for the target domain(i.e., the non-optimal solution finetune problem[12][8]). The data distribution shift widely exists among domains, e.g., the click-through rate of the same item is different when it is displayed in different creative forms on different domains. When the parameters are well-trained to fit the source distribution, it is hard for the model to jump out and find a new suitable optimum in the target domain. Therefore, it is necessary to evaluate how much beneficial information the source can bring to the target.

In order to solve such problems caused by source domain samples in cross-domain modeling, we propose the CCTL (**C**ollaborative **C**ross-domain **T**ransfer **L**earning framework) algorithm. The CCTL algorithm mainly includes three parts: the Symmetric Companion Network, Information Flow Network, and Representation Enhancement Network. The symmetric companion network trains the mixed model (the target and source samples) and the pure model(only target samples). According to the difference in the effect of the two parts, it is evaluated whether the current source domain samples are helpful to the target domain. The information flow network transfers the sample weight calculated for each source sample and performs semantic alignment between domains. Finally, the representation enhancement network act as an auxiliary task to maintain the domain-specific characteristics in each domain.

The main contributions of this paper are as follows:

- We propose CCTL, a simple but effective cross-domain modeling framework. CCTL can select samples from the source domain that are beneficial to the target domain training and add them to the target domain training, reducing the introduction of invalid samples and noise samples.
- In order to evaluate the gain efficiency of information flow from the source domain to the target domain, we propose an Information Flow Network to evaluate the potential gain of each source domain sample to the target domain.
- We propose the Representation Enhancement Network, through contrastive learning, to allow the id embedding of the source domain and the target domain to accommodate as much different information as possible.
- We evaluate CCTL on the industrial production dataset and deploy it in the display advertising system of Meituan in 2022. The consistent superiority validates the efficacy of CCTL. Up to now, the deployment of CCTL brings 4.37% CTR and 5.43% GMV lift.

2 METHODOLOGY

In this section, we first introduce a general expression formulation for cross-domain CTR prediction. Next, we provide an overview of CCTL, explaining our approach broadly. Afterward, we delve into the details of each component through several subsections.

2.1 Problem Setup

We define the cross-domain CTR prediction problem as using data from one or multiple source domains to enhance the performance of the CTR model on the target domain. Specifically, we annotate an input sample X_i^d and the corresponding label information $y_i^d \in \{0, 1\}$ from domain $d \in \{s, t\}$, where s represents the source domain and t represents the target domain. The goal of cross-domain CTR prediction is to train a model using CTR samples from two business domains, namely the source domain sample set (X_i^s, y_i^s) and the target domain sample set (X_j^t, y_j^t), to predict the click probability in the target domain.

The objective of the cross-domain CTR prediction problem can be formulated as equation (1), shown as follows:

$$\mathcal{J} = minimize\left\{\frac{1}{N_t}\sum_{x_i \in X^t}^{N_t} Loss(y_i^t, f(x_i, \Theta_{model}|X_j^s, X_j^t))\right\} \quad (1)$$

where x_i is a sample in the target sample set X^t, N_t is the number of target samples, y_i^t is the label information of the target domain, f is the model's function form, with input containing samples from both source and target domains (X^s and X^t) and model parameters Θ_{model}, $Loss$ is the loss function over the target sample set X^t with N_t samples, where cross-entropy is commonly adopted in the CTR estimation problem.

2.2 Model Overview (CCTL)

The overall workflow of CCTL is shown in Figure 1. It is composed of three components: the Symmetric Companion Network (SCN), the Information Flow Network (IFN), and the Representation Enhancement Network (REN). The main purpose of SCN is to evaluate the information gain from the source domain samples to the target domain through the dual-tower framework. The role of IFN is to evaluate how much information can be brought by each source sample, and help the information to flow partially through an output weight. And the semantic field of source and target domains is also aligned by IFN. REN is an auxiliary component that tries to preserve the unique information of different domains through a contrastive design. The three components work together to make the best use of cross-domain information and optimize the model training.

2.3 Symmetric Companion Network

Main Description: Train the target domain network and detect negative transfer from the source domain. By designing a symmetrical structure, more useful information from source domain samples can be transferred to the target domain.

The source and target domains have similarities in terms of users and items, but their data distributions differ. There may be information noise from the source domain that can negatively impact

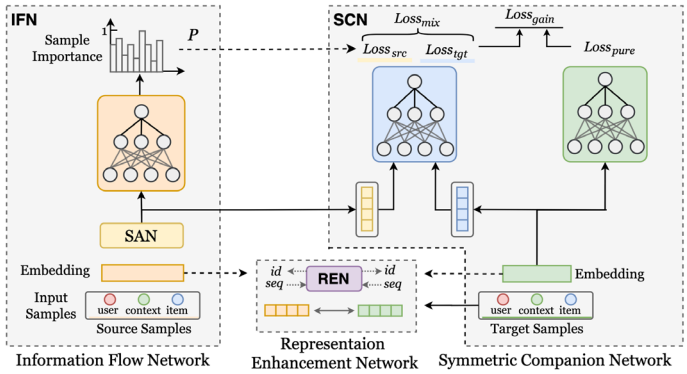


Figure 1: An overview of the CCTL model. a)Symmetric Companion Network(SCN) evaluates the information gain from the source domain samples to the target domain through the dual-tower framework. b)Information Flow Network(IFN) evaluates how much information can be brought by each source sample, and help the information to flow partially through an output weight. c) Representation Enhancement Network(REN) is an auxiliary component that tries to preserve the unique information of different domains through a contrastive design.

the target domain, causing a phenomenon known as the negative transfer. Simple transfer learning techniques like pre-training and fine-tuning or multi-task learning by combining all samples from both domains may not result in significant improvement, as the impact of source domain samples on the target domain is not evaluated. The main purpose of SCN is to identify negative transfer during training, and its principle will be explained in detail. The evaluation of source domain samples is primarily carried out by IFN, which will be covered in Section 2.4.

To accurately evaluate the impact of the source samples on the target domain, the straightforward approach is to compare the performance of a model trained on mixed domain samples to the one trained solely on the target domain. Then, the influence can be calculated as the difference in offline metrics between the two models. The SCN designed based on this concept adopts a dual-tower network architecture as shown in Figure 2. In SCN, one tower, referred to as the mixed tower, receives inputs from both the source and target domains simultaneously, while the other tower, referred to as the pure tower, only receives inputs from the target domain. In the view of the principle of the control variables, the difference between the mixed and pure towers is solely attributed to the impact of the source domain.

2.3.1 Feature Embedding. we encode features in different fields into one-hot encodings and map them to dense, low-dimensional embedding vectors suitable for deep neural networks. The attention mechanism is applied later for the embedded item id and user behavior sequence(item ids). We omit the attention mechanism details because it's not the key design in this paper, and can be replaced by other modeling strategies such as a sequential encoder, etc. Finally, the sample of the target domain is transformed into a vector representation $\mathbf{v}_i^t$, shown as follows:

$$\mathbf{v}_i^t = [\mathbf{e}_1||\mathbf{e}_2||...||\mathbf{e}_{n_t}] \tag{2}$$

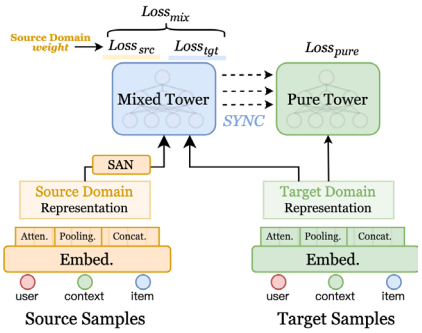


Figure 2: An illustration of the SCN component.

where n_t is the number of categorical features in the target domain, and || denotes the vector concatenation operation. Similarly, we apply the same embedding techniques for the samples in the source domain.

It is worth noting that in real scenarios, there will be differences in the feature scheme of the source and target domains, which will cause the shapes of $\mathbf{v}^t$ and $\mathbf{v}_i^s$ to be inconsistent, thus preventing them from using the same network for training. This paper will use the semantic alignment network(SAN) structure in the information flow network(IFN) to process the Embedding of $\mathbf{v}^s$ into the same shape as $\mathbf{v}^t$. See the IFN chapter in section 2.4 for more details.

2.3.2 Mixed Tower. As shown in Figure 2, the mixed tower will have two parallel data inputs, i.e., reading source domain samples and target domain samples simultaneously for training. In this way,

the loss of the mixed tower is calculated as follows:

$$\begin{aligned} Loss_{mixed} &= Loss_{src} + Loss_{tgt} \\ &= \sum_{x_i \in X^s}^{N_s} \mathbf{p}_i^s L(y_i^s, f(x_i, \Theta_{mixed})) \\ &+ \sum_{x_i \in X^t}^{N_t} L(y_i^t, f(x_i, \Theta_{mixed})) \end{aligned} \tag{3}$$

where x_i is the sample from the source or the target domain, f is short for the neural network operation, L denotes the loss function, and we use cross-entropy in this paper, and Θ_{mixed} is the trainable parameters in the mixed network. $\mathbf{p}_i^s$ is the output of another component indicating the weight of each source sample, and we will discuss it in the next subsection. In this way, we can obtain a mixed tower trained jointly by the source domain and the target domain.

2.3.3 Pure Tower. The pure tower only reads the target samples, which are exactly the same as the target samples in the mixed tower. and the loss in the pure target domain is formulated as follows:

$$Loss_{pure} = \sum_{x_i \in X^t}^{N_t} L(y_i^t, f(x_i, \Theta_{pure})) \tag{4}$$

where $x_i \in X^t$ is the sample only from the target domain, and Θ_{pure} is the parameters in the pure tower. L_{pure} guides the backpropagation of the pure network during training, which is the same as the traditional single-domain training process. In this way, we can obtain a pure tower trained only by the target domain.

2.3.4 Impact of Source Domain. In SCN, the two towers (mixed and pure) have the same shape of network structure, learning rate, etc. The only difference is that the mixed tower additionally reads the samples of the source domain, and these source domain samples affect the network parameters in the mixed tower through the gradient of backpropagation. Therefore, we only need to use the same set of target domain samples to calculate the loss on the two towers, and the difference between the two losses is the impact result of the source domain. Fortunately, during the training process, the loss on the same target domain samples has already been calculated as $Loss_{pure}$ and $Loss_{tgt}$, so the information gain can be defined as follows:

$$\begin{aligned} r = Loss_{gain} &= Loss_{pure} - Loss_{tgt} \\ &= \sum_{x_i \in X^t}^{N_t} [L(y_i^t, f(x_i, \Theta_{pure})) - L(y_i^t, f(x_i, \Theta_{mixed}))] \end{aligned} \tag{5}$$

where N_t is the number of target samples, Θ_{pure} is the training parameter of the pure tower in SCN. The term r represents how much the loss reduces after the parameters are updated by additional source samples. When the information brought by the source domain samples is positive, the mixed tower in the SCN can predict more accurately. In particular, when predicting the same target domain samples, $Loss_{tgt}$ will be smaller than $Loss_{pure}$, and $r > 0$ at this time. Otherwise, the noisy information will result in $r <= 0$, meaning that there is a negative transfer in the cross-domain training.

2.3.5 Parameter Synchronization. To reduce the offset caused by the two-tower learning route, we perform parameter synchronization every k steps, where $k = 1000$ in this paper, that is, synchronize the parameters of the mixed network to the pure network, so that $Loss_{mixed/src/tgt/pure}$ will not introduce too much noise due to network learning offset.

2.4 Information Flow Network

Main Description: Receive the reward and gradient updates from the SCN, and transmit source domain sample weights and representations. Since not all information from source domain samples is useful, so weighted transmission is necessary.

The SCN is capable of assessing negative transfer and detecting it during cross-domain training. As mentioned before, negative transfer in existing methods is often caused by misusing source domain samples, some of which are only partially useful for the target domain. For instance, a user's interests in movies and books may overlap, but not completely match due to differences in presentation format.

The main function of the information flow network(IFN) in figure 3 has three aspects: 1) Evaluate the potential profit coefficient of a single sample in the source domain for the target domain.2) Consistent alignment of the evaluation goal with the goal of maximizing the effect of the target domain model. 3) Effective transfer of the information gained in the source domain to the target domain.

Note that IFN mainly predicts the benefit of a single source domain sample to the target domain, while SCN accurately evaluates the total benefit of the source domain based on the difference in the prediction accuracy of the two towers for the target domain. Conceptually, IFN is responsible for prediction and SCN is responsible for evaluation. In section 2.4.2, the role of SCN in helping IFN training will be explained in detail. In the following section 2.4.1, we will first discuss how to effectively transfer the information obtained in the source domain to the target domain.

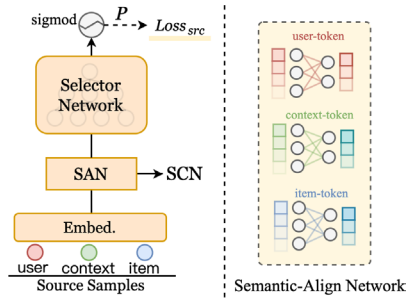


Figure 3: An illustration of the IFN component.

2.4.1 Semantic-Align Network. In this section, we will discuss the challenge of addressing the discrepancy in sample representation caused by the dissimilarities of features in different domains. In cross-domain transfer learning, it is crucial to effectively transfer information from the source domain to the target domain, however,

the mismatch of features can create a significant difference in sample representation. The number of features in various domains can be vastly different, leading to the discrepancy between the source and target domains in the SCN. Hence, a single mixed network cannot be utilized for samples from both domains.

Despite the challenges, transfer learning has proven to be a powerful tool in the fields of computer vision (CV) and natural language processing (NLP). The success of transfer learning is attributed to its ability to preserve cross-domain information through semantic tokens between different domains. In the field of CV, a token can be represented as pixels composing a point, line, or surface, while in NLP, a token can be represented as a word. When facing a new scenario, even though the problem form may change, these basic tokens remain highly consistent with the source task, allowing for knowledge transfer and application.

We consider users, items, and contexts as basic semantic tokens, meaning that users interact with items in specific contexts. This user-item-context relationship can be applied to any recommendation scenario. To handle this, we split the source domain sample vector representation $\mathbf{v}_i^s$ into three separate tokens: $\mathbf{v}_{user}^s$, $\mathbf{v}_{item}^s$, and $\mathbf{v}_{context}^s$. Each token comprises features related to it. For example, the user token representation can be simplified as follows:

$$\mathbf{v}_{user}^s = [\mathbf{e}_{uid}||\mathbf{e}_{male}||\mathbf{e}_{ios}] \tag{6}$$

where $\mathbf{e}_{uid}$, $\mathbf{e}_{male}$, $\mathbf{e}_{ios}$ are the embeddings of user-id, gender, and device type respectively.

In order to achieve the alignment of the source domain to the target domain, we design a special compression network. The input shape of this network is equivalent to the source domain semantic token, and the output shape is equivalent to the corresponding target domain semantic token:

$$\mathbf{v}_{user}^{s'} = MLP(\mathbf{v}_{user}^s) \in \mathbb{R}^{dim_{user}^t} \tag{7}$$

where dim_{user}^t is the dimension of embedded user features in the target domain, and MLP is a set of DNN layers with activation functions. The transformed source sample representations have the same dimensionality as the target domain and are aligned at the semantic token granularity. In this way, the subsequent source domain can use the same network as the target domain for later processing.

But it has to be mentioned that if in an ideal multi-domain environment, the source domain and the target domain use exactly the same features, then this part of the SAN structure can be ignored, that is, the source domain can be directly used without additional semantic token mapping. The original embedding result of the domain is enough.

2.4.2 Selector Network. The selector network is mainly responsible for evaluating the information gain of the source domain samples for the target domain. Regarding the network design, this paper uses a multi-layer MLP network, with the last layer using the sigmoid activation function:

$$\mathbf{p}_i^s = \frac{1}{1 + exp(-W_m h_{m-1} + b_m)} \tag{8}$$

where m is the number of the DNN layers, h_{m-1} is the output of the last DNN layer, and W_m and b_m are the params to be trained.

In the previous SAN structure, the source domain sample $\mathbf{v}_i^s$ has been obtained and fed into the dual-tower to obtain the final information gain evaluation.

Through the weight $\mathbf{p}_i^s$ of IFN, the loss on each source sample is dynamically adjusted. The network parameters in SCN are thus updated through gradient from both the target domain and weighted source domain, so the information from the source domain can be "**partially adapted**" to the target.

The selector network itself has no explicit labels, i.e., there is no label information to indicate whether a source sample is suitable for the target or not. Recall that in the SCN structure, it is already possible to evaluate the gain increase r of a batch of source domain samples for the target domain, but this gain r is a scalar value, which cannot correspond to each source domain sample one-to-one, and thus cannot use the traditional stochastic gradient descent to get updated. But fortunately, there are already mature solutions to such problems in the field of reinforcement learning, which can be updated through reinforcement learning algorithms.

The term r_k can be regarded as the reward, where k is the batch indicator. The accumulated reward can be defined as $r^{accu} = r_k + \gamma * r_{k+1} + ... + \gamma^{n-k} * r_k$, where r_n is the reward of the last batch and γ is the weight factor and is set to 0.80 uniformly.

Finally, we adopt REINFORCE[25] algorithm to update the IFN component, and the parameters are updated as follows:

$$\Theta_{ifn} \leftarrow \Theta_{ifn} + \alpha \frac{1}{N} \sum_{i=1}^{N_s} \nabla \Theta_{ifn} log(\mathbf{p}_i^s) r^{accu} \tag{9}$$

where Θ_{ifn} is the parameter of the IFN, $\mathbf{p}_i^s$ is the output weight for the ith source sample, r^{accu} is the accumulated reward, and α controls how much influence the source samples should have on the target domain. Like this, the r^{accu} is regarded as the indirect loss part. In order to make the gradient update of the selector more stable, we accumulate the reward r^{accu} and perform the gradient return of the IFN every 1000 steps, so as to avoid excessive fluctuation of the batch effect and stabilize the parameter update process.

With the symmetric design of SCN, we are able to evaluate the information gain of source domain samples. When the information gain brought by the sample is positive, the mixed tower in SCN can predict the target more accurately, so that $Loss_{tgt}$ decreases, and the reward is positive, that is, the reward encourages IFN to increase this weight on such source samples. Similarly, when the sample has a negative impact on the target domain, $Loss_{tgt} > Loss_{pure}$, the reward is negative, and the selection strategy of IFN will be corrected when the gradient is updated.

2.5 Representation Enhancement Network

Main Description: Use contrastive learning strategies to distinguish the similarity and differences between samples from different domains, so as to reduce the harmful impact of domain-specific knowledge in the source domain on the target domain.

In the paradigm of transfer learning, we have found through practice that a single embedding representation cannot fully cover cross-domain meaning when the differences between domains are large. Because each domain has its own unique knowledge, even for the same item in different domains. For example, in the search and feed scenarios, the search domain has query information, while the

feed domain does not. Therefore, we attempt to retain some domain-specific knowledge during the transfer process to accommodate situations where the source and target domains have significant differences. Specifically, in our scenario, our approach is mainly based on the following two ideas:

Maximizing the similarity between sequence embedding. For the same user, although there are differences in the behavior sequences in different domains, the user's interest remains stable and is expressed through different heterogeneous sequence representations in different domains. Therefore, we believe that the user representations between different domains should be kept as consistent as possible. User representations mainly consist of sequence features, so we believe that it is necessary to maximize the similarity between sequence embeddings.

Minimizing the similarity between ID embedding. Based on maximizing user similarity, we also want to retain the unique information of each domain, expecting to provide differentiated information from the source domain to the target domain. ID Embedding is a microscopic representation of a domain, so we try to control the differentiated representation of the same ID between different domains, thereby retaining more additional information.

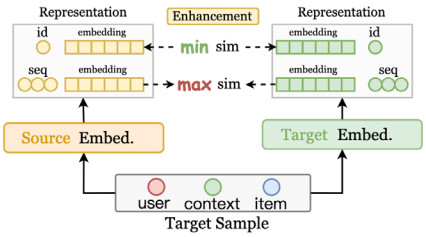


Figure 4: An illustration of the REN component.

As illustrated in Figure 4, we proposed the Representation Enhancement Network (REN) to address the above ideas. These two ideas act as constraints, somewhat similar to minimax game design, can be formalized as follows:

$$\begin{aligned} &min\left[sim(\mathbf{v}^s_{id}, \mathbf{v}^t_{id})\right] \\ &max\left[sim(\mathbf{v}^s_{seq}, \mathbf{v}^t_{seq})\right] \end{aligned} \tag{10}$$

where $\mathbf{v}^s_{id}$ is the id embedding after source domain embedding lookup, and $\mathbf{v}^t_{id}$ is the target one. $\mathbf{v}^s_{seq}$ is for the sequence embedding. In particular, for the sequence embedding, we use the average pooling for the user behavior id sequence. sim is the similarity metric, where we use the normalized cosine similarity for simplicity as is shown:

$$sim(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{||\mathbf{v}_i||_2 \cdot ||\mathbf{v}_j||_2} \tag{11}$$

where $||\cdot||_2$ is the length in Euclidean space for the vector. The term evaluates how similar are the two vectors. The loss formula can be expressed as follows through an equivalent transformation while optimizing both max and min at the same time:

$$Loss_{ren} = sim(\mathbf{v}^s_{id}, \mathbf{v}^t_{id}) - sim(\mathbf{v}^s_{seq}, \mathbf{v}^t_{seq}) \tag{12}$$

The auxiliary loss of REN is added to the SCN and acts as an assistant to help the mixed tower better recognize the domain-specific and invariant features. The main idea of the dual embedding design is to maintain the differences between domains, and also, the separation of embedding also avoids possible conflicts of different encoding among domains.

3 EXPERIMENTS

In this section, extensive offline and online experiments are performed on both the large-scale recommender system in Meituan and public benchmark datasets to answer the following research questions:

RQ1 Does our proposed method outperform the baseline methods?

RQ2 How does each part of our CCTL model work?

RQ3 How does the model perform when deployed online?

Before presenting the evaluation results, we first introduce the experimental setup, including datasets, baselines, metrics, and parameter settings.

3.1 Experimental Setup

3.1.1 ***Datasets.*** We adopt public datasets and industrial datasets to comprehensively compare CCTL models and baseline models. The statistics of the datasets are shown in Table 1.

Amazon dataset: This dataset has been widely used to evaluate the performance of collaborative filtering approaches. We use the two largest categories, Books and Movies&TV, to form the cross-domain datasets. We convert the ratings of 4-5 as positive samples. The dataset we used contains 979,151 user-book ratings and 432,286 user-movie ratings. There are 61,437 shared users, 835,005 books, and 368,953 movies. We aim to improve the movie-watching domain recommendation(as the target domain led from knowledge). The statistics are summarized in the table, and hence we hope to improve the target domain performance by transferring knowledge from source domains.

Taobao dataset: This dataset was first released by Alibaba-Taobao and is widely used as a common benchmark in CTR prediction tasks. It is a user behavior log for Taobao mobile application, including click, purchase, add-to-cart, and favorite behaviors. This dataset involves 987,994 users and 4,162,024 items. We rank according to the number of samples under each category, divide the top 70% categories as the source domain, and the remaining categories as the target domain.

Industrial dataset: This dataset is an industrial dataset collected by Meituan App which is one of the top-tier mobile Apps in our country. It is much larger than the Amazon and Taobao public datasets. In the source domain, the data contains the sample information of the user on the list page, involving 230M users and 1.1M items, where M is short for 10^6. The target domain mainly includes 183M users and 0.7M items. There is an intersection between the two domains on users and items, which are 67.83% and 70.26% respectively.

3.1.2 ***Baselines.*** We compare both single-domain and cross-domain methods. Existing cross-domain methods are mainly proposed

Table 1: Statistics of public and industrial datasets.

Datasets	Source Domain			Target Domain		
	Users	Items	Samples	Users	Items	Samples
Amazon	61,437	835,005	979,151	61,437	368,953	432,286
Taobao	690,006	69,386,796	70,000,525	296,716	29,436,098	30,150,807
Meituan	230M	1.1M	1200M	183M	0.7M	250M

for cross-domain recommendation and we extend them for cross-domain CTR prediction when necessary (e.g., to include attribute features rather than only IDs and to change the loss function).

A. Single-Domain Methods.

- LR. Logistic Regression[20]. It is a generalized linear model.
- DNN. Deep Neural Network[1]. It contains an embedding layer, a fully connected layer(FC), and an output layer.
- DeepFM. DeepFM model[6]. It combines factorization machine(FM, the wide part) and DNN (the deep part), thus modeling multi-order feature interactions.
- DIN. Deep Interest Network model[35]. It models dynamic user interest based on historical behavior sequences.
- AFM. Attentional Factorization Machine[26]. It learns the weight of feature interactions through an attention design.

B. Cross-Domain Methods.

- Finetune. Finetune Network[31]. A typical transfer learning method in the field of CTR prediction trains on the source domain and finetunes the networks in the target domain.
- DANN. Domain-Adversarial Neural Network[5]. It maps the source and the target domain to the same space through the adversarial network and a gradient flip design, so that the domain-invariant information can be learned.
- CoNet. Collaborative cross Network[9]. The cross-connection units are added on MLP++ to enable the knowledge to transfer among domains.
- MiNet. Mixed Interest Network[17]. It jointly models long and short-term interest in the source domain with a short-term interest in the target domain.
- STAR. An industrial framework using star topology design[22], which models different domains of CTR as multi-task learning.
- CCTL. CCTL is the proposed approach in this paper.

3.1.3 ***Parameter Settings.*** We set the dimension of the embedding vectors for each feature as embedding dim = 8. We set the number of fully connected layers in neural network-based models as L=4, where the industrial dataset is [1024, 512, 128, 1] and the public dataset is [256, 128, 32, 1]. For the industrial dataset, the batch size is set to 8192, while set to 128 for the Amazon and Taobao datasets. All the methods are implemented in Tensorflow and we use Adam[4] as the optimization algorithm uniformly. Each method has been run 3 times and the reported are the average results.

3.1.4 ***Metrics.*** We evaluate the CTR prediction performance with two widely used metrics. The first one is the area under the ROC curve (AUC) which reflects the pairwise ranking performance between click and non-click samples. The other metric is log loss, which is to measure the overall likelihood of the test data and has been widely used for classification tasks. At the same time, We deploy models online in real industrial systems. Our online evaluation metric is the real CTR($CTR = \frac{\#click}{\#pages}$) which is defined as the number of clicks over the number of impressions and GMV($GMV = 1000 * \frac{\#pay\ amount}{\#pages}$) which represents the revenue amount per thousand impressions, as an evaluation metric in our A/B tests.

3.2 RQ1: Does our CCTL model outperform the baseline model?

We evaluate the performance of CCTL and baseline models on three datasets. The performances of single-domain and multi-domain are compared simultaneously. The single-domain model is introduced for two purposes: 1) to explain the limitations of the current single-domain, and show the non-optimal fintune problem; 2) to make an intuitive comparison with the multi-domain model. From Table 2, we have the following observations:

In single-domain, the more complex the model, the better the effect, but its marginal returns show diminishing status. At the same time, we noticed that as the computational complexity of the model increases, the benefit of AUC increases to a marginally diminishing state, especially when both DIN and AFM adopt variants based on the attention structure, and the performance improvement is limited. Our analysis may be based on the current model complexity, which is gradually approaching the amount of information contained in the sample, resulting in limited effect improvement.

For cross-domain methods, the overall effect is better than single-domain ones, and the benefits of cross-domain information flow are better than the increase in the complexity of a single model. However, the finetune model may not be better than the traditional single model (for example, on the Meituan dataset, its effect basically slightly worse than DIN and AFM), we think the reason is the non-optimal solution finetune problem mentioned above. CoNet introduces cross-connection units to enable dual knowledge transfer across domains. DANN maps the source and target domains to the same space through an adversarial network and gradient flip design so that the samples in the source domain similar to the target can be integrated. However, these two methods also introduce higher complexity and random noise, because all samples are used indiscriminately. The MiNet method starts from the perspective of information flow from the source domain to the target domain, tries to split information from a long-term and short-term perspective, and transfers interest information from the source domain through a specific network structure. But the domain-specific info is not considered, and it lacks the ability to filter and constrain the information on the source domain. The STAR method adopts separate MLPs to tackle multiple domains, while we see the negative transfer in Meituan and Taobao datasets, which is probably caused by

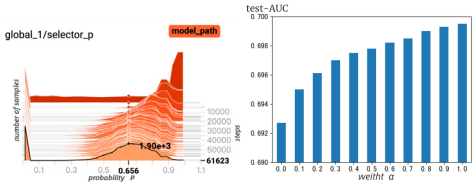
Table 2: Experimental results on different methods.

Domains	Model	Meituan		Amazon		Taobao	
		AUC	Loss	AUC	Loss	AUC	Loss
Single Domain	LR	0.6805	0.5589	0.7421	0.4677	0.7907	0.3894
	DNN	0.6876	0.5537	0.7456	0.4448	0.7953	0.3741
	DeepFM	0.6893	0.5497	0.7473	0.4432	0.7962	0.3724
	DIN	0.6932	0.5468	0.7480	0.4454	0.8095	0.3692
	AFM	0.6923	0.5457	0.7460	0.4492	0.7990	0.3854
Cross Domain	Finetune	0.6921	0.5452	0.7564	0.4381	0.8280	0.3585
	DANN	0.6946	0.5404	0.7610	0.4331	0.8420	0.3601
	CoNet	0.6942	0.5416	0.7621	0.4332	0.8418	0.3666
	MiNet	0.6953	0.5377	0.7665	0.4305	0.8447	0.3640
	STAR	0.6945	0.5391	0.7672	0.4298	0.8440	0.3643
	CCTL	**0.7001**	**0.5288**	**0.7790**	**0.4230**	**0.8521**	**0.3578**

domain shifts. In contrast, our proposed CCTL use an information flow network to provide filtered and beneficial samples to the target domain. Apart from the alignment for semantic tokens, we further use a representation enhancement network to maintain the domain differences. Therefore, the performance in the target domain can be improved compared with baselines.

3.3 RQ2: How does each part of our CCTL model work?

A.Effect of the Selector Network. We evaluate the role of the Selector Network in IFN in two ways on the Meituan dataset. Firstly, we track the output sample weight P of the selector during the training process, then plot the distribution histogram in tensorboard. The horizontal axis is the selector output probability, the vertical axis is the number of samples, and the right side is marked as the steps of model training. As can be seen from Figure 5(a), the distribution changes when training, and finally most of the samples form a similar Gaussian distribution with an average of P=0.656. And some of the source domain samples are not selected (P=0 on the left). This shows that the probability of the selector network is gradually learned as the model is trained.



(a) Selector P's trend in training. (b) Sensitivity for selector weight.

Figure 5: The effect evaluation of the Selector Network.

Secondly, we adjust the weight parameter α in eq (9) of the selector network, where $\alpha \in [0, 1]$. By adjusting α the importance of the selector network can be changed. When α is 0, it degenerates into an ordinary multi-domain sample task. As can be seen from Figure 5(b), when the size of α is changed from small to large, the effect test-AUC in the target domain shows a growing trend, and as α continues to approach 1, the effect growth slows down. This shows the selector has a positive influence on the model.

B.Effect of different modules. In the proposed CCTL framework of this paper, IFN and REN can be viewed as auxiliary tasks to assist the main module SCN in training the target domain. Therefore, in order to verify their impact, we conducted ablation experiments on the IFN and REN modules respectively. The results of the ablation experiments are shown in table 3.

Table 3: The ablation study for the components in CCTL.

Ablations	AUC	LogLoss
CCTL w/o IFN	0.8385	0.3647
CCTL w/o REN	0.8493	0.3602
Full CCTL	0.8521	0.3578

It can be seen from table 3 that the effect of removing IFN is greater than that of releasing REN. Removing IFN means using all source domain samples for training. In this case, the model's performance is only slightly better than that of the Finetune, which indicates that using source domain samples indiscriminately is not a wise choice. When the REN module is removed, the representations of different domains may be close to each other, resulting in a lack of discrimination between different domains. Through the above ablation experiments, the necessity of IFN and REN has been demonstrated, and it is shown that IFN contributes more benefits, indicating that the selection and weighting of source domain samples are very important.

3.4 RQ3: How does the model perform when deployed online?

We deployed CCTL in Meituan App, where the ad serving system architecture is shown in Figure 6. It is worth noting that although we involved multiple network structures during offline training when serving online, we only need to use the pure tower in SCN(note: the parameters in the mixed tower have been synchronized to the pure tower during training), i.e., just export the network of target domain in the online model service. In this way, we ensure that its model complexity is equivalent to that of the online model, without adding additional calculations to the online service.

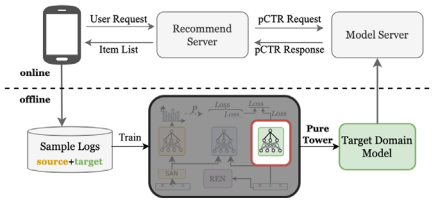


Figure 6: Architecture of the online deployment with CCTL.

We conducted online experiments in an A/B test framework over three months during Sep.2022 - Dec. 2022, where the base serving model is a variant of DIN according to our business characteristics. Our online evaluation metric is the real CTR($CTR = \frac{\#click}{\#pages}$) which is defined as the number of clicks over the number of impressions and GMV($GMV = 1000 * \frac{\#pay\ amount}{\#pages}$) which represents the revenue amount per thousand impressions. A larger online CTR indicates the enhanced effectiveness of a CTR prediction model. The online A/B test shows that CCTL leads to an increase of online CTR of 4.37% and GMV of 5.43% compared with the base model. This result demonstrates the effectiveness of CCTL in practical CTR prediction tasks. After the A/B test, CCTL serves the main ad traffic in Meituan App.

4 RELATED WORKS

Our paper belongs to the field of cross-domain CTR prediction and has close ties to single-domain CTR prediction. In this section, we provide a brief overview.

Cross-Domain CTR Prediction: Cross-domain learning refers to a technique that enhances the performance of a target domain by incorporating knowledge from source domains. This is accomplished through various algorithms, including transfer learning[16], multi-task learning[33], and multi-view learning[30]. These methods aim to transfer knowledge in a deep manner, allowing multiple base networks to benefit from each other, and provide more flexible representation transfer options. Cross-domain learning is useful for various applications, including computer vision, natural language processing, and recommendation systems, where data and tasks across domains are related. In the CTR prediction field, STAR[22] tries to mix multiple sources of data for training a unified model. CoNet[9] aims to make cross-domain information flow more efficiently by enabling dual knowledge transfer through cross-mapping connections between hidden layers in two base networks. MiNet[17] further divides multiple sources into long-term and short-term interests and directly models cross-domain interests. Under the paradigm of union training, the DASL[14] model includes both the dual embedding and dual attention components to jointly capture cross-domain information. Meanwhile, AFT[7] learns feature translations between different domains within a generative adversarial network framework.

Data sparsity[15] is also a classic problem that Cross-Domain Recommendation seeks to solve. This may be due to the sparsity of the target domain data itself, or to local sparsity caused by the long-tail effect and cold-start effect. Commonly used methods for such problems involve using data from the source domain to address the sparsity of target domain data. For example, ProtoCF[21] efficiently transfers knowledge from arbitrary base recommenders by extracting meta-knowledge to construct discriminative prototypes for items with very few interactions. PTUPCDR[38] proposes that a meta-network fed with users' characteristic embeddings is learned to generate personalized bridge functions for achieving personalized transfer of user preferences. TLCIOM[11] uses domain-invariant components shared between dense and sparse domains to guide neural collaborative filtering to achieve information flow from one dense domain to multiple sparse domains. CCDR[28] and SSCDR[10] address the problem of the cold-start domain lacking sufficient user behaviors through techniques such as contrastive learning and semi-supervised learning.

All these approaches tried different ways of separating domain info, or integrating the source samples through different model architectures, while few of them focus on the information weight of the source sample, which is the key point this paper addresses.

Single-Domain CTR Prediction: CTR prediction models have shifted from traditional shallow approaches to modern deep approaches. These deep models commonly use embedding and MLP. Wide&Deep[1] and DeepFM[6] improve expression power by combining low- and high-order features. User behavior is transformed into low-dimensional vectors through embedding techniques. DIN[35] uses attention to capture the diversity of user interest in historical behavior. DIEN[34] adds an auxiliary loss and combines attention with GRU to model evolving user interest. MIND[13] and DMIN[27] argue for multiple representations to capture complex patterns in user and item data, using capsule networks and dynamic routing. The Transformer is used for feature aggregation inspired by the success of self-attention in sequence-to-sequence learning. MIMN[18] employs a memory-based architecture to aggregate features and addresses long-term user interest modeling. SIM[19] extracts user interest using cascaded search units, improving scalability and accuracy in modeling lifelong behavior data. The framework of this paper focuses on the information flow design, so these single-domain techniques can be merged with ours easily in the SCN design, which may further boost the model performance.

5 CONCLUSION

In this paper, we propose a collaborative cross-domain transfer learning framework, which enables the reuse of beneficial source domain samples. In the framework, the information flow network(IFN) can evaluate the importance of the source domain samples, in which the SAN helps align meaningful semantic tokens among domains. The symmetric companion network(SCN) is designed with a symmetric structure to approximately fit the information gain of the samples in the source domain to the target domain. The representation enhancement network(REN) maintains domain-specific characteristics through a contrastive learning mechanism. In this way, only beneficial information in the source domain is transferred to the target, which reduces the seesaw effect and the negative transfer. Then we validate our method on both public and real-world industrial datasets by performance comparison and ablation study, which proved its effectiveness. Finally, we deployed the CCTL model on Meituan App. The CCTL model has brought significant business improvement and served as the mainstream traffic.

REFERENCES

[1] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[2] 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.

[3] Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796* (2020).

[4] 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).

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

[6] 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).

[7] Xiaobo Hao, Yudan Liu, Ruobing Xie, Kaikai Ge, Linyao Tang, Xu Zhang, and Leyu Lin. 2021. Adversarial feature translation for multi-domain recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2964–2973.

[8] Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 1121–1133.

[9] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 667–676.

[10] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1563–1572.

[11] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. 2020. Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1081–1090.

[12] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054* (2022).

[13] 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 *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.

[14] Pan Li, Zhichao Jiang, Maofei Que, Yao Hu, and Alexander Tuzhilin. 2021. Dual Attentive Sequential Learning for Cross-Domain Click-Through Rate Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3172–3180.

[15] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach.. In *IJCAI*, Vol. 17. 2464–2470.

[16] Shuteng Niu, Yongxin Liu, Jian Wang, and Houbing Song. 2020. A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence* 1, 2 (2020), 151–166.

[17] 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 *Proceedings of the 29th ACM international conference on information & knowledge management*. 2669–2676.

[18] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[19] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[20] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. 521–530.

[21] Aravind Sankar, Junting Wang, Adit Krishnan, and Hari Sundaram. 2021. Protocf: Prototypical collaborative filtering for few-shot recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 166–175.

[22] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4104–4113.

[23] Huinan Sun, Guangliang Yu, Pengye Zhang, Bo Zhang, Xingxing Wang, and Dong Wang. 2022. Graph Based Long-Term And Short-Term Interest Model for Click-Through Rate Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1818–1826.

[24] 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.

[25] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[26] 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).

[27] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2265–2268.

[28] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. 2022. Contrastive cross-domain recommendation in matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4226–4236.

[29] Yufeng Xie, Mingchu Li, Kun Lu, Syed Bilal Hussain Shah, and Xiao Zheng. 2022. Multi-task Learning Model based on Multiple Characteristics and Multiple Interests for CTR prediction. In *2022 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 1–7.

[30] Xiaoqiang Yan, Shizhe Hu, Yiqiao Mao, Yangdong Ye, and Hui Yu. 2021. Deep multi-view learning methods: A review. *Neurocomputing* 448 (2021), 106–129.

[31] Xiangli Yang, Qing Liu, Rong Su, Ruiming Tang, Zhirong Liu, Xiuqiang He, and Jianxi Yang. 2022. Click-through rate prediction using transfer learning with fine-tuned parameters. *Information Sciences* 612 (2022), 188–200.

[32] Xuanhua Yang, Xiaoyu Peng, Penghui Wei, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. AdaSparse: Learning Adaptively Sparse Structures for Multi-Domain Click-Through Rate Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4635–4639.

[33] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[34] 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 *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[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 *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[36] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. Dtcdr: A framework for dual-target cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1533–1542.

[37] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696* (2021).

[38] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1507–1515.

Modeling Dual Period-Varying Preferences for Takeaway Recommendation

Yuting Zhang
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
University of Chinese Academy of Sciences
Beijing, China
zhangyuting21s@ict.ac.cn

Yiqing Wu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
University of Chinese Academy of Sciences
Beijing, China
wuyiqing20s@ict.ac.cn

Ran Le
Meituan
Beijing, China
leran@meituan.com

Yongchun Zhu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
University of Chinese Academy of Sciences
Beijing, China
zhuyongchun18s@ict.ac.cn

Fuzhen Zhuang*
Institute of Artificial Intelligence, Beihang University
Beijing, China
SKLSDE, School of Computer Science, Beihang University
Beijing, China
zhuangfuzhen@buaa.edu.cn

Ruidong Han
Meituan
Beijing, China
hanruidong@meituan.com

Xiang Li
Wei Lin
Unaffiliated
Beijing, China
leo.lx007@qq.com
lwsaviola@163.com

Zhulin An*
Yongjun Xu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
anzhulin@ict.ac.cn
xyj@ict.ac.cn

ABSTRACT

Takeaway recommender systems, which aim to accurately provide stores that offer foods meeting users' interests, have served billions of users in our daily life. Different from traditional recommendation, takeaway recommendation faces two main challenges: (1) **Dual Interaction-Aware Preference Modeling**. Traditional recommendation commonly focuses on users' single preferences for items while takeaway recommendation needs to comprehensively consider users' dual preferences for stores and foods. (2) **Period-Varying Preference Modeling**. Conventional recommendation generally models continuous changes in users' preferences from a session-level or day-level perspective. However, in practical takeaway systems, users' preferences vary significantly during the morning, noon, night, and late night periods of the day. To address these challenges, we propose a **D**ual **P**eriod-**V**arying **P**reference modeling (DPVP) for takeaway recommendation. Specifically, we design a dual interaction-aware module, aiming to capture users' dual preferences based on their interactions with stores and foods. Moreover, to model various preferences in different time periods of the day, we propose a time-based decomposition module as well as a time-aware gating mechanism. Extensive offline and online experiments demonstrate that our model outperforms state-of-the-art methods on real-world datasets and it is capable of modeling the dual period-varying preferences. Moreover, our model has been deployed online on Meituan Takeaway platform, leading to an average improvement in GMV (Gross Merchandise Value) of 0.70%.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Takeaway Recommendation; Graph Neural Network; Dual Period-Varying Preferences

*Corresponding authors.

This work is licensed under a Creative Commons Attribution International 4.0 License.

KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599866

ACM Reference Format:
Yuting Zhang, Yiqing Wu, Ran Le, Yongchun Zhu, Fuzhen Zhuang, Ruidong Han, Xiang Li, Wei Lin, Zhulin An, and Yongjun Xu. 2023. Modeling Dual Period-Varying Preferences for Takeaway Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599866

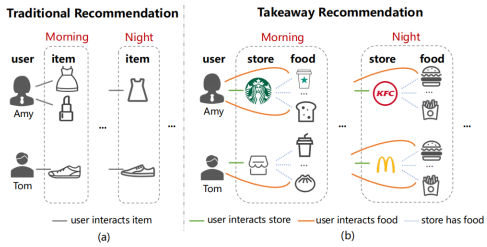


Figure 1: Illustration of differences between traditional recommendation and takeaway recommendation.

1 INTRODUCTION

With convenient online ordering and offline delivery services, takeaway platforms (e.g., Meituan[1]/Uber Eats[2]) are playing an increasingly important role in our daily life. The total number of users in takeaway platforms has reached approximately 2.5 billion in 2022 [1]. Aiming to accurately provide stores with foods that meet users' interests from a large store candidate pool, takeaway recommender systems generally capture users' preferences for foods or stores through their historical interaction behaviors with stores or foods. Compared with traditional recommendation, takeaway recommendation appears greatly more difficult due to the complicated relationships among users, stores and foods. Thus, takeaway recommender systems have attracted great attention from the research community and industry field [16, 23, 31, 38, 40].

Generally, traditional recommendation [3, 10, 27, 41, 44] models users' period-agnostic preferences from the single historical interaction data between users and items as Figure 1(a) shown. However, the following two notable characteristics of takeaway systems make it challenging to directly apply traditional methods for takeaway recommendation.

(1) **Dual Interaction-Aware Preference**. As shown in Figure 1, traditional recommendation, such as e-commerce recommendation, usually model user-item simple binary interaction for users' preferences for items, while the interaction between users, stores, and foods in takeaway recommendation appears more complex including three types of interaction relations. In particular, users' dual interactions with stores and foods imply users' preferences at different levels. Additionally, with the auxiliary relations between stores and foods, there is a more complex interplay between the users' dual preferences. For example, Tom might be fond of food baozi instead of caring about the store, while Amy might pay more attention to the brand of the dine-in store. Also, some people may weigh both preferences for the store and its food information, such as making a balance between high store ratings and low food prices.

(2) **Period-Varying Preference**. As depicted in Figure 1, in traditional recommendation, there is no obvious feature of user preferences changing with the time period of the day (i.e., period-agnostic preference modeling), while in takeaway recommendation, user preferences, especially for food, change significantly in different periods of a day. Visually, we obtain a total of 100,000 click interactions between users and five major food categories in a real takeaway dataset, and further present the proportion distribution of clicks on different categories of food in each period (i.e., Morning/Noon/Night/Late Night) according to the click occurring time in Figure 2. Note that as a traditional Chinese food, baozi generally serves as breakfast in the Morning. Apparently, in the Morning period, users are greatly more likely to interact with baozi than barbecue, while the opposite is true during Late Night. Particularly, user interactions in different periods also imply similar preferences. For example, when a user prefers spicy food, he/she is inclined to interact with spicy-flavored food over different periods.

Existing takeaway recommendation methods commonly leverage the coarse-grained single interaction data between the user and store or food and may further utilize the time as auxiliary information [16, 23, 38, 40]. With feature-enriched user and store representations as input, [40] utilized the classical two-tower model to make store recommendations based on the historical interaction data between users and stores. Further, [16] adopted an attention-based fusion mechanism at different times for users' single-level time-enhanced preference for foods, which is incapable of modeling users' period-varying preferences within a day. In brief, the existing methods have not systematically modeled users' dual period-varying preferences for food and store.

To address the above issues, we propose a novel model, named DPVP, for takeaway recommendation. Specifically, we design a dual interaction-aware module, aiming to capture users' preferences based on the ternary interaction data shown in Figure 1(b). Furthermore, We propose a time-based decomposition module to capture users' dual-period-specific preferences in the period hierarchy and simultaneously learn users' period-shared preferences. Then, a time-aware gate network is utilized to fuse the user's dual period-specific and period-shared preferences to obtain the user's dual period-varying preferences. Eventually, the model make a prediction based on the balance between the users' dual period-varying preferences for stores and foods. To sum up, our contributions are listed below:

- We highlight the challenges of dual interaction-aware and time-varying preference modeling in takeaway recommendation, which is more complicated than traditional recommendation modeling, and further design a novel model to address these challenges. To the best of our knowledge, we are the first to systematically model dual time-varying preferences for takeaway systems.
- We propose a dual interaction-aware module for capturing users' dual preferences. Besides, we design a time-based decomposition to model users' period-specific preferences in the period hierarchy, and further design a time-aware gate to adaptively model users' period-varying preferences.
- Both offline and online experiments were conducted to show the effectiveness of our model and the great ability of the model to capture the dual period-varying preferences for takeaway recommendation.

[1]https://about.meituan.com/en
[2]https://www.ubereats.com

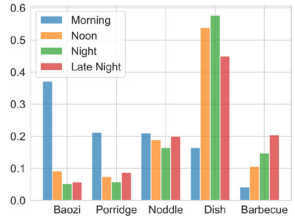


Figure 2: Proportion distribution of clicks between users and five major food categories in four different time periods. Note that the sum of the proportions of the five food categories in each time period is equal to 1.

2 RELATED WORK

2.1 Takeaway Recommendation

In the takeaway recommender system, multiple interactions of user, store, and food lead to complex relationships among the three, which is different from the simple binary interaction relationship usually modeled in traditional recommender systems [3, 10, 27, 41, 44]. For example, on the basis of the assumption that users with similar behaviors have similar preferences on stores, traditional Collaborative Filtering (CF) [6] has been extensively applied in modern takeaway recommender systems. Since CF-based methods [10, 14, 25] suffer from data sparsity and cold start issues, many works tried to utilize additional information such as time information [16, 23, 31, 38], spatial information[16, 23, 38] and attributes [16, 31, 40] to enhance recommendation accuracy. For example, DAT [40] enriched the user and store representations based on the corresponding user features and store attribute information for the recommendation. Moreover, TRISAN [23] adopted an attention-based fusion mechanism to capture the rich interplay between spatial information and time information to improve recommendation performance. Further, considering the importance of time information in takeaway systems, some methods attempted to not simply utilize the time information as the auxiliary input of the designed model. For example, AutoIntent [15] treated the interaction time factors as vertex nodes in graphs and then extracted the high-order relations of three groups of hypergraphs among user preferences for the food category factor, time factor, and spatial factor respectively for the recommendation. However, existing takeaway recommender systems rarely involve the inherent ternary complex user-store-food relationships and period-varying characteristics in the systems.

2.2 Graph-based Recommendation

With the rapid development and huge success of Graph Neural Network (GNN) in various machine learning areas [8, 36, 42, 45], great efforts have been devoted to applying it to the recommendation [5, 15, 22, 35, 39]. Most data in recommender systems essentially implies graph structure [2, 39]. For example, the interaction data can naturally form a bipartite graph between user and item nodes, with observed interactions denoted by links between corresponding nodes. Given the graph data, GNN mainly focuses on iteratively aggregating information from neighbors and integrating the aggregated information with the current central node representation during the propagation process. For example, GC-MC [2] aggregated the information from neighbors based on the types of ratings. Standard GCN [13] factorized user-item rating matrices into user and item embedding matrices for recommendation. NGCF [32] advanced GCN by additionally encoding the interactions via an element-wise multiplication. Due to the discovery of the non-necessity of nonlinear activation and feature transformation in GCN [33], SGCN [33] and LightGCN[9] were proposed for recommendation by removing these two parts. Moreover, owning to the powerful representation ability of GNN, a great number of GNN-based methods [34, 37, 39, 43] have been deployed in the industry to produce high-quality recommendation results. For example, Pinsage [39] combined efficient random walks and graph convolutions to generate embeddings of items that incorporate both graph structure as well as item feature information, which was the first work applying GCN in industrial recommender systems. DANSER [34] took advantage of the interaction between users and articles and social relations for a real-world article recommender system, WeChat Top Story. Different from previous works, our method retains the traditional user–item structure, extending it to dual period-varying tripartite graphs among users, stores and foods, aiming at modeling the users' dual interaction-aware and period-varying preferences for takeaway recommendation.

3 PRELIMINARIES

In practical takeaway systems, an interaction commonly forms "who interacts (e.g., clicks) which food in which store at what time", which involves a user, a store, several foods as well as a time period. To comprehensively model the interactions, we define the *period-varying ternary interaction data* in this work as follows:

Definition 1.*Period-Varying Ternary Interaction Data.* Given $\mathcal{U}, \mathcal{S}, \mathcal{O}, \mathcal{M}$ denoting the universal sets of users, stores, foods, and time periods, each record x in the interaction dataset $\mathcal{X}$ can be formulated as $x = (u, s, O, m)$, which represents that user $u \in \mathcal{U}$ interacts N foods $O = \{o_1, o_2, \cdots, o_N\}$ (where $o_j \in \mathcal{O}, \forall j = 1, 2, ..., N$) in store $s \in \mathcal{S}$ at time period $m \in \mathcal{M}$.

Specifically, we divide a day into 4 time periods: Morning, Noon, Night and Late Night, i.e., $M = 4$ with M denoting the number of time periods $|\mathcal{M}|$. Example interaction records can be found in Figure 1(b). Based on the ternary interaction data, the *full-period global graph* including user-store, user-food and store-food interactions can be defined as follows:

Definition 2.*Full-Period Global Graph.* Given the ternary interaction dataset $\mathcal{X}$, the *full-period global graph* can be formulated as $\mathcal{G} = \{\mathcal{U} \cup \mathcal{S} \cup \mathcal{O}, \mathcal{E}_{\mathcal{US}} \cup \mathcal{E}_{\mathcal{UO}} \cup \mathcal{E}_{\mathcal{SO}}\}$. Let $\mathcal{E}_{\mathcal{US}}, \mathcal{E}_{\mathcal{UO}}$ and $\mathcal{E}_{\mathcal{SO}}$ be edges that represent user–store, user–food, store–food relations. For each record $x = (u, s, O, m)$, there exists the nodes $\{u, s, o_j\}$ and the bi-directional interaction edges $(u, s) \in \mathcal{E}_{\mathcal{US}}, (u, o_j) \in \mathcal{E}_{\mathcal{UO}}, (s, o_j) \in \mathcal{E}_{\mathcal{SO}}$. Each edge is attached with a time period attribute m.

Particularly, $\mathcal{G}$ is depicted in Figure 3(a). In this work, we only focus on recommending stores, an urgent practical task for takeaway applications, while the food recommendation is left as future work. In order to consider the user's preference for food in the store

recommendation, we define the most frequently clicked food set of the store s with the fixed size N' as $\Gamma(s)$. Based on the above definitions, the takeaway recommendation task can be defined as follows:

Definition 3. ***Takeaway Recommendation***. Given the *period-varying ternary interaction data* $\mathcal{X}$ and *full-period global graph* $\mathcal{G}$, DPVP aims to recommend the stores $s \in \mathcal{S}$ with the food set $\Gamma(s)$ that user u would be interested in at time period m.

4 METHODOLOGY

In this section, we will illustrate in detail how to realize the aforementioned takeaway recommendation based on the *period-varying ternary interaction data* and *full-period global graph* constructed based on it. The overview of our proposed DPVP is depicted in Figure 3. Firstly, we propose an interaction-aware module to decompose the *full-period global graph* into store-level and food-level graphs, which can also be denoted as dual interaction graphs. Then, we design a unified time-based decomposition module to deal with the dual graphs to obtain dual period-specific graphs. After aggregation on dual full-period and period-specific graphs, our model can model users' dual period-shared and period-specific preference representations. In particular, due to the multiple foods in the food set, a user-aware gate mechanism is designed to effectively ensemble different food information to obtain the personalized representation of the food set. Then, a time-aware gate network is designed to adaptively obtain the dual period-varying representations, finally followed by a prediction layer for our takeaway recommendation.

In the following, we first describe the construction process of dual period-varying graphs including the dual interaction-aware and time-based decomposition modules in Section 4.1. Then, we detailedly illustrate the aggregation of dual interaction-aware graphs for enriching the representations as well as the user-aware gating network for personalized food representation in Section 4.2. Furthermore, we present the modeling of the period-varying preferences in Section 4.3, followed by the prediction layer and optimization process illustrated in Section 4.4.

4.1 Dual Period-Varying Multigraphs Construction

4.1.1 Dual Interaction-Aware Module. To capture the dual preferences, we propose an interaction-aware module to decompose *full-period global graph* into the dual interaction graphs, i.e., food-level and store-level graphs shown in Figure 3(b). Specifically, in the food-level graph with food as the central node, there exist two kinds of interaction edges: (1) user-food interaction edge implies the user's preference for food; (2) food-store edge which can form a food-store-food structure indicating the same-store relationship between foods. Similarly, in the store-level graph with the store as the central node, there are user-store and store-food edges, where the formed store-food-store structure implies the food similarity of stores. Thereby, the dual interaction graphs including food-level and store-level graphs can be correspondingly formulated as:

$$\begin{aligned} \mathcal{G}^{[o]} &= \{\mathcal{U} \cup \mathcal{S} \cup \mathcal{O}, \mathcal{E}_{\mathcal{U}\mathcal{O}} \cup \mathcal{E}_{\mathcal{S}\mathcal{O}}\}, \\ \mathcal{G}^{[s]} &= \{\mathcal{U} \cup \mathcal{S} \cup \mathcal{O}, \mathcal{E}_{\mathcal{U}\mathcal{S}} \cup \mathcal{E}_{\mathcal{S}\mathcal{O}}\}. \end{aligned} \tag{1}$$

4.1.2 Time-Based Decomposition Module. With the aim of learning the period-varying preferences, we design a unified time-based decomposition module to deal with the dual interaction graphs. Specifically, each edge set in dual graphs is unified as $\mathcal{E}$, and the time period of each interaction edge c is denoted as c_t. In the time-based decomposition module, the hard weight of each edge c and the edge set in time period m can be formulated as:

$$\begin{aligned} w_c^m &= \mathbb{I}(c_t = m), \\ \mathcal{E}^m &= \{w_c^m * c\}, \forall c \in \mathcal{E}, \end{aligned} \tag{2}$$

where $\mathbb{I}(\cdot)$ represents the indicator function. Thereby, given the time period m of each interaction data, the dual interaction graphs $\mathcal{G}^{[o]}, \mathcal{G}^{[s]}$ can be decomposed into M dual period-specific graphs depicted in Figure 3(c) as $\forall m = 1, 2, ..., M$:

$$\begin{aligned} \mathcal{G}_m^{[o]} &= \{\mathcal{U} \cup \mathcal{S} \cup \mathcal{O}, \mathcal{E}_{\mathcal{U}\mathcal{O}}^m \cup \mathcal{E}_{\mathcal{S}\mathcal{O}}^m\}, \\ \mathcal{G}_m^{[s]} &= \{\mathcal{U} \cup \mathcal{S} \cup \mathcal{O}, \mathcal{E}_{\mathcal{U}\mathcal{S}}^m \cup \mathcal{E}_{\mathcal{S}\mathcal{O}}^m\}. \end{aligned} \tag{3}$$

4.2 Dual Interaction-Aware Preference

4.2.1 Dual Embedding Layers. Before aggregating graphs, we propose to initialize the embeddings of users, stores, and foods. Since we hope to capture the user's preferences for two completely different types of entities (i.e., store and food) in dual subgraphs, we initialize two groups of embeddings for the two graphs. Moreover, intuitively, users hold the aforementioned period-shared preferences in different time periods and there is no significant difference in the meaning of foods and stores over different time periods. Therefore, we utilize the same initialization embedding group of the same interaction-aware level for the subgraphs of different time periods at the same food/store level.

Take a user $u \in \mathcal{U}$ as an example: after turning u into the food-level user matrix $H_u^{[o]} \in \mathbb{R}^{|\mathcal{U}| \times d}$ and the store-level matrix $H_u^{[s]} \in \mathbb{R}^{|\mathcal{U}| \times d}$ with d denoting the embedding dimension, we can obtain the initialization embedding of user node u as $h_u^{[o],0}$ and $h_u^{[s],0}$ respectively. For the subgraphs of time period m, the user embeddings of the food-level and store-level are initialized as $h_{u,m}^{[o],0} = h_u^{[o],0}$ and $h_{u,m}^{[s],0} = h_u^{[s],0}$ correspondingly. Since the dual interaction-aware preference modeling process of dual graphs in all periods and period-specific graphs is the same and the aggregation method of the food-level graphs is similar to that of store-level graphs, we mainly elaborate the food-level period-specific preference modeling process of graphs $\mathcal{G}_m^{[o]}$ in time period m.

4.2.2 Single-Level Preference. Following [8, 28], we simply apply the mean-pooling operation to aggregate neighbor embeddings in the food-level graph $\mathcal{G}_m^{[o]}$. In detail, the aggregated neighbor embedding for the central food node in the food-level graph can be formulated as:

$$\mathbf{h}_{o,m}^{[o],l+1} = \frac{1}{c_{o,m}^{(u,o)}} \sum_{(u,o) \in \mathcal{E}_{\mathcal{U}O}^m} \mathbf{h}_{u,m}^{[o],l} + \frac{1}{c_{o,m}^{(s,o)}} \sum_{(s,o) \in \mathcal{E}_{\mathcal{S}O}^m} \mathbf{h}_{s,m}^{[o],l}, \tag{4}$$

where $\frac{1}{c_{o,m}^{(u,o)}}$ and $\frac{1}{c_{o,m}^{(s,o)}}$ are normalization constants and $c_m^{(u,o)}/c_m^{(s,o)}$ is equal to the in-degree from neighbor user/store nodes to the food node of o. Since both the user and store interact solely with food

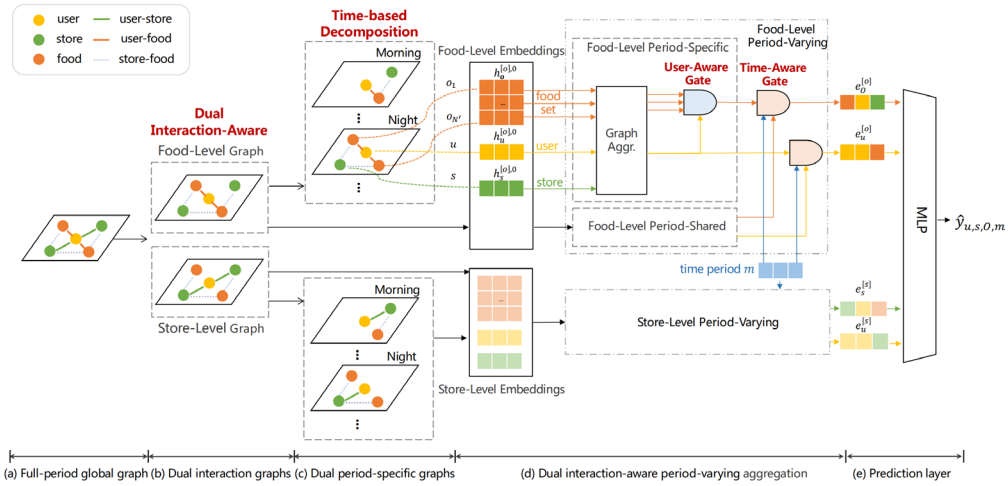


Figure 3: Overview of our proposed DPVP model.

nodes in the food-level graph, the aggregated neighbor representations of the user and store can be formulated as:

$$\begin{aligned}\mathbf{h}_{u,m}^{[o],l+1} &= \frac{1}{c_{u,m}^{(u,o)}} \sum_{(u,o)\in\mathcal{E}_{uO}^{m}} \mathbf{h}_{o,m}^{[o],l}, \\ \mathbf{h}_{s,m}^{[o],l+1} &= \frac{1}{c_{s,m}^{(s,o)}} \sum_{(s,o)\in\mathcal{E}_{SO}^{m}} \mathbf{h}_{o,m}^{[o],l}.\end{aligned} \tag{5}$$

Eventually, the weighted-pooling operation is applied to generate the aggregated user/food representations in the food-level graph at time period m by operating on propagated L layers:

$$\mathbf{h}_{o,m}^{[o],*} = \sum_{l=0}^{L} \alpha_l \mathbf{h}_{o,m}^{[o],l}, \quad \mathbf{h}_{u,m}^{[o],*} = \sum_{l=0}^{L} \alpha_l \mathbf{h}_{u,m}^{[o],l}, \tag{6}$$

where α_l denotes the importance of the l-th layer representation in constituting the final embedding. Similar to LightGCN [9], we uniformly set α_l as $1/(l+1)$ since the focus of our work is not on the choice of α_l. Therefore, after the aggregation of the food-level graph, we can obtain the food-level representations of the user u and food o as $\mathbf{h}_{o,m}^{[o],*}$ and $\mathbf{h}_{u,m}^{[o],*}$. After the similar aggregation method, we can also obtain the final aggregated embeddings of user u and store s in store-level graph as $\mathbf{h}_{u,m}^{[s],*}$, $\mathbf{h}_{s,m}^{[s],*}$.

4.2.3 Personalized Food Representation. For each store, its N' most frequently clicked foods are selected as its candidate foods. However, users' interest in food varies from person to person. For instance, user A may prefer baozi, while user B is more interested in noodles in the same store. In order to model the personalized food representation, a user-aware gate mechanism is designed to effectively ensemble different food information in the candidate food set $O = \Gamma(s)$. Specifically, the user-aware gating network g_u produces a distribution over N' foods based on the input, and the final ensembled personalized food representation is formulated as the weighted sum of the graph representations of all the candidate foods of the store:

$$\begin{aligned}g_u(o_i) &= \frac{exp(\mathbf{h}_{o_i,m}^{[o],*} \times \mathbf{h}_{u,m}^{[o],*\top})}{\sum_{j=1}^{N'} exp(\mathbf{h}_{o_j,m}^{[o],*} \times \mathbf{h}_{u,m}^{[o],*\top})}, \\ \mathbf{h}_{O,m}^{[o],*} &= \sum_{i=1}^{N'} g_u(o_i) \mathbf{h}_{o_i,m}^{[o],*}.\end{aligned} \tag{7}$$

4.3 Period-Varying Preference

To capture the period-varying patterns of the interaction, we obtain the period-specific enriched graph embedding group at each time period m as $\mathbf{h}_{u,m}^{[o],*}$, $\mathbf{h}_{O,m}^{[o],*}$, $\mathbf{h}_{u,m}^{[s],*}$, $\mathbf{h}_{s,m}^{[s],*}$. Ideally, when making a recommendation in time period m, DPVP ought to merely utilize the corresponding graph embedding group in time period m for prediction. In other words, this is a hard one-hot integration method. However, considering the following factors: (1) the user's interactions in different periods imply similar preferences; (2) period-specific data appear sparse compared to full-period interaction data, we adopt a soft time-aware gating method instead of the hard integration method for the integration of representations in different time periods. The soft time-aware gate combines the representations in different time periods and full time periods, and further filters out irrelevant information while strengthening the relevant information of the target time period. Without loss of generality, due to the same integration methods of representations $\mathbf{h}_{u,m}^{[o],*}$,$\mathbf{h}_{O,m}^{[o],*}$,$\mathbf{h}_{u,m}^{[s],*}$,$\mathbf{h}_{s,m}^{[s],*}$, we take $\mathbf{h}_{u,m}^{[o],*}$ as an example to illustrate

the time-aware gate integration as follows:

$$g_m(u,m) = \frac{exp(\mathbf{h}_{u,m}^{[o],*} \times [\mathbf{h}_u^{[o],*}, \mathbf{e}_m]^\top)}{\sum_{k=1}^{M} exp(\mathbf{h}_{u,k}^{[o],*} \times [\mathbf{h}_u^{[o],*}, \mathbf{e}_k]^\top)},$$
$$\mathbf{e}_{u,m}^{[o]} = \sum_{m=1}^{M} g_m(u,m)\mathbf{h}_{u,m}^{[o],*}, \tag{8}$$

where $[\cdot]$ denotes the operation of concatenation and $\mathbf{e}_m$ indicates the embedding of time period m.

4.4 Prediction and Optimization

After the above operations, we obtain the dual period-varying representations. That is, $\mathbf{e}_{u,m}^{[o]}, \mathbf{e}_{o,m}^{[o]}$ generated from food-level graphs and $\mathbf{e}_{u,m}^{[s]}, \mathbf{e}_{s,m}^{[s]}$ aggregated from store-level graphs. Since the candidate food set O contains the fine-grained food information of the store s, we employ MLP (Multi-Layer Perceptrons) to build a bridge between fine-grained food and coarse-grained store levels to predict how likely the user would adopt the store. Eventually, the predicted score of candidate store s with candidate food set O for user u at time period m can be formulated as:

$$\hat{y}_{u,s,O,m} = MLP([\mathbf{e}_{u,m}^{[o]}, \mathbf{e}_{O,m}^{[o]}, \mathbf{e}_{u,m}^{[s]}, \mathbf{e}_{s,m}^{[s]}]). \tag{9}$$

Then, we adopt the Bayesian Personalized Ranking (BPR) [25] loss to emphasize that the observed interaction should be assigned a higher score than unobserved ones as follows:

$$\mathcal{L} = \sum_{(u,s,s')\in Y, O=\Gamma(s), O'=\Gamma(s')} -\ln \sigma(\hat{y}_{u,s,O,m} - \hat{y}_{u,s',O',m}), \tag{10}$$

where $Y = (u,s,s')|(u,s) \in R^+, (u,s') \in R^-$ denotes the pairwise training data; R^+ indicates the positive observed interaction set, and R^- represents the randomly-sampled negative set; $\sigma(\cdot)$ stands for the sigmoid function.

5 EXPERIMENTS

In this section, we present empirical results to demonstrate the effectiveness of our proposed DPVP. These experiments are designed to answer the following research questions:

- **RQ1** How does DPVP perform compared with state-of-the-art recommendation models?
- **RQ2** What is the effect of dual interaction-aware preference and period-varying preference modeling in our proposed DPVP?
- **RQ3** How does DPVP perform in the real-world online recommendations with practical metrics?
- **RQ4** How do hyper-parameters in DPVP impact recommendation performance?

5.1 Experimental Settings

Dataset Description. We conduct experiments on two real-world datasets[3] (denoted as MT-large and MT-small datasets, with large differences in data size) of two cities in two different time intervals from the Meituan Takeaway platform, one of the largest takeaway platforms in China. The time span of both datasets is eight days. In order to model users' period-varying preferences, we group the whole day into four time periods, which are 5: 00-10: 00 for Morning, 10: 00-15: 00 for Noon, 15:00-20:00 for Night and 20:00-5:00+1 for Late Night. To evaluate the model performance, we split the first six days' data for training, the following one-day data for validation, and the last day's data for testing. For each ground truth test data, we randomly sample 99 stores that user did not interact with as negative samples. Detailed statistics of these two datasets and detailed implementation are presented in Appendix A.1 and A.2.

Evaluation Metrics. To evaluate the performance of the recommendation models, we adopt four commonly-used metrics including three ranking metrics MRR[24], NDCG@K [12] and Hit@K (we set K as 10 by default), and an accuracy metric AUC[7].

5.2 Overall Performance Comparison (RQ1)

We compare our proposed models with four graph-free baselines (i.e., NeuMF [10], DNN, ENMF [4], SimpleX [19]) and seven graph-based baselines (i.e., GCN [26], GAT [30], NGCF [32], HGT [11], LightGCN [9], UltraGCN [20] and SVD-GCN [21]). The detailed information of these baselines is provided in Appendix A.3. In particular, in order to test whether the improvement of the model performance is caused by the dual interaction-aware and period-varying preference modeling proposed in our DPVP rather than the graph aggregation method of DPVP, we design DPVP(full-period global graph) which makes prediction based on the *full-period global graph* depicted in Figure 3(a). The overall performance on two adopted datasets is reported in Table 1. From the results, we can observe that:

- DPVP significantly outperforms all the competitive baselines. Compared to the best performance of baselines, for the two datasets, DPVP gains an average 5.86% improvement in terms of the three ranking metrics as well as an average 1.84% improvement with respect to AUC. Such significant gains verify that DPVP can greatly enhance the recommendation performance.
- The performance of DPVP(full-period global graph) with the same mean-pooling aggregation on the *full-period global graph* is remarkably lower than that of DPVP. This observation demonstrates that the significant improvement of our model mainly lies in our proposed dual interaction-aware and period-varying preference modeling.
- The graph-based methods universally outperform most graph-free methods, indicating the importance of enhancing interactions by incorporating high-order neighborhood information to alleviate data sparsity.
- The performance of MT-large with large data volume is better than that of MT-small with small data volume for the same model. The main reason would be that with more data in each time period, the model performs better in modeling time-related behaviors.

5.3 Ablation Study (RQ2)

As the modeling of dual interaction-aware and period-varying preferences are the core of DPVP, we conduct the following ablation studies to investigate the effectiveness of these core modules.

[3]We collected the Meituan datasets because no public datasets are suitable for our task. We will release part of experimental datasets later and our code will be available in https://github.com/17231087/DPVP.git.

Model	MT-small				MT-large			
	Hit@10	NDCG@10	AUC	MRR	Hit@10	NDCG@10	AUC	MRR
NeuMF	0.3957	0.2104	0.7814	0.1788	0.4554	0.2488	0.8198	0.2101
DNN	0.4190	0.2503	0.7549	0.2188	0.6073	0.3898	0.7903	0.3174
ENMF	0.5513	0.3845	0.7356	0.3450	0.5911	0.4285	0.7496	0.3887
SimpleX	0.4851	0.2710	0.8171	0.2276	0.5634	0.3289	0.8457	0.2763
GCN	0.4446	0.2478	0.7968	0.2110	0.6288	0.4199	0.8535	0.3698
GAT	0.4384	0.2488	0.7529	0.2105	0.6202	0.3871	0.8361	0.3290
NGCF	0.5009	0.2839	0.8163	0.2411	0.6358	0.4222	0.8593	0.3725
HGT	0.4467	0.2474	0.8032	0.2097	0.6217	0.3905	0.8352	0.3329
LightGCN	0.4942	0.2784	0.7979	0.2330	0.6309	0.4250	0.8562	0.3756
Ultra-GCN	0.3666	0.2057	0.7624	0.1777	0.4959	0.3164	0.7751	0.2840
SVD-GCN	0.5745	0.3973	0.8205	0.3402	0.6370	0.4248	0.8609	0.3736
DPVP(full-period global graph)	0.5094	0.3022	0.7875	0.2562	0.6324	0.4117	0.8621	0.3592
DPVP	**0.6167***	**0.4180***	**0.8392***	**0.3715***	**0.6599***	**0.4563***	**0.8741***	**0.4075***
Imp%	+7.3455	+5.2102	+2.2791	+7.6812	+3.5950	+6.4877	+1.3919	+4.8366

Table 1: Overall performance on MT-small and MT-large datasets. The last row Imp% indicates the relative improvements of the best performing method (bolded) over the strongest baselines (underlined) and marker * indicates that the improvement is statistically significant compared with the best baseline (paired t-test with p-value < 0.005).

Model	MT-small				MT-large			
	Hit@10	NDCG@10	AUC	MRR	Hit@10	NDCG@10	AUC	MRR
DPVP(user-food)	0.3990	0.2531	0.7487	0.2298	0.4054	0.2299	0.7651	0.1986
DPVP(food-level)	0.4438	0.2750	0.7804	0.2444	0.4547	0.2842	0.7845	0.2602
DPVP(user-store)	0.5355	0.3432	0.7888	0.3017	0.6110	0.3879	0.8558	0.3359
DPVP(store-level)	0.5461	0.3588	0.7902	0.3161	0.6271	0.4158	0.8639	0.3664
DPVP(global level)	0.5865	0.3909	0.8296	0.3467	0.6494	0.4476	0.8714	0.3998
DPVP	**0.6167***	**0.4180***	**0.8392***	**0.3715***	**0.6599***	**0.4563***	**0.8741***	**0.4075***

Table 2: Impact of dual interaction-aware preference modeling and * indicates p-value < 0.005.

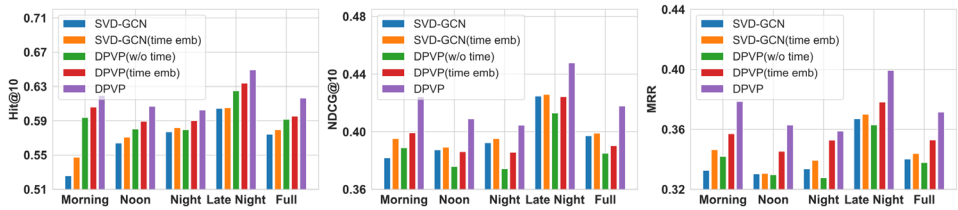


Figure 4: Impact of period-varying modeling on results for different time periods in MT-small dataset.

5.3.1 Impact of Dual Interaction-Aware Preference. To investigate whether DPVP can benefit from dual interaction-aware preference modeling, we propose five ablation versions of DPVP: (1) DPVP (user-food). This baseline only considers the user-food interactions; (2) DPVP(user-store). This baseline merely considers the binary user-store interactions, usually modeled by the traditional recommender systems. (3) DPVP(food-level). This baseline only retains food-level graphs by removing the store-level graphs in Figure 3(b); (4) DPVP(store-level). This baseline only retains store-level graphs by removing the food-level graphs depicted in Figure 3(b); (5) DPVP (global level). This baseline directly performs time-based decomposition on *full-period global graph* without dual interaction-aware module, that is $\mathcal{G}_*^{[o]}, \mathcal{G}_*^{[s]}$ are replaced with $\mathcal{G}_*$). The third and fourth baselines are devised to verify whether our method is still effective with only a single-level interaction graph. For fairness, in order to avoid the influence of the parameter amount on the results, we double the embedding size of the single-level interaction

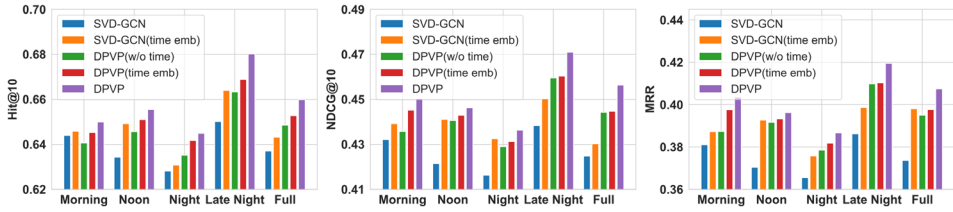


Figure 5: Impact of period-varying modeling on results for different time periods in MT-large dataset.

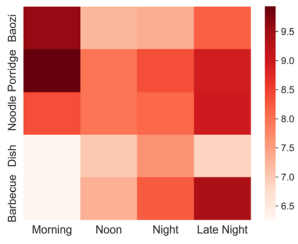


Figure 6: Visualization of the average predicted scores of five major food categories over different time periods.

and single-level graph model so that the parameter amount is the same as that of DPVP model. Besides, for the global level model, we replace the dual aggregated user representations $\mathbf{e}_u^{[o]}, \mathbf{e}_u^{[s]}$ with the global-level graph aggregated user representation respectively as part of the input of the prediction layer, so that the parameter amount is consistent with that of DPVP. From the results shown in Table 2, we have the following observations:

- As we expected, compared with the models that only consider binary interaction information DPVP(user-food) and DPVP(user-store), the models that consider ternary interactive information has better performance.
- The performance of the single-level model is significantly worse than that of the dual interaction-aware model. This is within expectation because dual interaction-aware modeling can capture the additional level of user preferences compared to single-level modeling.
- Encouragingly, the performance of preference modeling based on dual interaction-aware graphs is better than that based on global-level graphs regardless of hierarchical modeling. This observation indicates the effectiveness of our proposed dual interaction-aware preference modeling. This result is not surprising because the global level graph models the interaction at the two levels at the same time, which may confuse the user's preferences at the two levels, ultimately resulting in poor user representation. On the contrary, our model explicitly captures the hierarchical preference at different levels by considering the interaction with users at different levels.

5.3.2 Impact of Period-Varying Modeling. In order to verify the effectiveness of period-varying preference modeling, we conduct three additional ablation baselines as follows:

- SVD-GCN(time emb). On the basis of the original SVD-GCN with only user and store input, the time embedding $\mathbf{e}_m$ is added as an input to the prediction layer of SVD-GCN.
- DPVP(w/o time). This baseline completely removes the modeling of the period-specific subgraphs with only the dual interaction graphs in Figure 3(b), which is designed to prove the effectiveness of period-varying modeling in DPVP.
- DPVP(time emb). This baseline removes the modeling of the period-specific graphs in Figure 3(c) and similarly adds time embedding as input $\mathbf{e}_m$, which is devised to validate whether our period-varying modeling method is better than the method of simply adding period information.

To evaluate the improvement of our model in different periods, Figure 4 and Figure 5 present the performance of the competitive baselines SVD-GCN, DPVP as well as the above-mentioned three ablation baselines on the two datasets. Details of performance on AUC metric for the two datasets can be found in Appendix A.4. Please note that the *Full* in figures represents the performance of models for full-period test data without distinguishing the time periods. According to the results, we can observe that:

- Encouragingly, DPVP achieves the best results under various metrics over different time periods.
- With time embedding as input, the performance of SVD-GCN(time emb) and DPVP(time emb) is improved compared to the corresponding baselines (i.e., SVD-GCN and DPVP(w/o time)) ignoring the time information, which verifies that time information is beneficial to the modeling of takeaway recommendation indeed.
- Compared to DPVP(w/o time) without period-varying modeling component and DPVP(time emb) simply taking time embedding as input, DPVP performs better through modeling subgraphs of different periods constructed based on the periodic patterns. Specifically, compared with DPVP(w/o time), for MT-large dataset, the minimum improvement of the three metrics over different periods is 1.17% (i.e., MRR in Noon period) and for MT-small dataset, the smallest improvement of these metrics is 3.91% (i.e., Hit@10 for Late Night period).

In particular, to prove our model can effectively capture the periodic period-varying patterns shown in Figure 2, we provide a

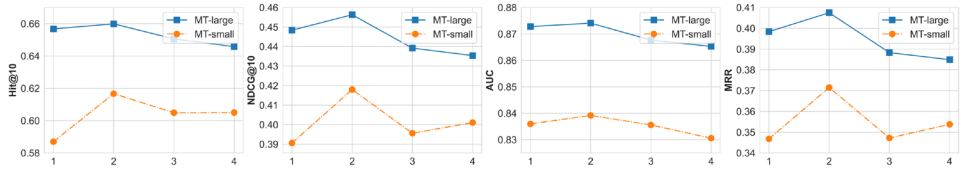


Figure 7: Performance w.r.t the number of layers L on the two datasets.

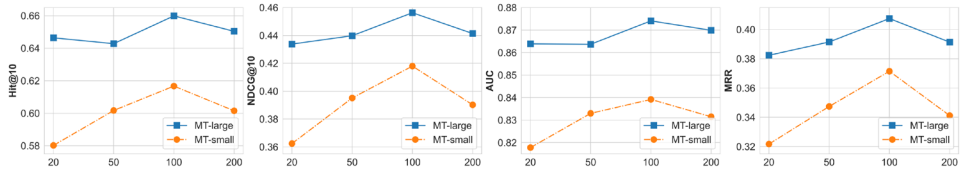


Figure 8: Performance w.r.t the dimension of embedding vectors d on the two datasets.

visualization of the average predicted scores (i.e., $\hat{y}_{u,s,O,m}$ in Eq. 9) of five major food categories o (o in O) over different time periods in MT-small test data. As shown in Figure 6, we can clearly observe that: the predicted score of Baozi/Porridge commonly served as a breakfast food in the Morning period is significantly higher than that in other periods; Likewise, the barbecue, which is usually chosen as a midnight snack, also scores higher in the Late Night period; Besides, encouragingly, the predicted scores of Dish are higher in Noon and Night periods than those in Morning and Late Night periods as expected; Moreover, the interaction characteristics between users and noodle category do not change significantly over time, as shown in Figure 2. Within the expectation, there is no apparent difference in the predicted scores of noodle over different time periods.

5.4 Online A/B test(RQ3)

We deploy our method to the Meituan Takeaway recommender system and conduct a two-week online A/B test. We leverage the learned graph embeddings from our DPVP model as additional features in the downstream recommendation model, and increase the GMV(Gross Merchandise Volume) by 0.70%, the CTCVR(Click Through & Conversion Rate) [18] by 0.68% in one of the main recommendation scenarios, which demonstrates the effectiveness of our method.

5.5 Hyper-Parameter Studies (RQ4)

Impact of Model Depth. To study the impact of the model depth of the graphs, we vary L from 1 to 4. As shown in Figure 7 shown, we find that DPVP basically improves first and then drops, which is a trade-off of performance gain from high-order information (by stacking L layers to reach L-hop neighbors) and performance degradation caused by over-smoothing. Since DPVP achieves the best performance with two layers, we set $L = 2$ for both datasets.

Impact of Embedding Dimension. To investigate the performance of DPVP on the two datasets with respect to the dimension of embedding vectors (i.e., d), we vary d in $\{20, 50, 100, 200\}$. As shown in Figure 8, with the increase of embedding dimension, the performance of DPVP shows a trend of first increasing and then decreasing and DPVP performs best with the embedding dimension equaling 100. Thus, we set $d = 100$ for the two datasets.

6 CONCLUSION

In this paper, in order to model users' dual period-varying preferences in takeaway recommendation, we propose a novel graph-based model, named DPVP. Specifically, we design a dual interaction-aware module for modeling the dual preferences. Furthermore, we propose a time-based decomposition module as well as a time-aware gating network for capturing time-varying preferences. Extensive experiments on two real-world datasets from Meituan platform and online A/B tests empirically demonstrate the superiority of our model. Additionally, studies prove that our model can improve the recommendation performance for all time periods and is able to capture users' varying preferences over different periods. As further work, we plan to explore incorporating additional time factors, such as seasons and holidays, to more accurately model users' varying preferences for takeaway recommendation.

ACKNOWLEDGMENTS

The research work is supported by the National Key Research and Development Program of China under Grant No. 2021ZD0113602, the National Natural Science Foundation of China under Grant No. 62176014, the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] 2023. Number of users of the online food delivery market worldwide in 2022 and 2027, by region. https://www.statista.com/forecasts/1358171/online-food-delivery-users-by-region-worldwide.

[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*. 1583–1592.

[4] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[5] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *Proceedings of the 2018 world wide web conference*. 1775–1784.

[6] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[7] Cesar Ferri, José Hernández-Orallo, and Peter A Flach. 2011. A coherent interpretation of AUC as a measure of aggregated classification performance. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 657–664.

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[10] 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.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.

[12] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[13] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[15] Yinfeng Li, Chen Gao, Xiaoyi Du, Huazhou Wei, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Automatically Discovering User Consumption Intents in Meituan. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3259–3269.

[16] Shaochuan Lin, Yicong Yu, Xiyu Ji, Taotao Zhou, Hengxu He, Zisen Sang, Jia Jia, Guodong Cao, and Ning Hu. 2022. Spatiotemporal-Enhanced Network for Click-Through Rate Prediction in Location-based Services. *arXiv preprint arXiv:2209.09427* (2022).

[17] Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[19] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A simple and strong baseline for collaborative filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1243–1252.

[20] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.

[21] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. 2022. SVD-GCN: A Simplified Graph Convolution Paradigm for Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1625–1634.

[22] Andreas Pfadler, Huan Zhao, Jizhe Wang, Lifeng Wang, Pipei Huang, and Dik Lun Lee. 2020. Billion-scale recommendation with heterogeneous side information at taobao. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1667–1676.

[23] Yi Qi, Ke Hu, Bo Zhang, Jia Cheng, and Jun Lei. 2021. Trilateral Spatiotemporal Attention Network for User Behavior Modeling in Location-based Search. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3373–3377.

[24] Dragomir R Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating Web-based Question Answering Systems.. In *LREC*. Citeseer.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[26] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[27] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In *International Workshop on Machine Learning Methods for Recommender Systems*.

[28] Jianing Sun, Yingxue Zhang, Chen Ma, Mark Coates, Huifeng Guo, Ruiming Tang, and Xiuqiang He. 2019. Multi-graph convolution collaborative filtering. In *2019 IEEE international conference on data mining (ICDM)*. IEEE, 1306–1311.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[31] Daheng Wang, Meng Jiang, Munira Syed, Oliver Conway, Vishal Juneja, Sriram Subramanian, and Nitesh V Chawla. 2020. Calendar graph neural networks for modeling time structures in spatiotemporal user behaviors. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2581–2589.

[32] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[33] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[34] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The world wide web conference*. 2091–2102.

[35] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[36] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[37] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. 2020. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2861–2868.

[38] Tianchi Yang, Luhao Zhang, Chuan Shi, Cheng Yang, Siyong Xu, Ruiyu Fang, Maodi Hu, Huaijun Liu, Tao Li, and Dong Wang. 2022. Gated Hypergraph Neural Network for Scene-Aware Recommendation. In *International Conference on Database Systems for Advanced Applications*. Springer, 199–215.

[39] 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 *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[40] Yantao Yu, Weipeng Wang, Zhoutian Feng, and Daiyue Xue. 2021. A dual augmented two-tower model for online large-scale recommendation. *DLP-KDD* (2021).

[41] 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.

[42] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[43] Jun Zhao, Zhou Zhou, Ziyu Guan, Wei Zhao, Wei Ning, Guang Qiu, and Xiaofei He. 2019. Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2347–2357.

[44] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining*. 425–434.

[45] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

A EXPERIMENT

A.1 Data desription

The details of the statistics of MT-small and MT-large datasets are summarized in Table 3.

Table 3: The statistics of the two datasets from Meituan.

Dataset	#User	#Store	#Food	#User-Store Interaction
MT-small	56,887	4,059	5,952	180,283
MT-large	385,381	18,770	17,111	1,492,164

A.2 Detailed Implementation

The hidden sizes of MLP described in Section 4.4 are set as [400,200] by default. For each store, we get its 10 most frequently clicked foods as its candidate food in the training data (i.e., $N' = 10$ in Eq. 7). Section 5.5 reports the impact of other essential hyper-parameters including the model depth L and the dimension of latent vectors d (i.e., the embeddings of users, stores, foods and time periods) in DPVP, and we utilize the best parameter settings. Under the above settings, all models are trained using AdamW optimizer [17] with a learning rate of 1e-4, batch size of 1024 and weight decay of 1e-5.

A.3 Baselines

We compare our proposed model against two groups of state-of-the-art recommendation methods as follows:

(1) *Graph-free* methods:

- NeuMF [10]. This method combines linear matrix factorization with the nonlinear MLP to extract low-order and high-order features for prediction.
- DNN. This is a vanilla deep-learning model for recommendation prediction.
- ENMF [4]. This model is a neural recommendation model that considers all samples in each parameter update without sampling.
- SimpleX [19]. This method integrates Matrix Factorization and user behavior modeling for prediction.

(2) *Graph-based* methods:

- GCN [26]. This method enrichs user and store representations through feature transformation, neighborhood aggregation, and nonlinear activation on graph.
- GAT [30]. Considering the different importance of neighbor nodes to the central node, this model leverages the attention mechanism [29] to weight the importance of neighbors in the aggregation process.
- NGCF [32]. This model advances the vanilla GCN by additionally encoding the interactions via an element-wise multiplication.
- HGT [11]. This GNN model designs edge-type and node-type dependent parameters to deal with different interaction types and learn the node representations from the heterogeneous graph.
- LightGCN [9]. This method simplifies GCN by removing unnecessary feature transformation and nonlinear activation parts. It merely adopts neighborhood aggregation to capture the collaborative filtering effect.
- UltraGCN [20]. This model further simplifies LightGCN by replacing neighborhood aggregation with weighted MF, leading to faster convergence and less complexity.
- SVD-GCN [21]. This method simplifies GCN-based methods by solely exploiting K-largest singular vectors of a flexible truncated SVD for the recommendation.

All graph-based methods are aggregated on the *full-period global graph* for the enriched representations. Since we focus on store recommendation in this work, we treat the store as the item. In all the baselines except CF-based methods (i.e., NeuMF and NGCF) , we adopt MLP described in Section 4.4 as the final prediction layer with user and store representations as input. To be fair, the parameter settings of MLP in baselines are the same as that of our proposed DPVP.

A.4 Supplement for impact of period-varying modeling

In Figure 9 and Figure 10, we present the impact of period-varying modeling on results for different time periods on AUC metric in the two datasets.

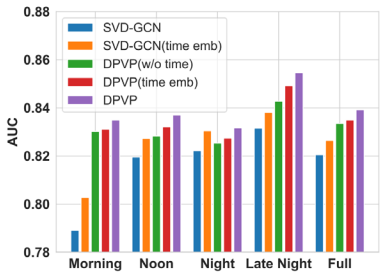


Figure 9: Impact of period-varying modeling on results for different time periods on AUC metric in MT-small dataset.

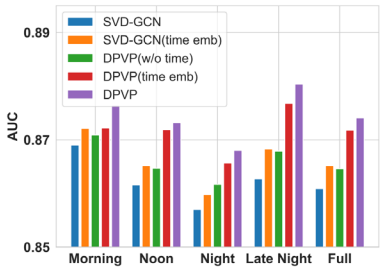


Figure 10: Impact of period-varying modeling on results for different time periods on AUC metric in MT-large dataset.

A Multi-stage Framework for Online Bonus Allocation Based on Constrained User Intent Detection

Chao Wang*†
Meituan Group
Shanghai, China
wangchao153@meituan.com

Xiaowei Shi*
Meituan Group
Beijing, China
shixiaowei02@meituan.com

Shuai Xu*
Meituan Group
Shanghai, China
xushuai12@meituan.com

Zhe Wang†
Meituan Group
Beijing, China
wangzhe36@meituan.com

Zhiqiang Fan
Meituan Group
Beijing, China
fanzhiqiang02@meituan.com

Yan Feng
Meituan Group
Shanghai, China
fengyan14@meituan.com

An You
Meituan Group
Beijing, China
youan@meituan.com

Yu Chen
Meituan Group
Beijing, China
chenyu17@meituan.com

ABSTRACT

With the explosive development of e-commerce for service, tens of millions of orders are generated every day on the Meituan platform. By allocating bonuses to new customers when they pay, the Meituan platform encourages them to use its own payment service for a better experience in the future. It can be formulated as a multi-choice knapsack problem (MCKP), and the mainstream solution is usually a two-stage method. The first stage is user intent detection, predicting the effect for each bonus treatment. Then, it serves as the objective of the MCKP, and the problem is solved in the second stage to obtain the optimal allocation strategy. However, this solution usually faces the following challenges: (1) In the user intent detection stage, due to the sparsity of interaction and noise, the traditional multi-treatment effect estimation methods lack interpretability, which may violate the domain knowledge that the marginal gain is non-negative with the increase of the bonus amount in economic theory. (2) There is an optimality gap between the two stages, which limits the upper bound of the optimal value obtained in the second stage. (3) Due to changes in the distribution of orders online, the actual cost consumption often violates the given budget limit. To solve the above challenges, we propose a framework that consists of three modules, i.e., User Intent Detection Module, Online Allocation Module, and Feedback Control Module. In the User Intent Detection Module, we implicitly model the treatment increment based on deep representation learning and constrain it to be non-negative to achieve monotonicity constraints. Then, in order to reduce the optimality gap, we further propose a convex constrained model to increase the upper bound of the optimal value. For the third challenge, to cope with the fluctuation of online bonus consumption, we leverage a feedback control strategy in the framework to make the actual cost more accurately approach the given budget limit. Finally, we conduct extensive offline and online experiments, demonstrating the superiority of our proposed framework, which reduced customer acquisition costs by 5.07% and is still running online.

CCS CONCEPTS

• **Information systems** → **Computational advertising**; • **Applied computing** → *Electronic commerce*.

KEYWORDS

E-commerce, Bonus Allocation, Multi-treatment Effect Estimation, Monotonic Constraint, Convex Constraint

ACM Reference Format:
Chao Wang, Xiaowei Shi, Shuai Xu, Zhe Wang, Zhiqiang Fan, Yan Feng, An You, and Yu Chen. 2023. A Multi-stage Framework for Online Bonus Allocation Based on Constrained User Intent Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599764

1 INTRODUCTION

With the explosion of e-commerce, customer acquisition has become increasingly important and competitive. Assigning bonuses or coupons to customers is a crucial way to promote user conversion in customer acquisition management. Therefore, the most critical problem is how to convert more new customers within a given budget limit. For this problem, the mainstream solution is a two-stage method [2, 15, 27, 33]: a combination of machine learning (ML) and operation research (OR). The first stage is user intent detection, which predicts the user's conversion probability

*These authors contributed equally to this research.
†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 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.
KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599764

under each candidate bonus. The second stage is optimization. In this stage, based on the obtained conversion probability, the platform allocates an appropriate bonus amount online to achieve the overall optimal conversion. For example, Meituan, one of the most popular e-commerce platforms for service, takes tens of millions of orders daily. In its payment platform, bonuses are allocated to new customers through discounts on the order amount. In order to meet the needs of different users as much as possible, there are dozens of optional bonus amounts.

User intent detection is a multi-treatment effect estimation problem. Methods can be divided into several categories, such as stratification methods, matching models, meta-algorithms, and tree-based methods. Representation learning methods based on deep neural networks have also attracted much attention, including the pioneering method TARNET [21] and the PM model [20] for multi-treatment scenarios. Compared with traditional machine learning approaches, deep representation learning models are capable of automatically searching for correlated features and combining them to enable more effective and accurate counterfactual estimation [29]. In terms of optimization, the bonus allocation problem can be formulated as a multi-choice knapsack problem (MCKP). The bonus allocation is non-trivial because of the following challenges.

- For user intent detection, the existing model lacks interpretability, i.e., does not conform to domain knowledge, to make reliable predictions in the real world. In economic theory, the marginal gain of increasing the bonus amount should be non-negative [11]. However, due to the sparsity of the interaction and noise, as shown in Figure 1, the user response curve predicted by the traditional treatment effect estimation methods is not entirely monotonous, and there are many jitters.
- There is an optimality gap between the two stages of user intent detection and optimization. We prove that in the response curve obtained in the first stage, some bonus treatments have no chance of being decided, thereby reducing the upper bound of the optimal value.
- The distribution of online orders and users changes over time, such as during the week and weekends. How to adjust allocation strategy efficiently and dynamically so that the daily customer acquisition cost does not exceed the budget limit is crucial for budget control.

The main contribution of this work is to propose an online bonus allocation framework to address the above challenges, which consists of three modules: User Intent Detection Module, Online Allocation Module, and Feedback Control Module.

For the first challenge, we analyze the typical pattern of multi-treatment effect estimation methods based on representation learning. Based on it, we propose a monotonic constrained user intent detection model. It still deploys a multi-head network structure. The difference is that it implicitly models the effect increment of two adjacent bonus amounts and finally obtains the individual treatment effect through accumulation. The monotonic constraint is implemented by restricting the effect increment to be non-negative. In addition, the accumulation operator is added to the output layer

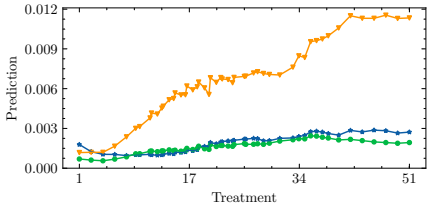


Figure 1: We randomly select three new customers online and obtain the potential outcomes predicted by the intent detection model under each bonus treatment, called the response curve.

to make the information fusion between the head layers more sufficient and to improve the stability of model training when there are many treatments.

For the second challenge, in the bonus allocation optimization problem, there are millions of new customers daily and dozens of bonus treatments, so the policy space is combinatorial and enormous. Inspired by [27, 32], we leverage the Lagrangian dual theory to calculate the Lagrangian multiplier to compress the policy space. Furthermore, we propose a user intent detection model with convex constraints that can be theoretically proven to reach an upper bound on the optimal value.

For the third challenge, we approach it as a feedback control problem and apply feedback control theory to address the dynamic system's response and control in the presence of external noise. Without loss of generality, the feedback control module can be applied not only in our scenario where customer acquisition cost (CAC) is the target constraint but also in other cost-related constraints.

We conduct extensive experiments to evaluate the performances of our proposed framework. Both online and offline experiments demonstrate that the proposed framework achieves better performance compared with other methods. The proposed bonus allocation system has brought significant profits to the platform and still runs online.

The rest of our paper is organized as follows. We describe the bonus allocation system in detail in Section 2. Then we introduce the three modules included in the proposed framework respectively in Section 3. We present the experimental setups and results in Section 4. We briefly review related work in Section 5. The conclusions are given in Section 6.

2 THE BONUS ALLOCATION FRAMEWORK IN MEITUAN PAYMENT

In this section, we introduce a bonus allocation algorithm framework for customer acquisition marketing in Meituan Payment, as illustrated in Figure 2,. It comprises a business layer and an algorithm layer. When a new customer places an order and makes a payment on the Meituan platform, the user intent detection module in the algorithm layer evaluates the customer's conversion probability for each optional bonus amount, which is provided by the

business layer and is consistent for all users. In the bonus allocation optimization module, we formulate an optimization problem based on the user conversion probability to determine the optimal allocation strategy that maximizes the total number of converted customers within a given budget limit, while adhering to the allocation rules in the business layer. We then apply the optimal allocation strategy in real-time to allocate each order. As the traffic to online orders is non-uniform, the budget consumption fluctuates. To ensure that the actual consumption approaches the budget limit accurately, we deploy a feedback control strategy to adjust the allocation strategy in real time.

Randomized controlled trials (RCTs) are considered the gold standard for estimating treatment effects. In this study, we collected samples through RCTs to assess the potential effects of different bonus treatments. The experiment was conducted as an online multivariate A/B test, which was deployed in the actual traffic of Meituan Payment and lasted for several days.

3 METHODOLOGY

In this section, we first formulate the problem of bonus allocation in section 3.1 and then propose the online allocation strategy based on the user's conversion probability detected in User Intent Detection Module described in section 3.2. Secondly, in order to meet the business knowledge and increase the upper bound of the optimal value under the Lagrangian relaxation paradigm, we leverage monotonic and convex constraints to the common multi-treatment effect estimation model in User Intent Detection Module detailed in section 3.3. Finally, we propose a feedback control strategy in section 3.4 so that online cost consumption does not violate the budget limit.

3.1 Preliminaries

Within a time period of one day, we assume that there are N new customers who visit the Meituan payment platform for transactions, and there are M types of candidate bonus amounts. For a given user i, we use $p_{i,j}$ to denote the conversion probability under the bonus amount j. The objective of bonus allocation is to identify an optimal allocation policy that converts as many users as possible within a given budget constraint B. Therefore, the bonus allocation problem can be formulated as the MCKP.

$$\max_{x_{i,j}} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} p_{i,j}x_{i,j}, \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} b_j p_{i,j}x_{i,j} \le B, \tag{2}$$

$$\sum_{j=1}^{M} x_{i,j} = 1,\ \forall i \in [N], \tag{3}$$

$$x_{i,j} \in \{0,1\},\ \forall i \in [N], \forall j \in [M], \tag{4}$$

Where if bonus j is allocated to user i, i.e $x_{ij} = 1$, thus $\mathcal{X} = \{x_{ij}\}$ denotes the optimal allocation policy. Eq.(1) denotes the objective, that is, the expected total conversion customers. Eq.(2) represents the budget constraint, b_j denotes the amount of bonus j, and only the converted users will consume the budget, so we calculate $p_{i,j}b_j$ represent the expected cost. Eq.(3) is because we only issue one bonus per user visit.

In order to solve this problem, we propose a framework including (1) Predicting the conversion probability in the **User Intent Detection Module**, (2) Solving the optimization problem in the **Online Allocation Module** to get optimal policy and making the online allocation, (3) Since new customers do not arrive uniformly, we are supposed to adjust the pace of budget consumption in the **Feedback Control Module** so that the cost does not violate the budget limit.

3.2 Online Allocation Module

The methods for solving MCKPs can be divided into two categories, one is the exact algorithm, such as dynamic programming and branch and bound method, and the other is the inexact algorithm, mainly including Lagrangian dual algorithm, evolutionary algorithm, etc. Due to the need for real-time allocation, it is difficult to obtain an exact solution. Inspired by [27, 32], we leverage the Lagrangian dual theory, so that we can calculate the Lagrangian multiplier λ to compress the parameter space $\mathcal{X}$.

By introducing a Lagrangian multiplier λ, we get the Lagrangian relaxation function of the original problem as follows:

$$L(\mathcal{X}, \lambda) = \sum_{i=1}^{N}\sum_{j=1}^{M} p_{i,j}x_{i,j} - \lambda\left(\sum_{i=1}^{N}\sum_{j=1}^{M} b_j x_{i,j}p_{i,j} - B\right), \tag{5}$$

Correspondingly, the dual problem is formulated as:

$$\min_{\lambda}\max_{x_{i,j}} \quad L(\mathcal{X}, \lambda), \tag{6}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} x_{i,j} = 1,\ \forall i \in [N], \tag{7}$$

$$x_{i,j} \in \{0,1\},\ \forall i \in [N], \forall j \in [M], \tag{8}$$

with optimality conditions:

$$\lambda\left(\sum_{i=1}^{N}\sum_{j=1}^{M} b_j x_{i,j}p_{i,j} - B\right) = 0, \tag{9}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{M} b_j x_{i,j}p_{i,j} - B \le 0, \lambda \ge 0. \tag{10}$$

Thanks to the Lagrangian dual decomposition, we remove Eq.(2) from the constraints, and for a fixed λ, the above optimization problem turns into a set of sub-problems (i.e., one for each new customer i^*):

$$\max_{x_{i^*,j}} \quad \sum_{j=1}^{M} x_{i^*,j}\left(p_{i^*,j} - \lambda p_{i^*,j}b_j\right), \tag{11}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} x_{i^*,j} = 1, x_{i*,j} \in \{0,1\}, \forall j \in [M]. \tag{12}$$

Eq.(11) holds because obtaining the optimal value for a new customer leads to the overall optimal value. When λ is given, all sub-problems become independent. Compared to the original problem with O(MN) decision variables, solving the sub-problem only

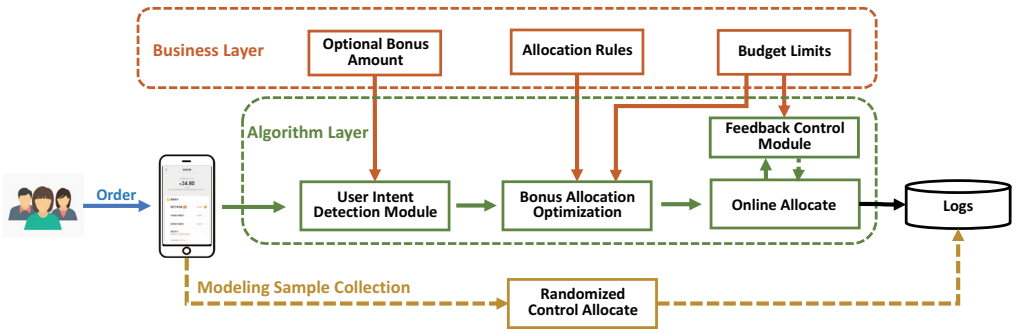


Figure 2: Our proposed framework for multi-stage bonus allocation at the algorithmic layer, including a user intent detection module, an online allocation module that optimizes the bonus allocation optimization problem, and a feedback control module.

requires traversing M bonuses, reducing the computational complexity to O(M). The optimal allocation policy is expressed as follows:

$$x_{i,j^*} = 1\{j^* = \arg\max_j p_{i,j} - \lambda p_{i,j} b_j\}, \tag{13}$$

Where $1\{\cdot\}$ is the 0/1 indicator function.

To solve the dual problem in Eq.(6), we can alternate between the following steps: (1) fix λ^* and calculate the optimal allocation solution X using Eq.(13), and (2) based on the fixed X^*, iterate λ to satisfy the optimality conditions described in Eq.(9,10). To improve the efficiency of the iterative process for λ, we leverage the bisection algorithm [27, 33] for searching. The initial value of λ is recommended to be set between 0 and 2 [26].

3.3 User Intent Detection Module

Due to the explosive development of deep learning, treatment effect estimation based on representation learning has become a hot topic. The architecture of representation learning is usually as shown in Figure 3 (a) [16, 20, 21, 30]. In order to prevent the influence of treatment variables from being lost during training, a multi-head network structure is often deployed. The loss in the network consists of two parts. $\mathcal{L}_{predict}$ represents the error between the predicted outcome $\hat{y}_k$ and the actual outcome y_k. In real scenarios, treatments are not always randomly issued, so the sample distribution of each treatment is not the same, which may cause the model to fail to obtain the real causal relationship between treatment and outcome, but a spurious relationship. In deep representation learning, we can transform the covariates from the original space to the latent space. $\mathcal{L}_{balance}$ represents the difference in the potential space of samples under different treatments, where the integral probability metric is widely used. However, we obtain samples through randomized controlled trials in Meituan payment customer acquisition so that this regular item can be eliminated. In our scenario, treatment corresponds to the bonus amount t, and we need to predict the effect of a customer under different bonus amounts, that is, the conversion probability.

3.3.1 *Monotonic constrained.*

Assumption 1. *Controlling other conditions unchanged, the user conversion probability and the bonus amount maintain a monotonically increasing relationship.*

Obviously, the bonus is a direct discount on the order amount in our scenario and does not include personalized creative content. Therefore, the higher the amount, the more attractive it is for user conversion. However, due to the presence of noise and the sparsity of the interaction data, we observe that even in randomized controlled trials, monotonicity cannot be guaranteed. To ensure monotonicity, as shown in Figure 3 (b), we propose a multi-treatment effect estimation model with monotonic constraints. We implicitly model the effect increment between two adjacent treatments and obtain individual treatment effects through accumulation. Next, we describe the model structure in detail.

A new customer u is represented by a set of characteristics (x, c, t), where x includes the statistical characteristics of the user's historical behavior, such as the number of transaction orders and the average amount of orders in the past n days, and user identity descriptions, such as age and membership status, consumption preference level for each business line, etc. c represents real-time contextual features, such as the amount of the order, which business line it originated from, and order time, etc., and t represents the exposed bonus treatment. The output of the shared dense layers L is defined as:

$$\phi = L(x), \tag{14}$$

Where ϕ denotes the common user representation learned from the shared layers, it enables the efficient sharing of information across multiple treatments.

As shown in Figure 3 (b), δ_k is used as input to head layers to model incremental effects. It includes two parts, context features c and treatment information δ_k^t. Since the contextual features are more important, drawing on the idea of the deep&wide model [7], we place it close to the output to strengthen the memorization of the model. δ_k^t includes the embedding of k-th and $(k-1)$-th treatment.

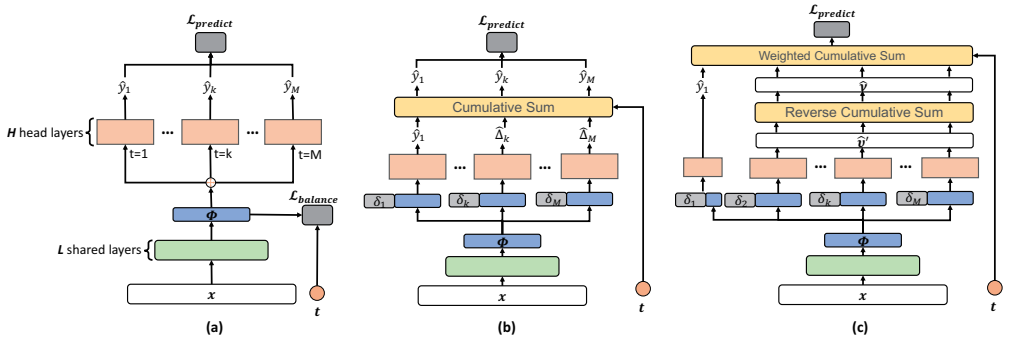


Figure 3: (a) Multi-treatment effect estimation pattern based on representation learning. A multi-head network structure is adopted, where T represents treatment. (b) The proposed multi-treatment effect estimation model with monotonic constraint. The outcome is calculated by accumulating the increment by modeling the positive effect increment between two adjacent sorted treatments. (c) The proposed convex constrained model is realized by restricting increments monotonically decreasing.

Considering the physical meaning of accumulation, δ_1^t will be used to model $\hat{y}_1$. For convenience, we construct a treatment placeholder t_0, and then δ_1^t contains the embedding of t_0 and the first treatment. We define:

$$\hat{\Delta}_k = f_k([\phi; \delta_k]), \tag{15}$$

Where the function $f_k(\cdot)$ is the head, $[\cdot;\cdot]$ denotes the concatenation of two vectors.

Finally, the prediction of user u for the t-th bonus treatment is formulated as:

$$\hat{y}_t = \begin{cases} \hat{y}_1, & if\ t = 1 \\ \hat{y}_1 + \sum_{k=2}^{t} \hat{\Delta}_k, & if\ t > 1. \end{cases} \tag{16}$$

To implement the monotonic constraint, we only need to make the incremental effect $\hat{\Delta}_k$ greater than 0. It is easily achieved by squaring the raw output of the last head layer. The loss function of the proposed model is defined as follows:

$$\mathcal{L}_{predict} = -\frac{1}{N} \sum_{(u,y)\in\mathcal{D}}^{N} (ylog\hat{y}_t + (1-y)log(1-\hat{y}_t)), \tag{17}$$

Where $u = (x, c, t)$ is the feature tuple, y is the label indicating whether to convert. N is the number of samples in the entire sample space $\mathcal{D}$.

3.3.2 Convex constrained.

Proposition 3.1. *Given b^l and b^u, there exists $b^l \le b_j \le b^u, \forall j \in [M]$, then the optimal value in Eq.(1) will increase, when adding new bonus candidates.*

Proposition 3.2. *For the sub-problem (11), let $C_i = \{c_{i,j} | c_{i,j} = p_{i,j} * b_j\}$ represents the expected cost set. Sort the $(c_{i,j}, p_{i,j})$ pairs in increasing order of $c_{i,j}$, then only pairs that are monotonically increasing on the convex hull can be decided.*

Combining proposition 3.1 and 3.2, in order to obtain a higher optimal value in the bonus allocation problem, we should add new optional bonuses or reduce the size of non-monotonic and non-convex cases.

Theorem 3.3. *If $y = f(x)$ is a monotonically increasing and convex function, there is $y = g(xf(x))$, then $g(\cdot)$ is also a monotonically increasing and convex function if $x > 0$.*

Thanks to theorem 3.3, we can transform the convex constraint between conversion probability and expected cost into that between conversion probability and bonus amount, which is convenient for modeling. Therefore, we propose a multi-treatment effect estimation model with monotonically increasing and convex constraints.

As shown in Figure 3 (c), the model inference process consists of two steps. First, the increment of the effect incremental slope is formulated:

$$\hat{v}'_k = f_k([\phi; \delta_k]). \tag{18}$$

We square the output of the last head layer to ensure that $\hat{v}'_k \ge 0$. and then the effect incremental slope $\hat{v}_k$ is obtained through reverse accumulation, so that it decreases monotonically with the increase of the bonus amount, satisfying the definition of a convex function.

$$\hat{v}_k = \sum_{i=k}^{M} \hat{v}'_i. \tag{19}$$

Secondly, we define the incremental effect $\hat{\Delta}_k$ between two adjacent treatments as:

$$\hat{\Delta}_k = \hat{v}_k \times \alpha_k, \tag{20}$$

Where α_k indicates the amount difference between the $(k-1)$-th and the k-th bonus treatment. Finally, the predicted outcome is obtained by accumulating the effect increments, as shown in Eq.(16).

The loss function is identical to Eq.(17). Obviously, according to Eq.(16,19), the monotonically increasing and convex constraints between the outcome $\hat{y}$ and bonus amount can be realized.

3.4 Feedback Control Module

As shown in 3.2, we iterate λ to satisfy the budget constraint based on the historical data. Unfortunately, obtaining the optimal λ in the dynamic payment platform is extremely difficult. First, uncontrollable changes in factors including customer profile, traffic, and coupon write-off rate make the distribution of the variables non-stationary. Second, the marketing campaign settings such as budget and target audience may be changed irregularly.

Feedback control, which deals with the dynamic systems from feedback and outside noise [3], is widely adopted for its robustness and effectiveness. In our scenario, to cope with the fluctuation of the dynamic bonus allocation environment, we regard the Lagrangian multiplier λ as the adjustable system input and use CAC as the performance indicator. Naturally, the constraint problem is transformed into a feedback control problem: a low λ leads to a generous bonus distribution so that CAC might be higher than the target and the budget may run out early. On the contrary, a high λ results in a stingy bonus distribution, so that CAC might be lower than the target and the budget may not be fully spent.

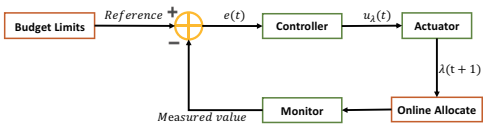


Figure 4: Framework of the feedback control module

Figure 4 presents the framework of the feedback control module. The system output follows a desired control signal, called the reference, which is preset depending on the specific task. The monitor measures the actual value of the variable under online allocation and transmits it to the controller. The controller is designed to diminish the gap between the actual measured value and the reference and outputs the control signal, which is then transformed by the actuator into the system input signal sent back to the dynamic online allocation system.

Proportional-Integral-Derivative (PID) controller [5]is the most popular feedback controller, especially in the absence of knowledge of the underlying process [6]. A PID controller continuously calculates the error $e(t)$ between the measured value $h(t)$ and the reference $r(t)$ at every time step t, and generates the control signal $\mu_\lambda(t)$ based on the combination of proportional, integral, and derivative terms of $e(t)$. The control signal $\mu_\lambda(t)$ is then sent to adjust the Lagrangian multiplier $\lambda(t)$ by the actuator model $F_\lambda(\lambda(0), \mu_\lambda(t))$. It is practical and common to use discrete time steps $(t_1, t_2, ...)$ in online advertising scenarios, so the process of PID could be formulated as following equations, where k_p, k_i, and k_d are the weight parameters of a PID controller.

$$e(t) = r(t) - h(t), \tag{21}$$

$$u_\lambda(t) = k_p e(t) + k_i \sum_{i-1} e(k) + k_d(e(t) - e(t-1)), \tag{22}$$

$$\lambda(t+1) = F_\lambda(\lambda(0), \mu_\lambda(t)). \tag{23}$$

4 EXPERIMENTS

To verify the effectiveness of the proposed framework, we conduct offline experiments and online A/B tests. We first introduce the datasets, baseline methods, and evaluation metrics. Finally, we present the experimental results and analysis.

4.1 Experimental Setup

4.1.1 Experimental Settings. To demonstrate the effectiveness of the proposed framework, our experiments consist of three parts: user intent detection evaluation part, budget allocation evaluation part, and feedback control evaluation part. We mainly conduct experiments in real industrial data, and the sample collection process is shown in Figure 2. Through an online randomized controlled trial, we collected data on 6.6 million orders and 2 million users within 11 days, including 51 optional bonus amounts. We randomly selected 5.6 million order data as the training set in the user intent detection module to train the multi-treatment effect estimation model.

4.1.2 Compared Methods.

- **Meta-learner**: Meta-learners decompose the causal problem into separate prediction problems that can be solved by standard machine learning algorithms and subsequently combined to estimate the causal parameters of interest [17]. We compare S-learner and X-learner [14] in the experiment.
- **TARNET** [20, 21]: As shown in Figure 3(a), TARNET is a general base structure for treatment effect estimation, which adopts a two-stage structure. Since we collected samples through randomized controlled trials, the regularization term $\mathcal{L}_{balance}$ considering the difference in representation distribution under different treatments was removed.
- **Semi-black box model** [27, 33]: Inspired by the demand curve in economics, the semi-black box model plays an important role in the response prediction of e-commerce. There are many kinds of demand curves, such as linear, log-linear, constant-elasticity, logit, and so on [18, 23]. Among them, the logit response curve is the most popular [19, 24, 27]. We use the acceptance rate prediction model in Meituan's multi-stage bonus allocation as a comparison method.
- **Unconstrained**: As shown in Figure 3(b), the model proposed in Section 3.3.1. The difference is that we do not enforce that the increment is non-negative to get an unconstrained model.
- **Monotonic constrained**: As shown in Figure 3(b), we square the output of the last layer in the head layers to ensure that the incremental effect is non-negative.
- **Convex constrained**: As shown in Figure 3(c), the treatment effect estimation model with convex constraint proposed in Section 3.3.2.

4.1.3 Evaluation Protocol. For the user intent detection evaluation part, in addition to adopting the evaluation metrics commonly used in classification models, such as AUC and Logloss, we also evaluate counterfactual predictions, which is more important.

- **The error of ATT estimation**: ϵ_{ATT} is a common metric for standard causal effect estimation [8]. ϵ_{ATT} can be trivially extended to multiple treatments by considering the average ATT between every possible pair of treatments.
- **Reverse pair rate (RPR)** [22]: For each order in the test set, we can get its predicted probability under each bonus treatment, and count the proportion of monotonic violations in all possible treatment pairs.
- **Upper left convex hull ratio (Ulchr)**: Based on proposition 3.2, we verify the proportion of bonus treatments that meet the monotonic and convex relationship between expected cost and conversion probability.

For the joint evaluation of the user intent detection module and allocation module, metrics are as follows:

- **The average expected response (AER)**: From [34], we estimate the expected response in the entire feature space by the law of total expectation, which was proved to be an unbiased estimate of the expected response in the randomized controlled trial setting.

4.1.4 Implementation Details. We apply LightGBM [1] as base-learner in meta-learner methods, the learning rate, feature fraction, bagging fraction, bagging frequency, max bin, boosting type are set as 0.1, 0.8, 0.8, 5, 100, gbdt, respectively.

For the neural network-based models, in order to evaluate the generalization ability and the fairness of the comparison, TARNET and our proposed models adopt the same hyperparameters, the shared layers are single-layer MLP with dimension 128, and the head layers are three-layer MLP with dimension [64, 32, 1]. In the semi-black box model, hidden layer 0 is a single-layer MLP with dimension 128, and hidden layer 1 and hidden layer 2 are three-layer MLPs with dimension [64, 32, 1]. In the experiment, we train the neural network with SGD using the Adam Optimizer [13] with the hyperparameters of $\epsilon = 0.0001, \beta_1 = 0.9, \beta_2 = 0.999$. We conduct experiments with NVIDIA Tesla V100 GPU with 16G memory.

4.2 User Intent Detection Evaluation

The intent detection results are shown in Table 1. From these results, we have the following insightful findings.

- In terms of AUC and Logloss, the evaluation metrics for factual predictions, the proposed monotonic constrained model and unconstrained model have nearly identical optimal performance, achieving 0.0010 and 0.0007 AUC gains compared to the best semi-black box model in the baseline methods respectively. The proposed convex-constrained model has the same AUC as the semi-black box model, but the Logloss is 0.00014 less.
- Regarding the effect estimation evaluation metrics ϵ_{mATT}, the monotonic constrained model outperforms others, and compared with the best method X-learner model in the baselines, its metric ϵ_{mATT} is relatively reduced by 8.44%. However, the proposed convex constrained model performs worse than the compared methods S-learner and X-learner. Obviously, neither of the semi-black box and TARNET models is a clear winner.
- Observing the RPR and Ulchr metrics, our proposed model strictly satisfies the constraints. In baseline methods, the S-learner method violates the monotonic assumption slightly, and the semi-black box model has more decision-making bonus treatments.
- The performance of the methods of predicting factual outcomes and estimating counterfactual effects is not always consistent. For example, the semi-black box model has a high AUC, but the ϵ_{mATT} performs poorly.
- In the meta-algorithms, it is more counter-intuitive that X-learner performs worse than the S-learner according to the experimental results. This is because X-learner was proposed to solve the situation when the number of units in one treatment group is much larger than in the other. However, in our RCTs, the number of samples per treatment is almost the same, so simpler models such as S-learner generalize better.
- Our proposed monotonic constrained model and convex constrained model outperform TARNET by a significant margin of about 62.23% and 58.08%, respectively, in terms of ϵ_{mATT}. This result confirms the rationality of the assumptions and the superiority of the proposed methods. Additionally, we believe that the good performance is due to the more complete fusion of information in the head layers through the accumulation of outputs, as evidenced by the comparison between TARNET and the proposed unconstrained model. During the training of the i-th treatment sample, the first $i-1$ head layers are also involved in the training process.

Table 1: Results of different intent detection methods.

Method	AUC↑	Logloss↓	ϵ_{mATT} ↓	RPR ↓	Ulchr ↑
S-learner	0.9157	0.05908	0.000868	6.21%	14.39%
X-learner	0.9111	0.06215	0.000865	15.71%	14.57%
Semi-black box	0.9163	0.05885	0.002054	**0.0%**	92.31%
TARNET	0.9096	0.06010	0.002097	40.33%	10.35%
Unconstrained	0.9170	0.05866	0.000976	29.46%	18.90%
Monotonic	**0.9173**	**0.05863**	**0.000792**	**0.0%**	16.16%
Convex	0.9163	0.05871	0.000879	**0.0%**	**100.0%**

In terms of AER, since there were not enough positive samples matched, the metric fluctuated greatly, so we performed regression processing, and the results are shown in Figure 5. We have the following observations:

- The proposed monotonic constrained model outperforms other compared methods, however, the convex constrained model is inferior to the S-learner method;
- The metrics AER and ϵ_{mATT} are approximately consistent.

From the experimental results, we found that the convex constrained model does not perform well in estimating the treatment

[1] https://github.com/microsoft/LightGBM

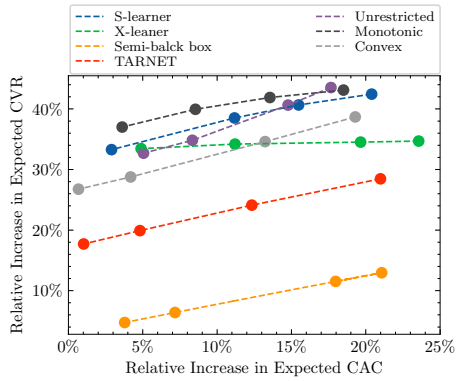


Figure 5: The average expected response metrics of the methods under multiple budget constraints. The X-axis represents the relative increase in cost compared to a given baseline, and the Y-axis represents that of CVR.

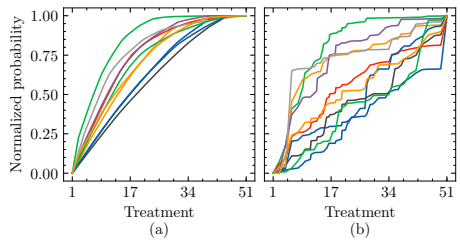


Figure 6: For the case study, we randomly pick ten users and observe their conversion probabilities for each bonus treatment predicted by the model. (a) comes from the convex constrained model, (b) comes from the monotonic constrained model. In addition, in order to facilitate the observation of the difference between user response curves, the probability is min-max normalization.

effect. As shown in Figure 6, we randomly selected 10 users from the test set and observed their normalized conversion probabilities for each bonus treatment. It can be observed that the predictions of the models fully satisfy the respective constraints. However, because the convex constraints are too strict, the predicted user response trends are relatively consistent and lack fitting ability. Even if it is proved that the convex constraint can improve the upper bound of the optimal value, it is necessary to consider the damage caused by the strict constraint to the estimation of the treatment effect. Although the convex constraints are too strong in our scenario, we believe that the convex constrained model or its weakened version may work in other scenarios with diminishing marginal gains.

4.3 Bonus Allocation Optimization Evaluation

We evaluate the optimality ratio between the Lagrangian dual solution against the LP solution (i.e., the upper bound). In our optimization problem, there are millions of users and dozens of bonuses. Since the current LP solver is difficult to handle such a large-scale problem [32], We formulate some modest-sized problems by sampling customers and bonuses for experimentation. We adopt the SCIP solver from the popular Google OR-tools LP tool, which is a framework for constraint integer programming and branch-cut-and-price.

We test the optimality ratios for $|user| \in \{5k, 10k, 20k, 30k\}$ across $|bonuses| \in \{5, 15, 30, 51\}$. The results show that the optimality ratio is above 99.99%, and the gap is negligible, with a maximum of 5.4×10^{-16}.

4.4 Feedback Control Evaluation

4.4.1 Implementation Details. We adopt the linear model actuator [31] shown in Eq.(24), where $\mu_\lambda(t)$ is derived by Eq.(22) in the PID controller and λ is restricted between λ_{min} and λ_{max} to guarantee the system robustness. In addition, there are two key points in practical applications. First, the reference is frequently modified for practical reasons. Second, we actually care about the accumulated CAC when the campaign ends, and there is a significant difference in real-time CAC of each time step contributing to the accumulated CAC. Therefore, we modify the error $e(t)$ as shown in Eq.(25,26), where $r(t)$ and $h(t)$ respectively refer to the reference accumulate CAC and the measured accumulated CAC, $conversion(i)$ represents the number of conversions in the time step i. By such a modification, the PID controller would constantly increase its attention on the accumulated CAC and can be adapted to scenarios where the reference is frequently modified.

$$\lambda(t+1) = \lambda(t)(1 + \mu_\lambda(t)), \tag{24}$$

$$e(t) = \frac{r(t) - h(t)}{r(t)}, \tag{25}$$

$$h(t) = \frac{\sum_{i=1}^{t} conversion(i) \cdot CAC_{measured}(i)}{\sum_{i=1}^{t} conversion(i)}. \tag{26}$$

To determine the weight parameters of a PID controller, we use the grid-search method to find the best setting on the training dataset and apply it to the test dataset. We regard every hour as a time step, so the maximum time steps T equals 24.

4.4.2 Evaluation Protocol. The goal of the feedback control module is to make the actual daily CAC approach the set reference value by adjusting the Lagrangian multiplier.

We verify the effectiveness of the feedback control strategy from the following three aspects.

- **Accuracy**: As shown in Figure 7, we observe the relative error of the feedback control strategy for each day within one month. The minimum absolute value is 0.026%, and the maximum value is 5.35% on 12-16. This is because the reference value changed at 19:00 on that day, and the feedback control is too late to adjust. The average absolute relative error for the whole month is 0.84%, which satisfies business requirements well.

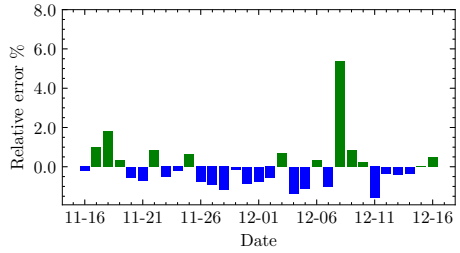


Figure 7: The relative error of feedback control per day, defined as $(CAC_{measured} - CAC_{reference})/CAC_{reference}$

Table 2: Competitive analysis of solutions.

	Lagrangian multiplier λ	Optimal value
Offline	3.66×10^{-5}	4085.56
Online	3.59×10^{-5} *	4084.95

* The mean value regulated by feedback control.

- **Efficiency**: Usually, it takes three days to adjust the online CAC to the reference value manually. However, the feedback control is in place on the same day, and the efficiency has increased by 300%.
- **Optimality**: We perform a competitive analysis for online decision algorithms and feedback control strategies. As shown in Table 2, aligned with the budget limits of the online solution, we solve the optimal policy offline and then compare the optimal values obtained in the online and offline scenarios. The Online demonstrates a near-optimal performance with a negligible optimality gap up to 1.5×10^{-4}.

4.5 Performance of Online A/B Tests

Considering the negative impact on the business and offline evaluation results, we did not deploy all the baseline methods online. With the development of our business, we have successively deployed S-learner, the monotonic constrained model, and the convex constrained model to the online system. Each model serves millions of orders every day. For a fair comparison, we control the orders and the new customers involved in the A/B testing to be homogeneous.

From table 3, we see that our proposed monotonic constrained model achieves a significant -5.07% reduction in CAC with almost the same CVR (-0.44%) compared with the online mainstream S-learner model. In addition, compared with the S-learner model, the CAC of the proposed convex constrained model is reduced by 6.62%, and the conversion rate is reduced by 2.59%. The results of the online A/B experiment are basically consistent with the offline analysis.

5 RELATED WORK

Bonus allocation (or coupons, budgets, discounts, etc.) is a typical problem in the field of intelligent marketing. By issuing incentives to users to promote user conversion, in order to achieve business goals. Mainstream solutions often include two stages, a combination of machine learning and operations research. The first stage is user intent detection, and the second stage is to solve the assigned optimization problem.

Table 3: Online A/B test results.

Method	Gain	
	CVR	CAC
Monotonic constrained vs S-learner	-0.44% (p-value=0.68)	-5.07% (0.0082)
Convex constrained vs S-learner	-2.59% (0.0327)	-6.62% (<0.001)

There are some studies [2, 9, 15, 22, 27, 33] on marketing allocation. [33] propose a unified framework in which the user intent detection part extends the traditional logit response model to a semi-black-box model. [27] further improve the method to a two-tower structure, corresponding to fitting two parameters in the logit model. [15] selects the best model from a number of uplift modeling techniques such as two-models, transformed outcome, and X-learner. [9] leverages Retrospective Estimation, is a modeling approach that relies exclusively on data from positive outcome examples. In addition, user intent detection is a multi-treatment effect estimation problem, including tree-based [1, 4, 25] and representation learning methods [10, 16, 20, 21, 30]. It has come to our attention that there is a similar approach being used for user intent detection with monotonic constraints [22]. However, our method differs in that we utilize an end-to-end training approach and a parallel multi-head network structure.

As one of the most effective methods, the Lagrangian dual method is widely deployed in the second stage for online decision-making [1, 15, 27, 33]. [2] provides a real-time adaptive method and extends the existing literature by addressing cases with negative weights and values. To address the dynamic environment in marketing, [27] apply some simple periodic control strategies, which require additional workload and at the expense of efficiency. We took advantage of the feedback control theory, which has been proven effective in advertising display by work of [12, 28, 31] and adapted to the marketing scenario after delicate improvement.

6 CONCLUSION

In this paper, we formulate the online bonus allocation problem as a MCKP. To overcome its challenges, we propose a framework consisting of three modules: User Intent Detection module, Online Bonus Allocation module, and Feedback Control module. In the User Intent Detection module, in order to overcome the non-monotonic problem of the response curve, we propose a multi-treatment effect prediction model with monotonic constraints. To reduce the optimality gap between user intent detection and optimal allocation stages, a convex constrained detection model is further proposed. Facing the challenge that the distribution of online orders and users changes over time, we leverage the feedback control strategy to make online costs more accurately approach the budget limit. Offline and online experimental results demonstrate significant improvement compared with baseline methods.

REFERENCES

[1] Meng Ai, Biao Li, Heyang Gong, Qingwei Yu, Shengjie Xue, Yuan Zhang, Yunzhou Zhang, and Peng Jiang. 2022. LBCF: A Large-Scale Budget-Constrained Causal Forest Algorithm. In *Proceedings of the ACM Web Conference 2022*. 2310–2319.
[2] Javier Albert and Dmitri Goldenberg. 2022. E-Commerce Promotions Personalization via Online Multiple-Choice Knapsack with Uplift Modeling. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2863–2872.
[3] Karl Johan Åström and Panganamala Ramana Kumar. 2014. Control: A perspective. *Autom.* 50, 1 (2014), 3–43.
[4] Susan Athey, Julie Tibshirani, and Stefan Wager. 2019. Generalized random forests. (2019).
[5] Stuart Bennett. 1993. Development of the PID controller. *IEEE Control Systems Magazine* 13, 6 (1993), 58–62.
[6] Stuart Bennett. 1993. *A history of control engineering, 1930-1955*. Number 47. IET.
[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.
[8] Lu Cheng, Ruocheng Guo, Raha Moraffah, Paras Sheth, Kasim Selcuk Candan, and Huan Liu. 2022. Evaluation methods and measures for causal learning algorithms. *IEEE Transactions on Artificial Intelligence* (2022).
[9] Dmitri Goldenberg, Javier Albert, Lucas Bernardi, and Pablo Estevez. 2020. Free lunch! retrospective uplift modeling for dynamic promotions recommendation within roi constraints. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 486–491.
[10] Fredrik Johansson, Uri Shalit, and David Sontag. 2016. Learning representations for counterfactual inference. In *International conference on machine learning*. PMLR, 3020–3029.
[11] Daniel Kahneman and Amos Tversky. 2013. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, 99–127.
[12] Niklas Karlsson and Jianlong Zhang. 2013. Applications of feedback control in online advertising. In *2013 American control conference*. IEEE, 6008–6013.
[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[14] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences* 116, 10 (2019), 4156–4165.
[15] Liangwei Li, Liucheng Sun, Chenwei Weng, Chengfu Huo, and Weijun Ren. 2020. Spending money wisely: Online electronic coupon allocation based on real-time user intent detection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2597–2604.
[16] Belbahri Mouloud, Gandouet Olivier, and Kazma Ghaith. 2020. Adapting neural networks for uplift models. *arXiv preprint arXiv:2011.00041* (2020).
[17] Gabriel Okasa. 2022. Meta-Learners for Estimation of Causal Effects: Finite Sample Cross-Fit Performance. *arXiv preprint arXiv:2201.12692* (2022).
[18] Robert L Phillips. 2021. Pricing and revenue optimization. In *Pricing and Revenue Optimization*. Stanford university press.
[19] Huashuai Qu, Ilya O Ryzhov, and Michael C Fu. 2013. Learning logistic demand curves in business-to-business pricing. In *2013 Winter Simulations Conference (WSC)*. IEEE, 29–40.
[20] Patrick Schwab, Lorenz Linhardt, and Walter Karlen. 2018. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656* (2018).
[21] Uri Shalit, Fredrik D Johansson, and David Sontag. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*. PMLR, 3076–3085.
[22] Yitao Shen, Yue Wang, Xingyu Lu, Feng Qi, Jia Yan, Yixiang Mu, Yao Yang, YiFan Peng, and Jinjie Gu. 2021. A framework for massive scale personalized promotion. *arXiv preprint arXiv:2108.12100* (2021).
[23] Kalyan T Talluri, Garrett Van Ryzin, and Garrett Van Ryzin. 2004. *The theory and practice of revenue management*. Vol. 1. Springer.
[24] Ruben van de Geer, Arnoud V den Boer, Christopher Bayliss, Christine SM Currie, Andria Ellina, Malte Esders, Alwin Haensel, Xiao Lei, Kyle DS Maclean, Antonio Martinez-Sykora, et al. 2019. Dynamic pricing and learning with competition: insights from the dynamic pricing challenge at the 2017 INFORMS RM & pricing conference. *Journal of Revenue and Pricing Management* 18, 3 (2019), 185–203.
[25] Stefan Wager and Susan Athey. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *J. Amer. Statist. Assoc.* 113, 523 (2018), 1228–1242.
[26] Stephen J Wright. 2015. Coordinate descent algorithms. *Mathematical Programming* 151, 1 (2015), 3–34.
[27] Zhuolin Wu, Li Wang, Fangsheng Huang, Linjun Zhou, Yu Song, Chengpeng Ye, Pengyu Nie, Hao Ren, Jinghua Hao, Renqing He, and Zhizhao Sun. 2022. A Framework for Multi-Stage Bonus Allocation in Meal Delivery Platform. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 4195–4203. https://doi.org/10.1145/3534678.3539202
[28] Xun Yang, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, and Kun Gai. 2019. Bid optimization by multivariable control in display advertising. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1966–1974.
[29] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 5 (2021), 1–46.
[30] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. 2018. Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems* 31 (2018).
[31] Weinan Zhang, Yifei Rong, Jun Wang, Tianchi Zhu, and Xiaofan Wang. 2016. Feedback control of real-time display advertising. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 407–416.
[32] Xingwen Zhang, Feng Qi, Zhigang Hua, and Shuang Yang. 2020. Solving Billion-Scale Knapsack Problems. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 3105–3111. https://doi.org/10.1145/3366423.3380084
[33] Kui Zhao, Junhao Hua, Ling Yan, Qi Zhang, Huan Xu, and Cheng Yang. 2019. A unified framework for marketing budget allocation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1820–1830.
[34] Yan Zhao, Xiao Fang, and David Simchi-Levi. 2017. Uplift modeling with multiple treatments and general response types. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 588–596.

A APPENDIX

A.1 Proofs

A.1.1 Proposition 3.1. Given b^l and b^u, there exists $b^l \le b_j \le b^u, \forall j \in [M]$, then the optimal value in Eq.(1) will increase, when adding new bonus candidates.

Proof. Under the Lagrangian relaxation paradigm, we prove that the greedy strategy Eq.(13) is optimal, indicating that for a given λ, there is no better strategy $\mathcal{X}^*$ such that when the cost increase ΔC, the optimal value increment ΔV is greater than $\lambda * \Delta C$. It can be easily proved by contradiction, assuming that there is at least one customer i, under the more optimal strategy, the allocation result changes from $x_{i,j} = 1$ to $x_{i,j^*} = 1$. Since $\Delta V(i) > \lambda * \Delta C(i)$, it yields $p_{i,j^*} - \lambda p_{i,j^*} b_{j^*} > p_{i,j} - \lambda p_{i,j} b_j$, which conflicts with Eq.(13). However, when new bonus candidates are added, the solution space expands. Since N is very large, according to the optimal greedy allocation, some customers are assigned to new bonus candidates. This part of the group has $\Delta V > \lambda * \Delta C$, leading to an increase in the optimal value. Figure 8 shows the relationship between the number of bonuses and the optimal value in our actual business. Thus, we complete the proof. □

A.1.2 Proposition 3.2. For the sub-problem (11), let $C_i = \{c_{i,j} | c_{i,j} = p_{i,j} * b_j\}$ represents the expected cost set. Sort the $(c_{i,j}, p_{i,j})$ pairs in increasing order of $c_{i,j}$, then only pairs that are monotonically increasing on the convex hull can be decided.

Proof. As shown in Figure 9, we discuss non-monotonic and non-convex case by case. First, for the non-monotonic increasing point $(c_{i,d}, p_{i,d})$, it exists $p_{i,d} - p_{i,c} < 0 < c_{i,d} - c_{i,c}$. We assume that $c_{i,b}$ is assigned to user i under the greedy allocation strategy Eq.(13). Then it can yields $p_{i,d} - \lambda c_{i,d} \ge p_{i,c} - \lambda c_{i,c}$, indicating that $p_{i,d} - p_{i,c} \ge \lambda(c_{i,d} - c_{i,c})$, which contradicts the truth of non-monotonic increasing. For the non-convex point $(c_{i,b}, p_{i,b})$, we also

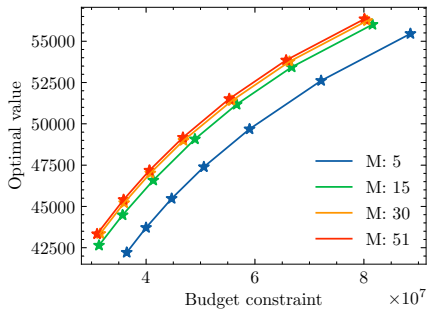


Figure 8: The X-axis denotes budget constraint B, and Y-axis denotes the optimal value obtained by using the Lagrangian dual method. As the number of bonus M increases, the optimal value obtained within the same budget constraint will increase marginally.

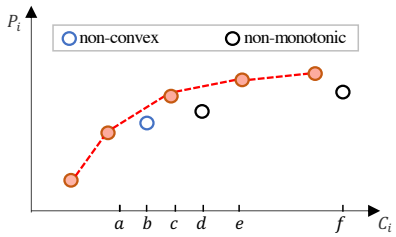


Figure 9: A toy example to assist in proving proposition 3.2. The red dotted line indicates the convex hull.

assume that it is assigned, indicating that:

$$p_{i,b} - \lambda c_{i,b} \geq p_{i,a} - \lambda c_{i,a}, \tag{27}$$

$$p_{i,b} - \lambda c_{i,b} \geq p_{i,c} - \lambda c_{i,c}, \tag{28}$$

Further, it yields:

$$\begin{vmatrix} p_{i,b} - p_{i,a} & p_{i,c} - p_{i,b} \\ c_{i,b} - c_{i,a} & c_{i,c} - c_{i,b} \end{vmatrix} \geq 0,$$

which contradicts the definition of non-convex. Thus, we complete the proof. □

A.1.3 Theorem 3.3. If $y = f(x)$ is a monotonically increasing and convex function, there is $y = g(xf(x))$, then $g(\cdot)$ is also a monotonically increasing and convex function if $x > 0$.

Proof. $y = f(x)$ yields $x = f^{-1}(y)$, thus $g^{-1}(y) = yf^{-1}(y)$. Derivation of f^{-1} is as follows:

$$\left(f^{-1}\right)' = \frac{1}{f^{-1}}, \tag{29}$$

$$\left(f^{-1}\right)'' = -\frac{f''}{(f')^3}, \tag{30}$$

Where $'$ represents the first derivative and $''$ denotes the second derivative.

Since f is a convex function, according to the definition of the convex function, it can be deduced that $f'' \leq 0$. It also yields $f' > 0$ from the definition of monotonically increasing. Therefore, it further yields $\left(f^{-1}\right)'' \geq 0$ described in Eq.(30).

Let $h(y) = yf^{-1}(y)$, the derivation is as follows:

$$h' = y \cdot (f^{-1})' + f^{-1}, \tag{31}$$

$$h'' = y \cdot (f^{-1})'' + 2 \cdot (f^{-1})'. \tag{32}$$

Since $x = f^{-1}(y) > 0$, it can yields $h' > 0$ and $h'' > 0$. Therefore, the function h, that is, the function g^{-1} is a monotonically increasing and concave function, It can be further proved that $g(\cdot) = \left(g^{-1}(\cdot)\right)^{-1}$ is a monotonically increasing and convex function by repeating the above steps Eq.(29,30). Thus, we complete the proof. □

C-AOI- Contour-based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform

Yida Zhu
zhuyida@meituan.com
Meituan
Beijing, China

Liying Chen
chenliying03@meituan.com
Meituan
Beijing, China

Daping Xiong
xiongdaping@meituan.com
Meituan
Beijing, China

Shuiping Chen*
chenshuiping@meituan.com
Meituan
Beijing, China

Fangxiao Du
dufangxiao@meituan.com
Meituan
Beijing, China

Jinghua Hao
haojinghua@meituan.com
Meituan
Beijing, China

Renqing He
herenqing@meituan.com
Meituan
Beijing, China

Zhizhao Sun
sunzhizhao@meituan.com
Meituan
Beijing, China

ABSTRACT

Online food delivery (OFD) services have become popular globally, serving people's daily needs. Precise area-of-interest (AOI) boundaries help OFD platforms determine customers' exact locations, which is crucial for maintaining consistency in delivery difficulty and providing a uniform customer experience within an AOI. Existing AOI generation methods primarily rely on predefined shapes or density-based clustering, which limits the quality of the contours. Recently, Meituan has treated the AOI contours as a binary semantic segmentation problem. Their approach involves a multi-step post-process to address the issues with boundary breaks caused by semantic segmentation models, leading to decreased quality and inefficiency in the learning process. In this paper, we propose a novel method for AOI contour generation called C-AOI (Contour-based Area-of-Interest). C-AOI is an instance segmentation model that focuses on generating high-quality AOI contours. Unlike the former method, which relies on pixel-by-pixel classification, C-AOI starts from the center point of the AOI and regresses the boundary. This approach results in a higher-quality boundary and is less computationally intensive. C-AOI first corrects errors on the contour using a local aggregation mechanism. Then, we propose a novel deforming module called the contour transformer, which captures the global geometry of the object. To enhance the positional relationship among vertices, we introduce a learnable cyclic positional encoding applied to the contour transformer. Finally, to improve the boundary details, we propose the Adaptive Matching Loss (AML) that eliminates over-smoothed boundaries and promotes optimized convergence pathways. Experimental results on real-world datasets collected from Meituan have demonstrated that C-AOI significantly improves the mask and boundary quality compared to Meituan's previous work. Moreover, Its inference speed is comparable to that of E2EC, a state-of-the-art real-time contour-based method. It is noteworthy that C-AOI has been deployed in the Meituan platform for producing AOIs.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

areas-of-interest (AOIs); contour-based instance segmentation; online food delivery (OFD); multimodal data

ACM Reference Format:
Yida Zhu, Liying Chen, Daping Xiong, Shuiping Chen, Fangxiao Du, Jinghua Hao, Renqing He, and Zhizhao Sun. 2023. C-AOI: Contour-based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599786

1 INTRODUCTION

Over the past decade, online food delivery (OFD) services have seen tremendous growth globally [11]. As per the latest statistics, the online food delivery market size was valued at $189.70 billion in 2021 and is estimated to grow at a compound annual growth rate (CAGR) of 10.8% from 2022 to 2028 [24].

In the context of OFD platforms, area-of-interest (AOI), also known as region-of-interest (ROI), refers to a geospatial unit with a defined shape in the form of a polygon and a precise spatial boundary [13]. For instance, an AOI can be an apartment complex, a shopping plaza, an office park, etc. As shown in Fig.1, an example of an AOI named *Jiangfu apartment complex* is depicted, where 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 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.
KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599786

spatial boundary is represented in purple color and nine residential buildings are covered.



Figure 1: An illustration of the AOI in OFD platforms.

AOI plays a crucial role in the business operations strategy [14, 29] of OFD platforms, as it integrates various areas that attract customer attention and activities. To illustrate the importance of AOI, consider the scenario of a customer placing an order. As shown in Fig.2, when a customer begins the ordering process, the OFD platform typically performs the following steps: 1) The platform retrieves the GPS position of the customer and, using the geographical boundaries of known AOIs, suggests possible delivery addresses around the matched AOI; 2) If the customer accepts the recommended AOI, the AOI is considered accurate; otherwise, the modified address and positioning points are sent to the geospatial database to identify potential AOIs; 3) If the customer's position is outside any known AOI, the customer needs to manually type in the address, which, along with the geospatial positioning data, can be used to generate potential AOIs.

As a result, there is a pressing need for more accurate and efficient methods for generating AOIs in OFD platforms. The precise AOI boundaries is crucial in these platforms as it affects the ability to accurately locate customers, leading to consistent delivery

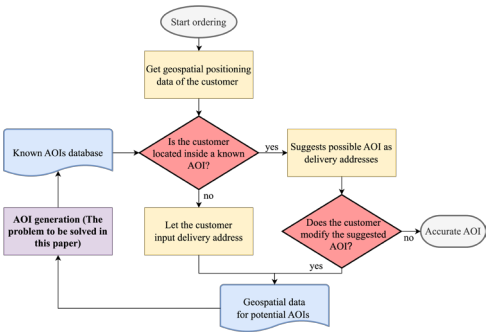


Figure 2: The role of the AOI in the ordering process in OFD platforms.

difficulty and a uniform customer experience across different buildings within the same AOI. For instance, a fast-food chain such as McDonald's may define its service area using a list of AOIs [20]. If the boundary of the *Jiangfu apartment complex* is inaccurately defined, excluding Building #1, then the OFD platform would not allow orders from this McDonald's to customers in Building #1, even though other buildings within the AOI could still place orders.

Most existing studies on AOI generation have relied on volunteered geographic information, such as geo-tagged photos [12, 25] and reviews [3, 19, 27], to group user location points with similar names and generate AOIs. However, these methods tend to produce rough boundaries, either through manual design of shapes like circles or rectangles, or by creating irregular convex hulls. These limitations present practical challenges for OFD platforms, which have typically relied on manual annotation to generate AOIs, a process that is both time-consuming and costly.

The recent trend in OFD platforms has been to address the problem of generating accurate AOI boundaries through the use of computer vision techniques and the exploitation of multimodal geospatial data generated during the ordering process. For example, Meituan, one of the largest OFD platforms in the world, proposed a framework [5] for AOI generation using image semantic segmentation. However, this framework suffers from the complexity of the AOI instance, which can contain various elements such as roads, buildings, and green plants. As a result, the pixel-wise semantic segmentation approach causes a loss of connectivity at the boundaries of the AOI instance. To overcome this challenge, the framework requires the sequential execution of three algorithms, which leads to a lower upper bound in terms of efficiency and accuracy. While this framework has demonstrated improved performance compared to other methods, its upper bound is limited due to the accumulation of errors from multiple models.

This study addresses the challenge of high-quality AOI generation in OFD platforms by proposing an end-to-end contour-based approach. Contour-based methods treat instance segmentation as a regression problem and predict the vertex coordinates of a contour represented by discrete points [34]. However, existing methods have limitations, such as regression of contour vertices along a specific direction [8, 30] or use of hand-crafted initial contour shapes [21, 23], leading to prediction errors. The E2EC [32] approach improves upon these limitations by using an adaptive deformation direction and regressing the complete contour based on the center point. However, E2EC still falls short in considering the global geometry relation among vertices on the contour. Furthermore, current contour-based methods have an inadequate pairing mechanism between ground truth and predicted vertices, leading to over-smoothed boundaries and incorrect predictions.

In this study, we present a novel and efficient instance segmentation model, C-AOI, for generating high-quality AOI from multimodal geospatial data on OFD platforms. Our approach replaces pixel-by-pixel classification with regression of a series of discrete vertices on the AOI contour, resulting in a higher quality boundary upper bound and more computational efficiency. Unlike existing contour-based methods, C-AOI adaptively learns the positional relationships between contour vertices to capture the global geometry of the object, and optimizes the convergence path of the

vertex regression problem. The key contributions of this paper are summarized as follows:

- To the best of our knowledge, we have conducted pioneering research on high-quality AOI generation by the contour-based method using geospatial data from OFD platforms. Our proposed model, C-AOI, has a more rational deformation path and is comprised of three innovative components: 1) a contour transformer architecture; 2) a learnable cyclic positional encoding; and 3) an adaptive matching loss (AML) function.
- The contour transformer is unique in that it captures the overall geometry of the object and examines the correlation of features among contour vertices from a global perspective. To further enhance the model's ability to capture the relative positional relationships of the vertices, we introduce a learnable cyclic positional encoding.
- AML adaptively calculates the optimal matching vertex set between the prediction and the ground truth. By prioritizing inflection vertex matching, AML eliminates over-smoothing of the boundary. Additionally, AML decouples the vertex regression problem into angular offset and regression distance, which promotes optimized convergence pathways.
- The performance of C-AOI was evaluated on real-world datasets collected from the Meituan platform, and showed state-of-the-art performance in terms of AOI mask and contour accuracy, with 13 times faster inference speed compared to Meituan's previous work. Our model outperforms the existing E2EC method in terms of AOI quality while maintaining the approximate inference speed. Furthermore, C-AOI is currently being used in the production of AOI on the Meituan platform.

2 RELATED WORK

2.1 AOI Generation

In prior research on the generation of AOI, various geospatial data sources such as geo-tagged photos [12, 18, 25], geo-tagged reviews [3, 19, 27], and delivery data [5, 13] have been employed. [25] extracts popular tourist routes from geo-tagged photos and represents the trajectory of coordinates as a series of AOIs using circles. In a study of trajectory pattern mining [1], rectangles are used to represent AOIs. However, the fixed contour shapes used in these methods limit their ability to generate polygonal AOIs. Density-based clustering methods, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), have been applied to address this issue [26]. The boundary of a cluster is often constructed using a convex hull after clustering, and the researchers of [18] have proposed a method to enhance the secondary density of road network sites to generate AOIs from geo-tagged images. In addition to convex hulls, the alpha-shape method [9] has been utilized to generate concave hulls for AOIs [12]. However, both convex hulls and concave hulls are rough AOI contours with significant limitations in practical applications. A polygon-based approach [10] has also been proposed for matching road networks with authority region data to generate AOI boundaries. However, the application of this road matching-based method is limited in cases where the AOI boundary is far from the road network or the road network data is incomplete. As a result, OFD platforms have long resorted to labor-intensive and time-consuming manual annotation to generate AOIs.

Recently, in efforts to save on labor labeling costs, OFD platforms such as Meituan have dedicated considerable efforts to the development of algorithms for the automatic generation of AOIs. To this end, Meituan [5] proposes a method that incorporates multi-modal features, including satellite images, road networks, and order delivery data, into a binary semantic segmentation classification problem for defining AOI boundaries. Despite yielding improved experimental results as compared to prior methods, the pixel-wise semantic segmentation approach adopted in this framework results in the loss of connectivity at AOI boundaries. To overcome this limitation, the framework requires the sequential execution of three algorithms to address boundary breaks caused by semantic segmentation models. However, this sequential execution leads to the accumulation of prediction errors, thus lowering the upper bound and rendering the method inefficient. To address these issues, the proposed C-AOI method replaces the pixel-by-pixel classification approach with the regression of a series of discrete vertices on the AOI contour, eliminating the challenge of loss of connectivity and resulting in a more effective and less computationally intensive solution.

2.2 Contour-based Instance Segmentation

Recently, there has been a resurgence of interest in contour-based instance segmentation methods, due to their demonstrated potential for improved accuracy and efficiency compared to mask-based methods. The majority of these methods formulate the instance as a polygon and propose various techniques for regressing the contour vertex coordinates. For example, Curve GCN [17] employs Graph Convolution Networks to iteratively deform an ellipse contour to attain the final instance contour, while Deep Snake [23] leverages circular convolution to extract features from the contour and regresses from an octagonal polygon. DANCE [21] improves the contour evolution process in the Deep Snake pipeline and introduces an edge attention module.

However, the fixed initial contour shape in these methods can result in many mismatched vertex pairs between predictions and ground truth. ESESeg [31], PolarMask [30], LSNet [8], and E2EC [32] attempt to address this issue by directly regressing the contour vertices based on the the center point of the instance. While ESESeg and PolarMask model the instance in polar coordinates, their approaches can lead to poor segmentation quality in complex polygons, as rays may cross the contour multiple times. LSNet and E2EC introduce a learnable contour initialization strategy to maintain the upper bound of performance. Despite achieving state-of-the-art results in contour-based instance segmentation, E2EC lacks the capability to capture the global geometric relationships between instances during contour deformation. Furthermore, the current contour-based methods in instance segmentation have been observed to exhibit some limitations in pairing the ground truth and predicted vertices. These limitations stem from disregarding the significance of inflection points and challenging convergence pathways, leading to over-smooth boundaries and incorrect predictions. The proposed C-AOI method addresses these limitations and preserves more details of contour boundaries.

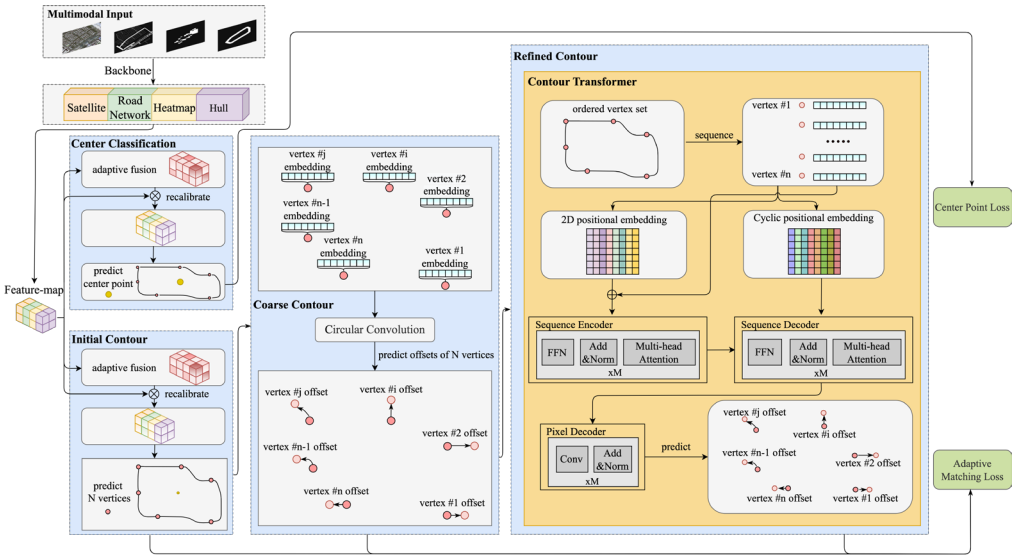


Figure 3: An overview of the proposed C-AOI framework.

3 APPROACH

In this section, we present an overview of multimodal data processing, before presenting a reformulation of the AOI boundary generation by the proposed multi-stage contour deformation approach. Subsequently, we describe a novel deformation module, the Contour Transformer, which is designed to improve the accuracy of contour refinement. Finally, we outline the design of a new Adaptive Matching Loss, which serves as an optimization tool for contour vertex regression.

3.1 Multimodal Data Processing

The use of rich multimodal information contributes to the generation of high-quality AOI, as demonstrated in Meituan's previous framework. This study follows the multimodal data processing in [5]. The pipeline for multimodal data processing is depicted in Fig. 4. During the customer ordering process, we collect both the delivery address and GPS location of the delivery point. The delivery address in the OFD platform typically include an AOI name, a POI name (e.g., building), and additional information such as room number. Utilizing Bert [6], a well-known address parsing model, we extract the AOI name from the delivery address. Additionally, we incorporate the customer's geospatial location to distinguish AOI names that may occur in different cities and regions. With the same AOI name, we generate an order heatmap by clustering based on geographic location. Specifically, we encode customer locations using GeoHash and then cluster using the DBSCAN algorithm. The order heatmap allows us to derive rough shape, the convex hull of AOI, using a convex algorithm [2]. We then obtain the road network data and satellite image data corresponding to the vicinity of the AOI by expanding its range based on the convex hull. Finally, we extract the heatmap, convex hull, and road network as grayscale images, and use them as input data for the model, along with the satellite image. The importance of multimodal data for C-AOI is validated in our ablation experiments in Section 4.2.

3.2 Multi-stage Contour Deformation

The C-AOI is an efficient end-to-end model that integrates a center localization branch and a learnable contour deformation branch. The architecture is illustrated in Fig. 3. Given an AOI mask, the process begins with identifying a candidate center (x_c, y_c) for the AOI and selecting a set of ordered vertices along the contour. These ordered vertices serve as the initial reference for the contour regression. The initial offsets $\{(\Delta x^i_{\text{init}}, \Delta y^i_{\text{init}}) \mid i = 1, 2, \ldots, N\}$ of each contour vertex relative to the center point are directly estimated, and a coarse contour is obtained by correcting the vertex offsets $\{(\Delta x^i_{\text{coa}}, \Delta y^i_{\text{coa}}) \mid i = 1, 2, \ldots, N\}$ from a local perspective. To further refine the contour, the Contour Transformer module is employed, which refines the contour $\{(\Delta x^i_{\text{ref}}, \Delta y^i_{\text{ref}}) \mid i = 1, 2, \ldots, N\}$ based on a global perspective using the coarse contour as input. In our experiments, we set $N = 128$.

Center Localization Branch. The center localization branch of C-AOI receives as input features extracted from multimodal information, with the lightweight DLA-34 serving as the backbone network in our experiments. Given the differing contributions of

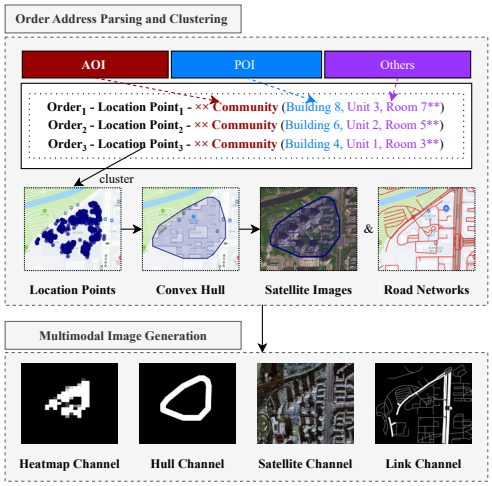


Figure 4: Pipeline of multimodal geospatial data processing.

features from different modalities to centroid classification, an adaptive recalibration attention mechanism is employed to amplify the most informative features. This mechanism consists of nonlinear transformation of the feature map through a 2-layer fully convolutional network (FCN) to generate a score map of the same size as the feature map, followed by normalization of the score map through the application of the softmax function along the channel dimension. The recalibrated features are then used for centroid classification. The point location (x, y) is considered as a positive sample if it lies in the vicinity of the center point of the AOI, or as a negative sample otherwise. Sampling positive samples is achieved through the use of a Gaussian kernel, following the method of CenterNet [33]. To balance the number of positive and negative samples during training of the center localization branch, focal loss [16] is also employed.

Initial Contour. The processing of features in the initial contour deformation module is similar to that in the center localization branch. The features are calibrated by the adaptive recalibration attention mechanism to emphasize the informative features and then used to estimate the offsets between each initial contour vertex and the center point. The initial contour vertices can be computed as the sum of the center point coordinates and the offsets. During the training phase, the true center point of each AOI is employed as the center point, while during inference, the predictions generated by the center localization branch are utilized instead. This design effectively decouples the center point detection and contour vertices regression tasks during the training process.

Coarse Contour. The direct regression of accurate contour vertices presents a challenge. To overcome this challenge, C-AOI first employs a local aggregation mechanism to correct errors in the contour representation. Specifically, the contour is represented as an ordered set of vertices and 1D circular convolution is utilized to learn features among neighboring vertices, which provides an efficient way to compute offsets by leveraging local information between the vertices.

Refined Contour. Using solely local features for contour deformation can prove to be insufficient in effectively correcting substantial prediction errors. Previous approaches, such as the global deformation module presented in E2EC, utilize simple concatenation of all vertex features and an MLP module to incorporate global information. However, the positioning of vertices on the contour is not adequately considered, which leads to a loss of the global geometry relations among the ordered vertex set, resulting in ineffective utilization of the global information and a decrease in the upper performance bound. To address this issue, C-AOI proposes a more effective global information aggregation module, referred to as the contour transformer, for refining the final contour based on the features of all contour vertices.

3.3 Contour Transformer

The proposed Contour Transformer is a novel deformation module designed to capture the global geometry of the object, as illustrated in Fig. 5. It leverages the potential similarity of features among candidate contour vertices while also considering their spatial topological relationship to enhance feature propagation along the contour. The Contour Transformer comprises three components: a sequence encoder to extract relationships between vertices, a sequence decoder to augment the global contour features of vertices, and a pixel decoder to compute the refined offsets of each vertex. Unlike other transformers applied to images [15, 22], the Contour Transformer focuses on the features near the contours rather than the entire feature maps. The ordered vertices are treated as a sequence of length N, where each vertex is considered a token in the sequence. The token-wise embedding has a dimensionality equal to the number of channels in the feature maps, obtained through

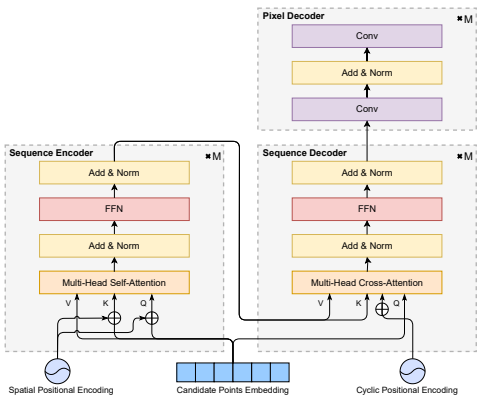


Figure 5: The detailed calculation flow of our proposed contour transformer.

the pixel indexes of each vertex. The workings of each component in the Contour Transformer will be extensively explained in the forthcoming sections. The number of encoding and decoding layers is set to three.

Sequence encoder. The features of all contour vertices are processed through the sequence encoder, along with the spatial positional encoding that is added to queries and keys at every encoder layer. The encoder layer comprises a multi-head self-attention mechanism and a feed-forward network (FFN) with residual connections. When a contour vertex is moved, its attached edges are also shifted, which further influences the position of other vertices. Thus, the positional relationships between vertices need to be taken into account when calculating attention weights between contour vertices from a global geometry perspective. To address this issue, we introduce spatial positional encoding [4] that incorporates the positional relationship between vertices in the image. The coordinates of each vertex on the feature map can be encoded along the two axes of the image by sinusoidal functions, superimposed to form the spatial positional encoding.

Sequence decoder. The sequence decoder in our approach utilizes a cross-attention mechanism to aggregate the effective deformation information from each token feature and the global contour information. Due to the permutation-invariant nature of the transformer architecture, it requires additional positional information to assist in decoding. Unlike the encoder, where the positions of the contour vertices are derived from the coarse contour and thus, the vertex coordinates are known, the positions of the refined contour vertices in the sequence decoder remain unknown. As a result, using sinusoidal functions for spatial positional encoding is not feasible. To address this issue, we propose a cyclic positional encoding that adaptively learns an embedding matrix. The sequence decoder computes the attention weight between a vertex pair as a sum of two attention scores, obtained from disentangled matrices on their features and position. The positional relationship between contour points is calculated by performing cyclic encoding in a clockwise direction and adaptively learning an embedding matrix based on the index of the table, as shown in Fig. 6.

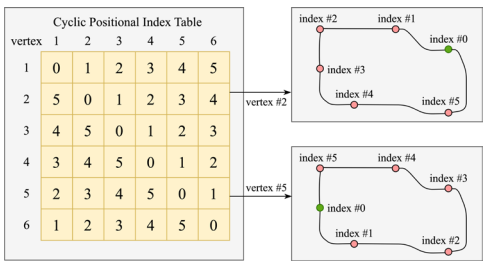


Figure 6: An example with a 6-vertex cyclic positional encoding index table.

Pixel decoder. The sequence encoder and sequence decoder utilize pair-wise relations between vertices, the model can globally reason about all the vertices while utilizing the overall contour information as contextual information. The outputs of the sequence decoder are further processed by a 3-layer 1D Convolutional Neural Network with ReLU activation and batch normalization, resulting in N final predictions, represented as pixel offsets.

3.4 Adaptive Matching Loss

The current point-pair matching approaches in the field suffer from over-smooth boundary predictions due to a lack of consideration for inflection points. While the Dynamic Matching Loss (DML) proposed by E2EC addresses this to some extent, it may still result in an uncertain regression direction when a predicted point closest to a non-inflection point is selected for regression, as demonstrated in Fig. 7. To address this issue, we propose the AML and apply it to contour deformation. In AML, the ground truth vertices are first uniformly sampled to $10N$ along the contour, and then divided into inflection vertex set $\widehat{\mathrm{Y}}$ and ordinary vertex set $\overline{\mathrm{Y}}$ by Douglas-Peucker algorithm [7]. AML prioritizes the matching of inflection vertices by minimizing the L2 distance between s^* and $\widehat{\mathrm{Y}}$ in Equation. 1. For the remaining vertices $\bar{S}$, AML uses Equation. 2 to match the nearest ordinary vertex set g^* to preserve the details of the boundary.

To derive an effective loss, we decompose the vertex regression problem into angular offset and regression distance. AML uses the center point (cp) to construct "cp-prd" vector v_p and "cp-gt" vector v_g, and calculates the cosine loss between v_p and v_g to constrain the angular offset and the Smooth-L1 loss to regress the distance in the optimal convergence direction. Equation. 3 describes the calculation of the AML loss. AML leads to the optimization of the convergence paths for vertices, resulting in an enhancement in the regression accuracy of the vertices.

$$s^* = \arg \min_{s \in S, g = \widehat{\mathrm{Y}}} \|s - g\|_2 \tag{1}$$

$$g^* = \arg \min_{s = \hat{S}, g \in \overline{\mathrm{Y}}} \|s - g\|_2 \tag{2}$$

$$L_{\mathrm{AML}}(S, Y) = L_{sm}(s^*, \widehat{\mathrm{Y}}) + L_{cos}(\widehat{v_p}, \widehat{v_g}) + L_{sm}(\bar{S}, g^*) + L_{cos}(\overline{v_p}, \overline{v_g}) \tag{3}$$

4 EXPERIMENTS

To validate the effectiveness of each component in the proposed C-AOI framework and to assess the validity of its design details, we conduct experiments on the real-world Meituan dataset. Our network is trained for 250 epochs. The weights of the backbone network (DLA-34) are initialized using pre-trained weights. The learning rate starts at 1e-3 and is decreased by 0.5 at epochs 25, 70, 125, and 180. The input images are resized to a size of 512. The performance and inference speed of all models are evaluated on a Tesla V100 GPU with 32GB of memory.

4.1 Datasets and Metrics

Datasets. In this study, real-world datasets were collected from seven Chinese cities during the period from April 1st to October 31st, 2022, by Meituan, a leading Online Food Delivery (OFD) platform. Each order contains detailed GPS coordinates and addresses of customers' locations, processed through the multimodal data processing approach to yield samples containing AOIs for detection.

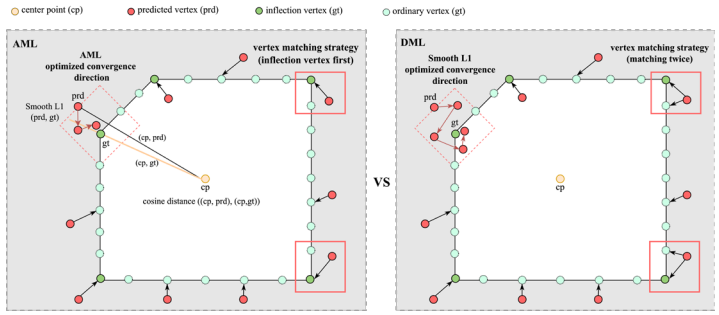


Figure 7: Comparison of AML and DML in terms of optimized convergence pathways and vertex matching strategies.

The data from Beijing were randomly divided into 80% for training and 20% for validation, while the models were evaluated on data from Xi'an, Chengdu, Nanjing, Wenzhou, Guilin, and Baoding. The characteristics of the datasets used in this study are outlined in Table. 1.

Metrics. In order to assess the quality of the generated AOIs, this study adopts the evaluation metrics used in [5]. The quality of the AOI masks is evaluated using the Average Intersection over Union (A-IOU) and Average Precision (AP). The quality of the boundaries is evaluated using the Contour Accuracy (CA). Emphasizing high-quality AOIs, this study sets the IOU threshold to 0.7 and 0.8 for AP, and the contour buffer threshold to 5 meters for CA, denoting them as AP@.7, AP@.8, and CA@.5 respectively. Additionally, the inference speed of the models is evaluated through the FPS metric.

4.2 Ablation Experiments

Contour Transformer. The primary concern regarding the proposed C-AOI method is the accuracy of the proposed contour transformer. In this subsection, we replace the global deformation module proposed in E2EC with the contour transformer in the refined stage of C-AOI as a baseline. The results of the predicted contours at different stages are summarized in Table. 2a and illustrated in Fig. 8. In C-AOI, the initial contour quality is improved by incorporating local information. By utilizing the contour transformer instead of the global deformation module in the refined stage, the method shows significant improvements, including an increase of 0.9 in A-IOU, 2.1 in AP@.7, 2.4 in AP@.8, and 2.1 in CA@.5. These improvements demonstrate the ability of the contour transformer to effectively utilize global geometric information to refine the deformation path.

Positional Embedding. In our proposed contour transformer model, positional embeddings play a critical role in capturing the global geometric information. We implement two forms of positional embeddings, i.e., the spatial positional embedding in the sequence encoder and the cyclic positional embedding in the sequence decoder. To evaluate the influence of different positional embedding methods, we conduct experiments with various combinations and present the results in Table. 2b. Our findings show that removing both spatial and cyclic positional embeddings results in a 2.3 decrease in AP@.8 and a 1.6 decrease in CA@.5 compared to the full contour transformer model. Additionally, removing either of the two positional embeddings yields similar performance decreases. Furthermore, using the fixed sine positional embedding, that is, the default setting of transformer [28], causes a 0.8 decrease in AP@.8 and a 1.3 decrease in CA@.5 compared to our proposed positional embedding method.

Adaptive Matching Loss. In this study, we aim to improve the regression of contours. An effective regression loss is a crucial aspect of contour generation. Smooth L1 loss, Dynamic matching loss (DML) [32], and AML are tested to supervise the deformation stage in our architecture. Experiment results are shown in Table. 2c. We note that the regression loss of AML is significantly superior to the Smooth L1 and DML. In the process of all the predicted vertices with the most appropriate label vertices, AML not only improves the importance of inflection vertex matching but also strengthens the regression direction constraint of each vertex. Replacing AML with DML brings about an improvement of 1.0 in AP@.8 and 1.1 in CA@.5. It can be seen from Fig. 9 that the model supervised by AML predicts the best contour, especially at inflection vertices.

Multimodal Information. The results of the impact of using different modalities in C-AOI on the quality of the generated AOI can be found in Table. 2d. The results show that removing either the heatmap data or the convex hull data from the input has a relatively minor effect on the metrics of C-AOI, causing a decrease of 2.3 in AP@.8 and 0.9 in CA@.5 in the former case, and 2.2 in AP@.8 and 1.2 in CA@.5 in the latter case. In contrast, removing the satellite image or the road network data has a significant negative

Table 1: Description of the Meituan dataset.

City	Status	AOIs	Images
Beijing	Train and Valid	10536	10536 × 4
Xi'an, Chengdu, Nanjing, Wenzhou, Guilin, Baoding	Test	26208	26208 × 4

Table 2: Ablation experiments for C-AOI. All models are trained on Beijing data and tested on other six cities.

Stage	A-IOU	AP@.7	AP@.8	CA@.5
initial	56.2	40.2	23.3	35.7
coarse	59.7	43.2	26.5	39.6
baseline	61.3	45.1	29.1	42.8
refined	**62.2**	**47.2**	**31.5**	**44.9**

(a) Results of ablation experiments with contour transformer. We replace the contour transformer in the refined stage of C-AOI with the global deformation module proposed in E2EC as a baseline.

Encoder	Decoder	A-IOU	AP@.7	AP@.8	CA@.5
-	-	60.1	45.5	29.2	43.3
spatial	-	61.0	46.1	30.4	43.8
-	cyclic	61.2	46.2	30.5	44.2
sine	sine	61.4	46.5	30.7	43.6
spatial	cyclic	**62.2**	**47.2**	**31.5**	**44.9**

(b) Results with different positional embedding. We additionally contrast the sine positional encoding of the sequences, as the default setting of transformer [28].

Loss	A-IOU	AP@.7	AP@.8	CA@.5
Smooth-L1	61.2	46.4	30.1	42.3
DML	61.6	46.6	30.5	43.8
AML	**62.2**	**47.2**	**31.5**	**44.9**

(c) Results of the C-AOI being supervised by different loss functions.

Modality	A-IOU	AP@.7	AP@.8	CA@.5
without S	58.3	41.1	25.2	40.4
without R	59.0	42.1	25.1	39.2
without C	60.4	45.7	29.3	43.7
without H	60.5	45.8	29.2	44.0
all modalities	**62.2**	**47.2**	**31.5**	**44.9**

(d) Contribution of each modality data to C-AOI. Satellite Image (S), Road Networks (R), Heatmap (H), Convex Hull (C).



Figure 8: The prediction contours of C-AOI at different stages. The initial stage produces predictions that closely resemble an elliptical contour. The coarse stage only provides a rough outline. In the refined stage of the global deformation module, the predicted vertices of the global deformation module fluctuate significantly on the boundary, increasing the risk of missing buildings near the boundary of the AOI, while the contour transformer fits the boundary better.

impact on the metrics, with a decrease of 6.3 in AP@.8 and 4.5 in CA@.5 when the satellite image is removed, and a decrease of 6.4 in AP@.8 and 5.7 in CA@.5 when the road network data is removed. Our conclusion is that all the modalities contribute to the performance of the AOI contour, with the road network and satellite image being the most crucial, while the convex hull and heatmap play a supportive role.

4.3 Comparison Against State-of-the-art

For baselines, we select a clustering algorithm Convex Hull (Hull) [2], a semantic segmentation framework Connectivity U-Net (CNU) [5], and a contour-based model PolarMask [30]. For SOTA methods including both the post-processing framework and end-to-end model, we compare C-AOI not only with the Connectivity FCENet

Table 3: Comparison with the state-of-the-art methods on the Meituan real-world test dataset. All supervised models are trained with multimodal data. FPS is reported on a V100 GPU, except for the convex hull algorithm. CNU and CNF are based on semantic segmentation schemes, so their post-processing methods are kept in the experiments.

Method	Backbone	A-IOU	AP@.7	AP@.8	CA@.5	FPS
Hull	-	50.5	20.9	7.2	24.6	-
PolarMask	ResNeXt-101	51.2	36.9	22.5	34.1	**15.4**
CNU	UNet	52.4	38.9	23.4	36.8	2.36
E2EC	DLA-34	57.3	41.6	25.0	39.2	13.0
CNF	UNet-FCENet	59.9	44.2	26.6	38.4	0.95
C-AOI	DLA-34	**62.2**	**47.2**	**31.5**	**44.9**	12.7

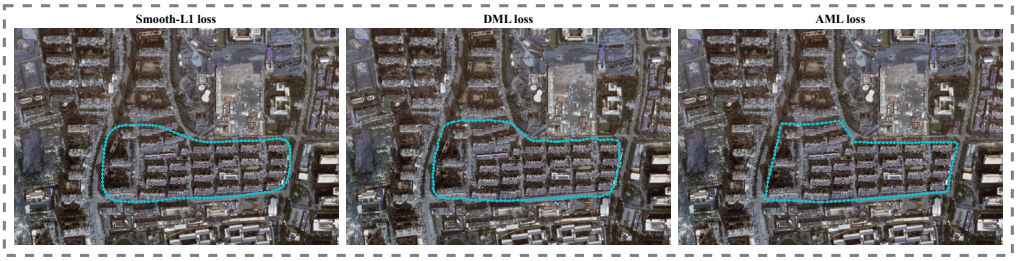


Figure 9: Different loss functions supervise the prediction results of C-AOI. The contour boundary of Smooth-L1 loss is too smooth, and DML is slightly better than Smooth-L1 loss. AML greatly improves the quality of boundary details, especially at inflection vertices.



(a) Hull (b) PolarMask (c) CNU (d) E2EC (e) CNF (f) C-AOI (g) Label

Figure 10: Visual cases obtained by different methods.

(CNF) [5] proposed by Meituan for AOI generation but also with E2EC [32], which has achieved state-of-the-art results in contour-based instance segmentation methods.

As demonstrated in Table. 3, C-AOI presents a highly competitive performance when compared to the state-of-the-art methods in terms of mask and boundary quality. Specifically, on the test dataset, C-AOI obtains a remarkable score of 62.2 in terms of A-IOU, and demonstrates an outstanding performance with 47.2 in AP@.7, 31.5 in AP@.7, and 44.9 in CA@.5. These results indicate a clear improvement over the CNF model proposed by Meituan, with a margin of 2.3 in A-IOU, 3.0 in AP@.7, 4.9 in AP@.8, and 6.5 in CA@.5. The difference in inference speed between C-AOI and CNF is attributed to the fact that C-AOI is an end-to-end model that regresses the AOI contour as a series of discrete vertices on the contour, resulting in a 13 times speedup. Furthermore, compared to the contour-based instance segmentation method E2EC, C-AOI still outperforms by 4.9 in A-IOU, 5.6 in AP@.7, 6.5 in AP@.8, and 5.7 in CA@.5, while being roughly equivalent in inference speed. As shown in Fig. 10, some of the results generated by different methods are visually compared, from which it is evident that C-AOI effectively captures the contour details of AOIs (such as inflection points and boundaries) that cannot be achieved by either E2EC or CNF.

5 CONCLUSION

C-AOI is a contour-based AOI generation approach that differs from previous work by Meituan. Rather than treating AOI contour prediction as semantic segmentation, C-AOI represents the contour through the regression of discrete vertices without any post-processing. This approach incorporates three unique components: a contour transformer architecture, a learnable cyclic positional encoding, and an adaptive matching loss function. The results of this study demonstrate that C-AOI significantly enhances both the mask and boundary quality of the generated AOIs on real-world datasets, while maintaining comparable inference speed to E2EC. Furthermore, C-AOI has been deployed in the Meituan platform for producing AOIs.

REFERENCES

[1] Albino Altomare, Eugenio Cesario, Carmela Comito, Fabrizio Marozzo, and Domenico Talia. 2016. Trajectory pattern mining for urban computing in the cloud. *IEEE Transactions on Parallel and Distributed Systems* 28, 2 (2016), 586–599.

[2] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)* 22, 4 (1996), 469–483.

[3] Loris Belcastro, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. 2018. G-RoI: automatic region-of-interest detection driven by geotagged social media data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 3 (2018), 1–22.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 213–229.

[5] Liying Chen, Daping Xiong, Shuiping Chen, Fangxiao Du, Renqing He, and Zhizhao Sun. 2022. Automatic generation of areas of interest using multimodal geospatial data from an on-demand food delivery platform (industrial paper). In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 1–10.

[6] 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).

[7] David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization* 10, 2 (1973), 112–122.

[8] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. 2021. Location-sensitive visual recognition with cross-iou loss. *arXiv preprint arXiv:2104.04899* (2021).

[9] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. 1983. On the shape of a set of points in the plane. *IEEE Transactions on information theory* 29, 4 (1983), 551–559.

[10] Hongchao Fan, Bisheng Yang, Alexander Zipf, and Adam Rousell. 2016. A polygon-based approach for matching OpenStreetMap road networks with regional transit authority data. *International Journal of Geographical Information Science* 30, 4 (2016), 748–764.

[11] Dsouza Prima Frederick and Ganesh Bhat. 2021. Review on Customer Perception Towards Online Food Delivery Services. (2021).

[12] Chiao-Ling Kuo, Ta-Chien Chan, I-Chun Fan, and Alexander Zipf. 2018. Efficient method for POI/ROI discovery using Flickr geotagged photos. *ISPRS International Journal of Geo-Information* 7, 3 (2018), 121.

[13] Bingnan Li, Liying Chen, Daping Xiong, Shuiping Chen, Renqing He, Zhizhao Sun, Samsung Lim, and Hai Jiang. 2022. Simultaneous detection of multiple areas-of-interest using geospatial data from an online food delivery platform (industrial paper). In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 1–10.

[14] Yinfeng Li, Chen Gao, Xiaoyi Du, Huazhou Wei, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Automatically Discovering User Consumption Intents in Meituan. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3259–3269.

[15] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. 2020. Polytransform: Deep polygon transformer for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9131–9140.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[17] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. 2019. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5257–5266.

[18] Jiajun Liu, Zi Huang, Lei Chen, Heng Tao Shen, and Zhixian Yan. 2012. Discovering areas of interest with geo-tagged images and check-ins. In *Proceedings of the 20th ACM international conference on Multimedia*. 589–598.

[19] Xinyi Liu, Qunying Huang, and Song Gao. 2019. Exploring the uncertainty of activity zone detection using digital footprints with multi-scaled DBSCAN. *International Journal of Geographical Information Science* 33, 6 (2019), 1196–1223.

[20] Yaxu Liu, Jui-Nan Yen, Bowen Yuan, Rundong Shi, Peng Yan, and Chih-Jen Lin. 2022. Practical Counterfactual Policy Learning for Top-K Recommendations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1141–1151.

[21] Zichen Liu, Jun Hao Liew, Xiangyu Chen, and Jiashi Feng. 2021. Dance: A deep attentive contour model for efficient instance segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 345–354.

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[23] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. 2020. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8533–8542.

[24] Grand View Research. 2022. *Online Food Delivery Market Size, Share & Trends Analysis Report By Type (Platform To Consumer Delivery, Restaurant To Consumer Delivery), By Region, And Segment Forecasts, 2022 - 2028*. Technical Report GVR-4-68039-942-2.

[25] Evaggelos Spyrou and Phivos Mylonas. 2016. Analyzing Flickr metadata to extract location-based information and semantically organize its photo content. *Neurocomputing* 172 (2016), 114–133.

[26] Vishal Srivastava, Priyam Tejaswin, Lucky Dhakad, Mohit Kumar, and Amar Dani. 2020. A geocoding framework powered by delivery data. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. 568–577.

[27] Ying Sun, Hengshu Zhu, Fuzhen Zhuang, Jingjing Gu, and Qing He. 2018. Exploring the urban region-of-interest through the analysis of online map search queries. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2269–2278.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[29] Zhuolin Wu, Li Wang, Fangsheng Huang, Linjun Zhou, Yu Song, Chengpeng Ye, Pengyu Nie, Hao Ren, Jinghua Hao, Renqing He, et al. 2022. A Framework for Multi-stage Bonus Allocation in meal delivery Platform. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4195–4203.

[30] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. 2020. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12193–12202.

[31] Wenqiang Xu, Haiyang Wang, Fubo Qi, and Cewu Lu. 2019. Explicit shape encoding for real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5168–5177.

[32] Tao Zhang, Shiqing Wei, and Shunping Ji. 2022. E2EC: An End-to-End Contour-based Method for High-Quality High-Speed Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4443–4452.

[33] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. 2020. Tracking objects as points. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*. Springer, 474–490.

[34] Chenming Zhu, Xuanye Zhang, Yanran Li, Liangdong Qiu, Kai Han, and Xiaoguang Han. 2022. SharpContour: a contour-based boundary refinement approach for efficient and accurate instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4392–4401.

NEON- Living Needs Prediction System in Meituan

Xiaochong Lan
Department of Electronic Engineering, BNRist, Tsinghua University
Beijing, China
lanxc22@mails.tsinghua.edu.cn

Chen Gao†
Department of Electronic Engineering, BNRist, Tsinghua University
Beijing, China
chgao96@gmail.com

Shiqi Wen
Meituan
Beijing, China
wenshiqi@meituan.com

Xiuqi Chen
Meituan
Beijing, China
chenxiuqi@meituan.com

Yingge Che
Meituan
Beijing, China
cheyingge@meituan.com

Han Zhang
Meituan
Beijing, China
zhanghan56@meituan.com

Huazhou Wei
Meituan
Beijing, China
weihuazhou@meituan.com

Hengliang Luo
Meituan
Beijing, China
luohengliang@meituan.com

Yong Li
Department of Electronic Engineering, BNRist, Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Living needs refer to the various needs in human's daily lives for survival and well-being, including food, housing, entertainment, etc. At life service platforms that connect users to service providers, such as Meituan, the problem of living needs prediction is fundamental as it helps understand users and boost various downstream applications such as personalized recommendation. However, the problem has not been well explored and is faced with two critical challenges. First, the needs are naturally connected to specific locations and times, suffering from complex impacts from the spatiotemporal context. Second, there is a significant gap between users' actual living needs and their historical records on the platform. To address these two challenges, we design a system of living **NE**eds predicti**ON** named NEON, consisting of three phases: feature mining, feature fusion and multi-task prediction. In the feature mining phase, we carefully extract individual-level user features for spatiotemporal modeling, and aggregated-level behavioral features for enriching data, which serve as the basis for addressing two challenges, respectively. Further, in the feature fusion phase, we propose a neural network that effectively fuses two parts of features into the user representation. Moreover, we design a multitask prediction phase, where the auxiliary task of needs-meeting way prediction can enhance the modeling of spatiotemporal context. Extensive offline evaluations verify that our NEON system can effectively predict users' living needs. Furthermore, we deploy NEON into Meituan's algorithm engine and evaluate how it enhances the three downstream prediction applications, via large-scale online A/B testing. As a representative result, deploying our system leads to a 1.886% increase *w.r.t.* CTCVR in Meituan homepage recommendation. The results demonstrate NEON's effectiveness in predicting fine-grained user needs, needs-meeting way, and potential needs, highlighting the immense application value of NEON.

CCS CONCEPTS

• **Information systems** → **Information systems applications**;

KEYWORDS

Living Needs Prediction; Deep Neural Networks; Multi-task Learning

ACM Reference Format:
Xiaochong Lan, Chen Gao†, Shiqi Wen, Xiuqi Chen, Yingge Che, Han Zhang, Huazhou Wei, Hengliang Luo, and Yong Li. 2023. NEON: Living Needs Prediction System in Meituan. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599874

1 INTRODUCTION

Living needs are the various needs generated by individuals in their daily routines for daily survival and well-being. Typical living needs include necessities such as food, housing, and personal care as well as leisure activities such as entertainment, for which there exists a diverse array of life service providers. Meituan is a large platform connecting customers to life service providers, in which users can meet almost all kinds of living needs. Unlike traditional information systems such as e-commerce websites, where users can only purchase products (*i.e.*, meeting one kind of living needs), Meituan allows access to various living services, such as booking

†Chen Gao is the corresponding author (chgao96@gmail.com).

This work is licensed under a Creative Commons Attribution International 4.0 License.

KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599874

https://www.metiuan.com

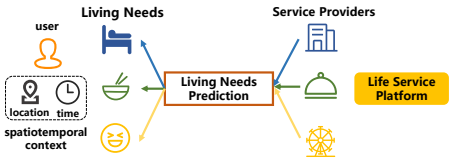


Figure 1: Living needs prediction aims to predict the specific living need of a user given the spatiotemporal context. By predicting users' living needs, life service platforms can recommend life services that can fulfill these needs.

a hotel, ordering takeout , etc. Moreover, an accurate understanding of users' needs can significantly improve user experience, for example, by enhancing the downstream recommendation tasks. Generally speaking, generating a specific kind of living need is naturally accompanied by a specific time or location, such as ordering food delivery at the office. Thus, the problem of *living needs prediction* can be defined as predicting the type of needs of the target user's spatiotemporal context, as illustrated in Figure 1.

For this new problem from the real-world scenario, there are two closely-related research topics, demand forecasting [10, 18, 29, 32, 34] and spatiotemporal activity prediction [3, 7, 15, 19, 41]. Demand forecasting is dedicated to predicting the quantity of a product or service that consumers will purchase. However, these works focus on the problem of aggregated demand prediction with times-series modeling, ignoring the needs of the specific user, time, and location. As for spatiotemporal activity prediction, the existing works focus mainly on either online activities such as App usage, or offline activities such as location visitation, while individuals' living needs can be fulfilled both in-store (offline) and via delivery (online), leading to a completely new problem.

There exist two critical challenges for living needs prediction as follows.

- **The impact of spatiotemporal context is complex.** As discussed above, for the same user, living needs at different locations or times are totally different. For example, the user in Figure 1 eats food delivery at noon and watches a movie at night. Additionally, the lifestyles, *i.e., the spatiotemporal pattern of living needs*, are extremely various for different users, further making it more difficult to model the spatiotemporal context.
- **There is a significant gap between users' actual living needs and their historical records on the platform.** Typically, users will face various kinds of situation in their life, and will generate multiple living needs. But they may only choose to satisfy one or a few of them on the platform, leading to a significant gap between their actual living needs and their historical records. We refer to the needs that can not be observed from historical records as *potential needs*. The case here is different from that in most recommender systems, where the actual interests and the historical record generally do not differ greatly for users, leading to another critical challenge.

To address these challenges, in this work, we described our deployed NEON system (short for living **NE**eds predicti**ON**) in Meituan, which includes three phases: **feature mining**, **feature fusion**, and **multitask prediction**. First, in the feature mining phase, to address the first challenge, we carefully design the spatial and temporal features for individual-level users, and to address the second challenge, we extract the behavioral-pattern features for group-level users. Second, in the feature fusion phase, we develop a feature-fusion neural network that combines internal preferences, impact from spatiotemporal context, and group behavior patterns to generate user representations, addressing both challenges. Last, as the complement to the main task of living needs prediction, we introduce the auxiliary task of needs-meeting way prediction to enhance the model's learning of spatiotemporal context, further addressing the first challenge.

The proposed NEON system plays a critical role in Meituan's recommendation engine with various downstream applications, including homepage recommendation, *Guess-you-like* page recommendation, and message pop-up recommendation, which requires the different-aspect ability of living needs prediction. After the deployment of the proposed system, we obtain stable and significant gains in three applications, providing strong real-world evidence for NEON's effectiveness from different perspectives.

The contribution of this work can be summarized as follows.

- To the best of our knowledge, we take the first step to study the problem of living needs prediction, which is a critical problem in real-world life service platforms but has not been well explored.
- We proposed the NEON system, which includes three phases of feature mining, feature fusion layer, and multitask prediction, which well addresses the two challenges, the complex impact of spatiotemporal context and missing behavioral data.
- We deploy NEON in Meituan's recommendation engine and conduct large-scale online A/B tests on three typical downstream applications, along with extensive offline evaluations. Offline experimental results verify that NEON can accurately predict users' living needs. The downstream evaluations strongly confirm NEON's high application value, with significant performance improvement in three downstream applications, among which a representative result is a 1.886% increase *w.r.t.* CTCVR for Meituan homepage recommendation.

2 PROBLEM STATEMENT

As we discussed above, in users' daily life, they generate various living needs, such as eating, accommodation, entertainment, beauty, etc. These needs can be fulfilled by life service providers in the city, which can be accessed through platforms connecting life service providers to customers. To enhance the user experience, it's crucial for these platforms to accurately predict users' needs and recommend appropriate services. This leads to the problem of living needs prediction.

As defined in the introduction, living needs prediction aims to predict the specific living needs of a user given the spatiotemporal context in which they are located (in the following we also refer to this as given the *user scene*). To clearly define the problem, with the help of experts, we divide all the living needs that users can satisfy on the platform into ten categories, shown in Table 1. We use $\mathcal{N}$ as the symbol for the set of all living needs. In response to these needs, all the life services on Meituan are also divided into 10 categories. The problem can be defined as follows.

Input: A dataset $\mathcal{O}^+$ of real-world life service consumption records that reflect users' living needs. Each instance in the dataset tells the

Table 1: 10 types of living needs that can be satisfied in Meituan

Living Needs	Ordering food delivery, Eating in a restaurant, Booking a hotel, Buying medicine, Specialty shopping online, Hair-dressing, Grocery shopping online, Beauty, Tourism and Entertainment

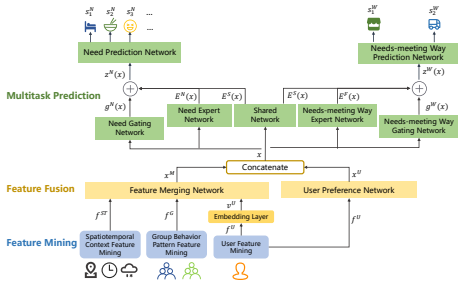


Figure 2: Illustration of living needs prediction system NEON.

kind of life service a specific user purchases, which indicates the specific living need n ($n \in \mathcal{N}$) of the user in a specific user scene i. **Output:** A model to estimate the probability that a user will generate the living need n in user scene i, formulated as $f(i, n|\mathcal{O}^+)$. Here $f(\cdot)$ denotes the function that the model aims to learn.

3 OUR NEON SYSTEM

To address the challenges mentioned in the introduction, we develop the NEON system made up of three phases: feature mining, feature fusion layer, and multitask prediction. First, in the feature mining phase, we address the first challenge by carefully designing spatiotemporal features for individual-level users and address the second challenge by extracting behavioral-pattern features for group-level users. The feature fusion phase then employs a feature-fusion neural network to seamlessly integrate internal preferences, spatiotemporal context impact, and group behavior patterns to generate complete user representations, overcoming both challenges. Last, in the multitask prediction stage, to enhance the model's understanding of spatiotemporal context, we introduce an auxiliary task of needs-meeting way prediction to the main goal of predicting living needs, providing additional support in addressing the first challenge. The deep feature fusion layer and the multi-task prediction parts of our system are illustrated in Figure 2.

3.1 Feature Mining

First of all, we use features that directly reflect user traits, such as users' profile, their recent behavior sequence, and their historical behaviors, as inputs for the model.

As mentioned above, for a specific user, his living needs are greatly affected by the spatial and temporal scenarios in which he is located. For example, *on a rainy midday, a person at work probably has the need to order food delivery; but on a sunny noon, he/she may have another need of going out to eat in the restaurant.* This complexity and variability of human living needs, driven by the flux of time and space, pose a considerable challenge in accurately modeling the impact of the spatiotemporal context. In order to tackle this challenge, we incorporate spatiotemporal context features as an integral part of our system's input.

What's more, on the platform, users may have potential needs with sparse or even non-existent history records. For example, *a person who never buys medicine on the platform may have a cold and need to buy medicine online one day.* Such potential needs are difficult for the model to grasp. To address the challenge of modeling potential needs, we introduce group behavior pattern features to help the model learn the potential living needs of users.

Below we give a detailed description of the three categories of features.

3.1.1 User Features. This group of features includes user profiles and user history behavior sequences.

- **User profiles f_p^U.** The user's profile, including their age, gender, etc.
- **User recent online behavior sequence f_{rb}^U.** The sequence of items recently clicked by the user in the platform; the sequence of items recently ordered by the user in the platform.
- **User aggregated historical online behavior f_{hb}^U.** The percentage of times users buy each type of life service.
- **User offline visitation record f_{ov}^U.** The 50 most visited POIs (point of interest) by the user in the last six months; the 50 most visited AOIs (area of interest) by the user in the last six months.

We concatenate all the mentioned features above to get a sparse user feature vector f^U, formulated as follows:

$$f^U = \left[f_p^U, f_{rb}^U, f_{hb}^U, f_{ov}^U\right]. \tag{1}$$

3.1.2 Spatiotemporal Context Features. Users' living needs are greatly affected by time, location, and other environmental factors. Thus, we introduce spatiotemporal context features as part of the input of our system to help our system model the complex impact of spatiotemporal context, which can be listed as follows.

- **Time f_t^{ST}.** Current time period. More than one time period feature of different granularity is applied, including hour, day, whether it is a holiday, etc.
- **Location f_l^{ST}.** The POI (point of interest) embedding of the user's real-time location; the AOI (area of interest) embedding of the user's real-time location; the city embedding of the user's real-time location. The location features are hourly real-time features.
- **Weather f_w^{ST}.** Weather information for the user's city or region, including wind, humidity, temperature, and weather type (sunny, rainy, snowy, etc.). Weather features are refined to hourly granularity.
- **Travel state f_{ts}^{ST}.** Information about whether the user is located in his/her resident city. Possible states include *based in resident city*, *about to travel*, and *on travel*.

The dense spatiotemporal context feature vector f^{ST} is created by concatenating all previously mentioned context features, formulated as follows:

$$f^{ST} = \left[f_t^{ST}, f_l^{ST}, f_w^{ST}, f_{ts}^{ST}\right]. \tag{2}$$

3.1.3 Group Behavior Pattern Features. We introduce group behavior pattern features to supplement the sparse individual behavior of users, in order to assist in identifying the potential living needs of individual users.

- **Group aggregated behavior f_a^G.** We first segment users into groups based on their profiles. In each group, we get the group aggregated behavior by calculating the percentage of views, clicks, and purchases of each type of life service among all views, clicks, and purchases initiated by the group. For each user, the group aggregated behaviors of the groups the user belongs to are used as features. For example, a middle-aged person has group aggregated behaviors feature of middle-aged users and other groups he/she is in.
- **Popularity in the current time period f_{ct}^G.** We cut all time into time periods according to different criteria, such as whether it is a holiday, if it is morning, noon, or night, etc. Then in each time period, we calculate the popularity of each type of life service by calculating the percentage of times the life service is viewed/clicked/purchased among all views/clicks/purchases happening in this time period. We determine the time periods in which the current time is located, and use the popularity of each type of life service in these time periods as a feature. For example, if the user opens the app on Christmas night, popularity on holiday and popularity at night of each kind of life service are set as features.
- **Group behavior pattern in spatiotemporal context f_{st}^G.** By discovering group preferences in different spatiotemporal contexts, we further capture more fine-grained group behavior patterns. We calculate the percentage of views/clicks/purchases of each type of life service initiated by each group in each kind of spatiotemporal context. These fine-grained patterns are used as features of the model. For example, the group preference of middle-aged people at work at noon on working days are used to enrich the representations of every individual within this demographic in such spatiotemporal scenario.
- **User behaviors augmented by inter-need correlation f_{ic}^G.** There is an inherent association across different types of users' living needs. This association can be leveraged to improve prediction performance. For example, *a user who frequently purchases hairdressing services may also be inclined to purchase beauty services.* We use the association rule mining algorithm to analyze the co-occurrence of different life service categories, filter out high-correlation relationships, and employ them to augment user behavior as input features.

We combine all previously mentioned group behavior pattern features to generate the dense group behavior pattern feature vector f^G, which is formulated as follows:

$$f^G = \left[f_a^G, f_{ct}^G, f_{st}^G, f_{ic}^G \right]. \tag{3}$$

3.2 Feature Fusion Layer

As mentioned in Section 2, we refer to a user in a specific spatiotemporal context as a user scene. For a user scene i, after feature mining, we have dense spatiotemporal features f_i^{ST}, dense group pattern features f_i^G, and sparse user features f_i^U. For brevity of presentation, we omit the subscript i in some of the expressions below. We designed a feature fusion layer to integrate these features into the input of the subsequent prediction module.

We first set up an embedding layer, which processes the high-dimensional sparse user feature vector f^U into a low-dimensional dense vector v^U. To address the challenges of complex impact of spatiotemporal context and users' potential needs, we mine spatiotemporal features and group behavior pattern features in the feature mining phase, respectively. With these features as input, we use a feature merging network to model the interaction between spatiotemporal contexts, group behavior patterns, and users as follows,

$$x^M = h^M \left(\left[f^{ST}, f^G, v^U \right] \right), \tag{4}$$

where $[\cdot]$ denotes concatenation operation. Here h^M is the feature merging network, which merges three information sources of spatiotemporal contexts, group behavior patterns, and user preference into a fusion representation x^M.

Moreover, users have their own internal characteristics that are independent of the spatiotemporal scene they are in and the group they belong to. To model the internal characteristics of users, we generate a representation as follows,

$$x^U = h^U \left(f^U \right), \tag{5}$$

where h^U denotes the user preference network that turns raw user features into dense user preference representation. We then concatenate the two parts of representations into the full representation of the user scene:

$$x = \left[x^M, x^U \right]. \tag{6}$$

In brief, we design a feature fusion layer to tackle both challenges by considering the influence of spatiotemporal context, incorporating group behavior patterns, as well as extracting individual preferences.

3.3 Multitask Prediction

We further design a prediction module which takes user scene representation as input to predict users' living needs. The module is tasked with two objectives: *fine-grained need prediction* and *needs-meeting way prediction*. Fine-grained need prediction is to predict the specific living need of the user. Neets-meeting way prediction is to predict the preferred way of the user to meet their needs.

Specifically, among the ten kinds of needs which we mentioned in the problem formulation, there are two ways to satisfy the needs: in-store and via-delivery. In other words, consumers can choose to satisfy their needs by visiting a physical store or by ordering online and then receiving goods via delivery. Each type of the 10 needs can be classified into one of two categories, in-store needs or via-delivery needs. We show the classification in Table 2. Actually, needs-meeting way prediction is to predict whether the preferred way of the user to meet their needs is in-store or via-delivery.

Users' preferences for needs-meeting ways are strongly affected by the spatiotemporal context. For example, *a person at work during lunchtime on a weekday is more likely to have the need to order food delivery (via delivery), while the same person in a shopping district on a weekend evening is more likely to visit a store for a meal (in-store).* With this in mind, we include the needs-meeting way prediction task which is jointed trained with the main task of need

Table 2: The classification of the 10 living needs that can be satisfied on Meituan

Living needs that can be satisfied via delivery	Specialty shopping online, Grocecy shopping online, Ordering food delivery, Buying medicine
Living needs that can be satisfied in store	Eating in a restaurant, Hotel, Hair-dressing, Beauty, Tourism, Entertainment

prediction to enhance the model's ability to learn spatiotemporal context information. Next, we describe how we get the prediction results of the two tasks. We use y^W and y^N to denote the prediction result of needs-meeting way and specific need. $y^k, k \in \{W, N\}$ can be generated as follows,

$$\begin{aligned} y^k &= t^k(z^k), \\ \text{where } z^k &= g^k(x)_0 E^k(x) + g^k(x)_1 E^S(x). \end{aligned} \tag{7}$$

Here t^k is the prediction neural network for task k. There are a variety of choices in the specific structure of the neural network. In Section A.1, we will state our specific choice. To avoid verbosity, we will use *network* to replace *neural network* in the following text. The output of t^N, y^N, is the scores of ten types of living needs, and the output of t^W, y^W, is the scores of in-store and via-delivery needs-meeting ways. We use s^N_{im} to denote the score of need m for user scene i, and use s^W_{in} to denote the score of needs-meeting way n for user scene i. E^k is the expert network [22, 35] for task k. E^S is the shared network between the two tasks. E^S is responsible for generating general representations that are common to both tasks, while E^k is responsible for learning task-specific representations that are more fine-tuned to the specific task k. The gating network g_k determines what proportion of information input each task's prediction network receives from the shared network and the expert network. We formulate the gating network as follows,

$$g^k(x) = \text{Softmax}(W_k x), \tag{8}$$

where $W_k \in \mathbb{R}^{2\times d}$ is trainable weights for task k. The gating network takes x as input, and outputs the relative importance of the shared and task-specific representations for a given tasking, allowing the model to selectively attend to the most relevant information and improve its performance. In summary, to address the complexity of spatiotemporal context impact, we introduce an auxiliary task of needs-meeting way prediction which is jointly trained with the main task of fine-grained living needs prediction to enhance our system's learning of spatiotemporal context. The multitask prediction module in our system produces a score for each living need and needs-meeting way.

3.4 Model Training

In this section we describe how our system is trained. Corresponding to the two tasks, we design two parts of loss. We design need prediction loss taking into account the fact that the frequency of different needs arising in users' lives is different. For example, *a user may need to order food delivery for lunch every workday, but rarely need to buy medicine.* In order to address the class imbalance issue for different living needs, we propose using a multi-class focal loss which can decrease the effect of needs with a high volume of training data on the final prediction loss. The need prediction loss can be formulated as follows,

$$\text{Loss}_{\text{need}} = -\sum_{i\in O}\left(\sum_{n=1}^{10}\left(1-q^N_{in}\right)^{\gamma}\chi^N_{in}\log\left(q^N_{in}\right)\right), \tag{9}$$

$$\text{where } q^N_{in} = \text{Softmax}\left(s^N_{in}\right) = \frac{e^{s^N_{in}}}{\sum_n e^{s^N_{in}}}. \tag{10}$$

Here O is the training set, s^N_{in} is the score of living need n for user scene i, γ is the hyperparameter which decides the importance of difficult samples, χ^N_{in} is 1 if n is the ground truth need for user scene i, else it is 0. For the needs-meeting way prediction task, we use BCE loss as prediction loss. We formulate it as follows,

$$\text{Loss}_{\text{way}} = -\sum_{i\in O}\left(\sum_{m=1}^{2}\chi^W_{im}\log\left(q^W_{im}\right)\right), \tag{11}$$

$$\text{where } q^W_{im} = \text{Softmax}\left(s^W_{im}\right) = \frac{e^{s^W_{im}}}{\sum_m e^{s^W_{im}}}. \tag{12}$$

Here O is the training set, s^W_{im} is the score of needs-meeting way m (in store or via delivery)for user scene i. χ^W_{im} is 1 if m is the ground truth needs-meeting way for user scene i, else it is 0. In our system, the feature integration module and multitask prediction module are trained end to end. The entire loss function is:

$$\text{Loss} = \lambda_1 \text{Loss}_{\text{need}} + \lambda_2 \text{Loss}_{\text{way}}. \tag{13}$$

λ_1 and λ_2 are hyperparameters that control the importance of the two parts of loss.

4 OFFLINE EVALUATION

4.1 Experimental Settings

4.1.1 Dataset. We conduct an offline experiment on a real-world dataset at a billion-scale. The dataset comprises a sampling of all 2022 purchase records on the platform, based on the percentage of purchases for each type of life service. It consists of over 7 billion actual purchase records from 65 million users. The details of datasets are provided in Appendix A.2.

4.1.2 Metrics. We design three metrics, namely Sort Accuracy (SA), Via-delivery Sort Accuracy (VDSA), and In-store Sort Accuracy (ISSA), to measure the performance of our system and baseline systems in predicting living needs. SA measures the overall accuracy of sorting living needs based on their scores. VDSA focuses on the accuracy of predicting needs that are satisfied via delivery, while ISSA focuses on the accuracy of predicting needs for in-store scenarios. Detailed definitions of these metrics are provided in Appendix A.3.

4.1.3 Baselines. To illustrate the effectiveness of our system, we compare it with two baselines widely in actual production environments, including **DIN [42]**, **DNN [5]**, **DCN [38]**, **ESMM [24]**, and **MMOE [22]**. We will provide a detailed description of these baselines in Appendix A.4.

Table 3: Offline experimental performance of NEON and baselines.

Method	VDSA	ISSA	SA
DIN	0.9044	0.7467	0.8700
DNN	0.9060	0.7500	0.8718
DCN	0.9051	0.7466	0.8701
ESMM	0.9080	0.7476	0.8708
MMOE	0.9097	0.7573	0.8757
NEON	**0.9175**	**0.8277**	**0.9070**
Improvement	0.86%	9.30%	3.57%

Table 4: Performance of NEON with/without multitask prediction.

	VDSA	ISSA	SA
w.o.	0.9089	0.8084	0.8968
with	0.9175	0.8277	0.9070
Improvement	0.95%	2.39%	1.14%

4.2 Overall Performance

We test the performance of our proposed system and baselines on the living needs prediction task, and show the results in Table 3. We can have the following observations.

- **Our system steadily outperforms all baselines on all metrics.** The improvement of our system compared to the best baseline is 0.86%, 9.30%, and 3.57% *w.r.t.* VDSA, ISSA, and SA, respectively. The significant performance gain confirms the effectiveness of our system on the living needs prediction task. Furthermore, such a significant improvement in the ability to predict living needs will result in a huge benefit in real-world production scenarios, which will be further confirmed through online evaluation.
- **Our system achieves greater improvement on ISSA.** The task of predicting users' in-store living needs is relatively difficult, since the in-store consumption data is sparser, and the relationships between spatiotemporal context and in-store needs are more complex. On this task, all methods perform the worst, and our system outperforms baselines on a large margin, with an improvement of 9.30% *w.r.t.* ISSA. This further confirms our model's ability to tackle the complex impact of spatiotemporal context and discovering potential needs.

4.3 Ablation Study

As mentioned in Section 3.3, we introduce the task of needs-meeting way prediction to enhance the system's learning of spatiotemporal context. To study the effectiveness of the multitask prediction design, we remove it from our system to observe the impact of the design on the system performance. Specifically, we change the system structure by removing the needs-meeting way prediction network t^W and taking the sum of z^N and z^W as the input of the need prediction network t^N, and test the performance of the changed system. The results are shown in Table 4. The results show that our proposed system outperforms the system without multitask prediction design. Our system performs better *w.r.t.* VDSA by 1.16%, *w.r.t.* ISSA by 2.39%, and *w.r.t.* SA by 1.14%. The significant performance improvement confirms the validity of the multitask prediction design.

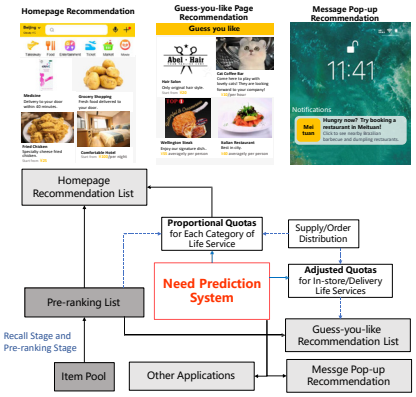


Figure 3: Online deployment of NEON in Meituan Recommendation Engine. In summary, NEON helps to generate the quotas of categories of life services in the recommendation list (homepage recommendation and *guess-you-like* recommendation), or decide the one category to be recommended to the user (message pop-up recommendation).

5 ONLINE EVALUATION

For life service platforms such as Meituan, understanding and predicting users' living needs are important in all business scenarios. In this section, we evaluate the performance of three downstream recommendation tasks when deploying NEON into Meituan's recommender engine, and the illustration of deployment is shown in Figure 3. Specifically, the three typical applications are homepage recommendation, *Guess-you-like* recommendation, and message pop-up recommendation. The performance of NEON on these applications reflects its effectiveness on fine-grained need prediction, needs-meeting way prediction, and potential need prediction, respectively. We elaborate on the online testing of three tasks one by one as follows.

5.1 Homepage Recommendation

In this section, we evaluate the performance of NEON when deployed to homepage recommendation, which reflects its ability on fine-grained living need prediction.

5.1.1 Deployment Scenario. Typically, users open Meituan mobile APP to meet their certain living need. In the overall recommendation list on homepage offered by Meituan, users probably only focus on the items which belong to the type of life service that can fulfill their living needs, and choose one item from the items. Whether an item is in the category of life service that the user *need* is at least as important as whether the user *likes* the item. To emphasize the importance of recommending *needed* items, the engine follows a two-step approach to generate the final recommendation

Table 5: Results of A/B tests on homepage recommendation

Metric	w/o NEON	with NEON	Improvement
CTR	3.0601×10^{-2}	3.0672×10^{-2}	+0.230%
CVR	1.1497×10^{-1}	1.1687×10^{-1}	+1.652%
CTCVR	3.5183×10^{-3}	3.5846×10^{-3}	+1.886%

list from the pre-ranking result. For a user scene, it first decides the quotas for all the ten categories of life services and then generates the final list according to the quotas and the prediction scores of the items. The supply quotas are generated based on the scores our system outputs along with other criteria.

In short, our NEON system is used to generate the *quotas* of different kinds of life services in the homepage recommendation list.

5.1.2 Experiment Setting. We will first give a detailed description of how our system is used in generating the quotas for each category of life services. At first, we recall local life service items into the recall pool using various strategies, such as popularity and collaborative filtering. Then the engine outputs a preliminary recommendation list based on the recall pool, called pre-ranking list. After that, we take the Softmax normalized scores output by our system as the proportional quotas for each category of life service. We further adjust the quotas taking the proportion of each category in the pre-ranking list, supply distribution by category, and order distribution by category into account. The generated quotas are the proportion of each category in the lists received by users. The lists are generated considering both quotas of categories and prediction scores of items.

We compare the homepage recommendation performance of our whole recommendation engine with and without our system through online A/B tests. The tests last for one week and involved around 4.5 million users.

In our tests, the users are randomly divided into two buckets of similar size and assigned different methods for calculating quotas for each category. Specifically, for the first group, we use the method described previously, while for the second group, we generate quotas based only on the proportion of each category in the pre-ranking list, supply distribution by category, and order distribution by category. We maintain consistency across all other modules to ensure a fair comparison.

For metrics, we use *Click Through Rate* (CTR), *Conversion Rate* (CVR), *Click to Conversion Rate* (CTCVR) to measure the quality of the final recommendation list, which are widely-used measurements [21, 24, 39].

5.1.3 Performance. The results of our A/B tests are shown in Table 5. From the results, we can have the following observations:

- There is a significant improvement with respect to all metrics. The increase *w.r.t.* CTR and CVR are 0.230% and 1.653%, respectively, which is a notable improvement.
- The increase *w.r.t.* CTCVR is 1.886%. CTCVR indicates how likely users are to purchase the recommended items. Such improvement can result in a substantial rise in total consumption on the platform.
- The rise *w.r.t.* GTV-CC is 3.627%. The remarkable uplift demonstrates the outstanding capability of our system in predicting users' living needs that has no historical record.

Table 6: KLD of different time periods between real user order distribution and quotas given with/without NEON.

Time Period	Kullback-Leibler Divergence		Improvement
	w/o NEON	with NEON	
0-4	0.2578	0.2201	-14.62%
5-8	0.2237	0.1907	-14.75%
9-10	0.2179	0.1902	-12.71%
11-12	0.2408	0.2138	-11.21%
13-16	0.2323	0.2053	-11.62%
17-19	0.2353	0.2096	-10.92%
20-24	0.2333	0.1995	-14.49%

We further calculate the *Kullback-Leibler divergence* [16] (KLD) between real user order distribution by category and the average proportional allocation given by the online engine with/without our system. Lower KLD indicates the quotas match better with real user order distribution, which can be regarded as real user needs distribution. The results of different time periods throughout the day are shown in Table 6. The time periods are separated based on the business characteristics of the platform during each hour. Similar hours are grouped within a single period.

From the results, it can be seen that in all time periods, the Kullback-Leibler Divergence between the actual user order distribution by category and the average proportional allocation generated by the online engine with NEON is significantly less than that without our NEON. The percentage of decrease (improvement) is 12.90% on average. This indicates that the quotas produced by the online engine with our system are more aligned with the actual user consumption distribution, or the real-life user living needs, compared to the ones generated without our system. This can be regarded as evidence of our model's effectiveness in addressing the intricate impact of spatiotemporal context, and further confirms our system's strong ability for predicting fine-grained user needs.

5.2 Guess-you-like Recommendation

This section assesses NEON's performance in guess-you-like recommendation, highlighting its ability on needs-meeting way prediction.

5.2.1 Deployment Scenario. Meituan designs a *Guess-you-like* page, which user may be guided to during their leisure time. Typically, users browse this page to satisfy their *unnecessary* living needs, such as entertainment or beauty, etc. In this page, they purchase items they need and also *like*. To provide more choices for users in the categories of life services they need, the recommendation list should have more items in those categories. In this page, we assume that users are more concerned with being recommended items they like. Therefore, we do not maintain the proportion of a category in the "Guess-you-like" recommendation list just because there is a slight possibility that it is essential, as users can use modules other than Guess-you-like such as the search button to find necessary life services. To ensure enough items are in the needed category, the online engine generates new quotas for Guess-you-like page based on the quotas for homepage recommendation list and the in store/via delivery score output by our system.

Table 7: Overall results of A/B tests on *Guess-you-like* page recommendation.

Metric	w/o NEON	with NEON	Improvement
CTR	9.5707×10^{-2}	9.5647×10^{-2}	-0.063%
CVR	1.3568×10^{-1}	1.3606×10^{-1}	+0.280%
CTCVR	1.3014×10^{-2}	1.2985×10^{-2}	+0.218%
NFR-UV	1.0128×10^{-4}	1.0294×10^{-4}	+1.605%
NFR-PV	8.3283×10^{-5}	8.6726×10^{-5}	+3.342%

In summary, the needs-meeting way prediction results of NEON are used to further adjust the *quotas* of different life services in *Guess-you-like* recommendation list.

5.2.2 Experiment Setting. We conduct online A/B tests involving about 7 million users over a period of two weeks. We randomly divide users into two buckets, each of which has a similar amount of users, and assign them different methods of generating the quotas of categories in the Guess-you-like page. Specifically, for the first bucket we adopt the same strategy as in the homepage recommendation. For the second bucket, we calculated the score of both needs-meeting ways, and turn higher the quotas of categories whose needs-meeting way gets a higher score. As for metrics, we use CTR, CVR, CTCVR, *Negative Feedback Rate for Unique Visitor* (NFR-UV), *Negative Feedback Rate for Page View* (NFR-PV) to measure the quality of the recommendation list in the Guess-you-like page. NFR-UV and NFR-PV emphasize that the recommendation list should not contain items users dislike.

5.2.3 Performance. Under the aforementioned settings, we conduct extensive online A/B experiments. The results are shown in Table 7. We list our observations as follows:

- The performance gain *w.r.t.* CVR, CTCVR, and OV are 0.280%, 0.218%, and 0.310%, respectively. Such improvement resulting from adjusting the quotas with the assistance of the needs-meeting way prediction module can lead to a significant increase in the consumption amount.
- The NFR-UV and NFR-PV decrease by 1.57% and 3.31% respectively, indicating that the recommendation engine is able to recommend fewer items that users dislike in the Guess-you-like page by adjusting the quotas with the aid of the needs-meeting way prediction module in our system.

We further calculate the performance increase on some popular categories of life services in Guess-you-like page. The results are shown in Table 8. From the result, we can observe that:

- For all these categories there is average relative improvement of 3.801% *w.r.t.* CVR and 3.765% *w.r.t* CTCVR. With such improvements, all these categories can enjoy a notable increase in order volume on Meituan platform.
- Among all these categories, the rise on Kids category is the most significant, which is 8.723% *w.r.t.* CVR and 9.691% *w.r.t.* CTCVR.
- There are also slight decreases *w.r.t.* CTR of 0.034% for Food Delivery, -0.449% for Beauty, and -0.595% for Hotel. These decreases are within the typical fluctuations observed in the market.

The above results, which demonstrate the successful implementation of our system in guess-you-like recommendation, further illustrate the efficacy of our system, particularly in needs-meeting way prediction.

Table 8: The A/B tests results on several popular categories of life services in *Guess-you-like* page.

Category	Metric	w/o NEON	with NEON	Improvement
Food	CTR	1.8298×10^{-2}	1.8292×10^{-2}	-0.034%
	CVR	2.8841×10^{-1}	2.8987×10^{-1}	+0.506%
	CTCVR	5.2774×10^{-3}	5.3023×10^{-3}	+0.472%
Beauty	CTR	1.8769×10^{-2}	1.8684×10^{-1}	-0.449%
	CVR	1.2583×10^{-3}	1.2889×10^{-3}	+2.427%
	CTCVR	2.3616×10^{-4}	2.4081×10^{-4}	+1.967%
Kids	CTR	1.7717×10^{-2}	1.7874×10^{-2}	+0.890%
	CVR	7.8825×10^{-3}	8.5701×10^{-3}	+8.723%
	CTCVR	1.3965×10^{-1}	1.5319×10^{-4}	+9.691%
Hotel	CTR	1.6567×10^{-2}	1.6469×10^{-2}	-0.595%
	CVR	4.1454×10^{-2}	4.2923×10^{-2}	+3.546%
	CTCVR	6.8678×10^{-1}	7.0690×10^{-4}	+2.930%

5.3 Message Pop-up Recommendation

In this section, we test our model's performance in message pop-up recommendation to observe its potential need prediction ability.

5.3.1 Deployment Scenario. In the case where a user has a living need that has never or rarely been fulfilled on Meituan platform, he/she probably won't launch Meituan mobile app, as they may not be aware that this need can be satisfied on Meituan or may not be accustomed to fulfilling this need on Meituan. So, if our engine runs in the background of the phone and detects the user's potential living need, it will send a message pop-up to the user with a recommendation for a life service solution, if the user allows it. The message pop-up recommends the user a life service that is in the category of the highest score output by our system.

In brief, with the ability of potential needs prediction, NEON is deployed for message pop-up recommendation.

5.3.2 Experiment Setting. We conduct online A/B tests involving one million users over a period of two weeks. The users are randomly divided into two buckets of similar volume and are assigned different strategies for selecting items to be sent in message pop-ups. For the first bucket, the message pop-up recommends items that belong to the category with the highest score output by our system. For the second bucket, in each hour of the day, we calculate the popularity and average CTR of each category. We distribute the traffic for message pop-ups to the categories with the most popularity and highest average CTR.

We use CTR, CVR, CTCVR, OV, *Number of Cold-start Customers* (NCC) to measure the performance of our system in potential need prediction. NCC represents the number of customers who purchase a lifestyle service that they have never acquired on the platform previously.

5.3.3 Performance. The results of the online A/B tests are shown in Table 9. We can observe that:

- By replacing the algorithm based on popularity and average CTR with our living need prediction system NEON to determine the category of recommendation in message pop-up, all metrics show

Table 9: Results of A/B tests on message pop-up recommendation.

Metric	CTR	CVR	CTCVR	OV	NCC
Improv.	+8.21%	+78.64%	+85.71%	+95.92%	+74.26%

significant improvement. CTR, CVR, CTCVR, and OV increase by 8.21%, 78.64%, 85.71%, and 95.92%, respectively.

- NCC increases by 74.26%, indicating that the online deployment of our system in message pop-up recommendations results in a 74.26% increase in the number of customers purchasing a life service that they have not previously acquired on the Meituan platform, via the message pop-up feature.

Our system is able to accurately detect the potential needs of users for a specific lifestyle service, even in instances where they have never previously purchased that service on the Meituan platform, and accordingly deliver targeted message pop-ups to them. This result strongly demonstrates the exceptional capability of our system in predicting users' potential needs.

In summary, the success achieved by our NEON system on three downstream recommendation tasks proves its effectiveness on fine-grained need prediction, needs-meeting way prediction, and potential need prediction, respectively. The significant performance gain in real applications can lead to huge benefits.

6 RELATED WORK

As discussed above, in this work we explore the problem of living needs prediction, defined as predicting the specific living **needs** of a user given the **spatiotemporal** context. Thus, there are two closely related research topics: demand forecasting and spatiotemporal activity prediction.

6.1 Demand Forecasting

Demand forecasting aims at predicting the quantity of a product or service that consumers will purchase. It helps in making informed decisions on inventory management, production scheduling, pricing strategy, etc. The problem of demand forecasting is broad and multifaceted, affecting many different industries, including restaurant [18], manufacturing [32], retail [29], tourism [34], energy [10], transportation [36], etc.

To address the problem of demand forecasting, researchers have proposed various methods which can be broadly classified into three categories: statistical models [8, 11, 18, 26, 37], machine learning models [2, 23, 30, 31], and deep learning models [6, 14, 17, 28]. Statistical models, such as exponential smoothing [33], are well-suited for long-term demand forecasting as they are based on historical trends and patterns. However, they are not adept at handling variations or outliers in the data, making them unsuitable for volatile or short-term demand forecasting. On the other hand, machine learning models for demand forecasting, such as Random Forest based models [1, 27], are efficient at short-term demand forecasting, but their performance drops when it comes to long-term forecasting. Deep learning models such as LSTM based models [17] and GAN based models [12] have the ability to capture complex patterns and dependencies in the data, making them suitable for both short-term and long-term demand forecasting. However, they require a large amount of data to work well.

Existing demand forecasting methods can not handle the problem of living needs prediction. These methods focus on the overall demand for a particular product or service in a market, but in this work, we aim at predicting the need of a specific consumer. What's more, demand forecasting methods predict demands in several months or years, while in this work we predict a user's need at a specific time and location.

6.2 Spatiotemporal Activity Prediction

Spatiotemporal activity prediction aims to predict the activity of a user at a given time and location. Previous works have employed various methods to perform spatiotemporal activity prediction. One popular approach is to build a tensor using historical data and then conduct tensor factorization to learn intrinsic association [3, 7, 41]. For example, Fan *et al.* [7] propose to integrate tensor factorization with transfer learning for online activity prediction. Additionally, WDGTC [20] proposes a low-rank tensor decomposition and completion framework for passenger flow prediction by introducing L1-norm and Graph Laplacian penalties. Recently, researchers have introduced Graph Convolutional Networks [15] (GCNs) to achieve high performance in spatiotemporal activity prediction. For example, SA-GCN [40] develops a Graph Convolutional Network with meta path-based objective function for App-usage prediction task. Furthermore, DisenHCN [19] utilizes a heterogeneous hypergraph to model fine-grained user similarities, resulting in significant performance gains.

However, existing works on spatiotemporal activity prediction focus on predicting the specific activities of people, while our work focuses on the general living needs which are the driving force behind specific consumption behaviors. What's more, these works mainly focus on either online or offline activities, but in our work, we predict living needs can be satisfied both in store (offline) and via delivery (online) by different kinds of life services, which is beyond the capabilities of existing methods.

7 CONCLUSION AND FUTURE WORK

In this work, we approach the new problem of living needs prediction, which is critical in life services platforms. We present the NEON system in Meituan, consisting of three phases, feature mining, feature fusion, and multitask prediction. Large-scale online A/B testing in three downstream applications, along with extensive offline evaluation, strongly confirm the effectiveness of our system. As for future work, we plan to test NEON's performance in more downstream applications.

ACKNOWLEDGEMENT

This work is supported in part by National Key Research and Development Program of China under 2022YFB3104702. This work is supported in part by National Natural Science Foundation of China under 62272262, 61971267, and 61972223. This work is supported in part by a grant from the Guoqiang Institute, Tsinghua University under 2021GQG1005. This work is supported in part by Beijing National Research Center for Information Science and Technology. This work is also supported by Meituan.

REFERENCES

[1] Raza Abid Abbasi, Nadeem Javaid, Muhammad Nauman Javid Ghuman, Zahoor Ali Khan, and Shujat Ur Rehman. 2019. Short term load forecasting using XGBoost. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*. Springer, 1120–1131.

[2] K Aishwarya, N Rao, Nikita Kumari, Akshit Mishra, and MR Rashmi. 2020. Food Demand Prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)* 7 (2020).

[3] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. 2015. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *WWW*. 130–140.

[4] Chen Cheng, Haiqin Yang, Irwin King, and Michael Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 26. 17–23.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[6] Brendan Andrew Duncan and Charles Peter Elkan. 2015. Probabilistic modeling of a sales funnel to prioritize leads. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1751–1758.

[7] Yali Fan, Zhen Tu, Yong Li, Xiang Chen, Hui Gao, Lin Zhang, Li Su, and Depeng Jin. 2019. Personalized context-aware collaborative online activity prediction. *UbiComp* 3, 4 (2019), 1–28.

[8] Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj El Moussami, and Abdeslam Lachhab. 2018. Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management* 10 (2018), 1847979018808673.

[9] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 315–323.

[10] Luis Hernandez, Carlos Baladron, Javier M Aguiar, Belén Carro, Antonio J Sanchez-Esguevillas, Jaime Lloret, and Joaquim Massana. 2014. A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. *IEEE Communications Surveys & Tutorials* 16, 3 (2014), 1460–1495.

[11] Jakob Huber, Alexander Gossmann, and Heiner Stuckenschmidt. 2017. Cluster-based hierarchical demand forecasting for perishable goods. *Expert systems with applications* 76 (2017), 140–151.

[12] Amir Mahmud Husein, Muhammad Arsyal, Sutrisno Sinaga, and Hendra Syahputa. 2019. Generative adversarial networks time series models to forecast medicine daily sales in hospital. *Sinkron: jurnal dan penelitian teknik informatika* 3, 2 (2019), 112–118.

[13] 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.

[14] Zeynep Hilal Kilimci, A Okay Akyuz, Mitat Uysal, Selim Akyokus, M Ozan Uysal, Berna Atak Bulbul, and Mehmet Ali Ekmis. 2019. An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. *Complexity* 2019 (2019).

[15] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[16] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[17] Balakrishnan Lakshmanan, Palaniappan Senthil Nayagam Vivek Raja, and Viswanathan Kalathiappan. 2020. Sales demand forecasting using LSTM network. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, 125–132.

[18] Agnieszka Lasek, Nick Cercone, and Jim Saunders. 2016. Restaurant sales and customer demand forecasting: Literature survey and categorization of methods. *Smart City 360°: First EAI International Summit, Smart City 360°, Bratislava, Slovakia and Toronto, Canada, October 13-16, 2015. Revised Selected Papers 1* (2016), 479–491.

[19] Yinfeng Li, Chen Gao, Quanming Yao, Tong Li, Depeng Jin, and Yong Li. 2022. DisenHCN: Disentangled Hypergraph Convolutional Networks for Spatiotemporal Activity Prediction. *arXiv preprint arXiv:2208.06794* (2022).

[20] Ziyue Li, Nurettin Dorukhan Sergin, Hao Yan, Chen Zhang, and Fugee Tsung. 2020. Tensor completion for weakly-dependent data on graph for metro passenger flow prediction. In *proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4804–4810.

[21] 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.

[22] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[23] Shaohui Ma, Robert Fildes, and Tao Huang. 2016. Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra-and inter-category promotional information. *European Journal of Operational Research* 249, 1 (2016), 245–257.

[24] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[25] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, Elena Baralis, Michele Osella, and Enrico Ferro. 2018. Knowledge graph embeddings with node2vec for item recommendation. In *The Semantic Web: ESWC 2018 Satellite Events: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers 15*. Springer, 117–120.

[26] Rahul Priyadarshi, Akash Panigrahi, Srikanta Routroy, and Girish Kant Garg. 2019. Demand forecasting at retail stage for selected vegetables: a performance analysis. *Journal of Modelling in Management* 14, 4 (2019), 1042–1063.

[27] B Sri Sai Ramya and K Vedavathi. 2020. An advanced sales forecasting using machine learning algorithm. *International Journal of Innovative Science and Research Technology* 5, 5 (2020), 342–345.

[28] Kamran Raza. 2017. Prediction of Stock Market performance by using machine learning techniques. In *2017 International conference on innovations in electrical engineering and computational technologies (ICIEECT)*. IEEE, 1–1.

[29] Shuyun Ren, Hau-Ling Chan, and Tana Siqin. 2020. Demand forecasting in retail operations for fashionable products: methods, practices, and real case study. *Annals of Operations Research* 291 (2020), 761–777.

[30] Dennis Reynolds, Imran Rahman, and William Balinbin. 2013. Econometric modeling of the US restaurant industry. *International Journal of Hospitality Management* 34 (2013), 317–323.

[31] Kimberly F Sellers and Galit Shmueli. 2010. Predicting censored count data with COM-Poisson regression. *Robert H. Smith School Research Paper No. RHS-06-129* (2010).

[32] Mahya Seyedan and Fereshteh Mafakheri. 2020. Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data* 7, 1 (2020), 1–22.

[33] Juliana C Silva, Manuel C Figueiredo, and Ana C Braga. 2019. Demand forecasting: A case study in the food industry. In *Computational Science and Its Applications–ICCSA 2019: 19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part III 19*. Springer, 50–63.

[34] Haiyan Song, Richard TR Qiu, and Jinah Park. 2019. A review of research on tourism demand forecasting: Launching the Annals of Tourism Research Curated Collection on tourism demand forecasting. *Annals of Tourism Research* 75 (2019), 338–362.

[35] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 269–278.

[36] Theodore Tskeris and Charalambos Tsekeris. 2011. Demand forecasting in transport: Overview and modeling advances. *Economic research-Ekonomska istraživanja* 24, 1 (2011), 82–94.

[37] Jiaxing Wang, GQ Liu, and Lu Liu. 2019. A selection of advanced technologies for demand forecasting in the retail industry. In *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*. IEEE, 317–320.

[38] 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.

[39] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2377–2386.

[40] Yue Yu, Tong Xia, Huandong Wang, Jie Feng, and Yong Li. 2020. Semantic-aware spatio-temporal app usage representation via graph convolutional network. *UbiComp* 4, 3 (2020), 1–24.

[41] Vincent W Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Twenty-fourth AAAI conference on artificial intelligence*.

[42] 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.

A APPENDIX FOR REPRODUCIBILITY

A.1 Model Implementation Details

In this section we provide implementation details of our system. Due to the massive amount of training data and the high demand for low latency in our online system, we simplify each network within the framework. In our final implementation, feature merging network h^M is a one-layer fully-connected network of 120 output units (120D FC). User preference network h^U is a 340D FC. Shared network E^S and expert networks E^W/E^N are 256D FCs. Need prediction network t^N is a 10D FC. In-store/delivery classification network t^W is a two-layer perceptron, which has 10 hidden units and 2 output units. Complexifying these neural networks can help us further optimize performance, but at the same time, it would lead to an increase in system latency. The activation function of all mentioned networks is ReLU [9]. We adopt batch normalization [13] right after h^M, h^U, E^S, E^P, and E^N. Following such implementation, we conduct rich online and offline experiments to prove the effectiveness of our model, which are shown in Section 4 and Section 5.

A.2 Dataset

We conduct our offline experiment on a real-world dataset at the scale of billions. The dataset is a sampling of all purchase records in 2022 on the platform. We sample the records according to the percentage of purchases of each kind of life service. The dataset includes more than 7 billion real purchase records from 65 million users. Each instance in the dataset includes user profile, time, location, and other real-time environmental factors, and the kind of life service the user purchase. The type of life service consumed by the user reflects their actual living need in the spatiotemporal context. Following existing works [4, 25], we randomly sample 80% of the dataset as the training set, and 20% as the test set.

A.3 Metrics

We design a metric named Sort Accuracy (SA) to measure the performance of systems on our problem. The metric SA can be defined as follows. For user scene i, our system outputs scores for all kinds of living needs. We sort the living needs by their scores and get a list. We define *Relative Ranking Error* as the difference between the actual ranking position and the ideal ranking position (the first position) of the ground truth need, and further define *Maximum Ranking Error* as the maximum relative ranking error any system can give for a user scene (the number of categories of user needs - 1). For example, in our system which handles 10 types of living need, for user scene i, if the ground truth need is ranked third, then the relative ranking error is 2 (2 = 3-1), and the maximum ranking error is 9 (9 = 10-1). Then Sort Accuracy (SA) can be defined as follows,

$$\text{SA} = \text{average}_{i \in T} \left(1 - \frac{\text{Relative Ranking Error}}{\text{Maximum Ranking Error}}\right) \tag{14}$$

where T denotes the testing set on which the metric is calculated. We first calculate 1−Relative Ranking Error/Maximum Ranking Error of every user scene in the testing set and then compute the average. We then define Via-delivery Sort Accuracy (VDSA) and In-store Sort Accuracy (ISSA) to measure systems' performance on delivery and in-store living needs.

$$\text{VDSA} = \text{average}_{i \in T_{VD}} \left(1 - \frac{\text{Relative Ranking Error}}{\text{Maximum Ranking Error}}\right) \tag{15}$$

$$\text{ISSA} = \text{average}_{i \in T_{IS}} \left(1 - \frac{\text{Relative Ranking Error}}{\text{Maximum Ranking Error}}\right) \tag{16}$$

T_{VD} and T_{IS} denote sets of testing samples where the ground truth needs-meeting way is via delivery and in store, respectively. In the following, we will use these metrics to measure the living needs prediction performance of our system and baseline systems.

A.4 Baselines

To illustrate the effectiveness of our system, we compare it with two baselines widely in actual production environments, including **DIN [42]**, **DNN [5]**, **DCN [38]**, **ESMM [24]**, and **MMOE [22]**. We will provide a detailed description of these baselines in the appendix. DIN is a recommendation algorithm that leverages deep neural networks to analyze users' historical behavior and make predictions about their potential interests. It uses an attention-based mechanism to weigh the importance of different historical behaviors for predicting the current interest of a user. As for DNN, We follow the design of Wide & Deep learning to build a Deep Neural Network (DNN) based system for our task. It can learn both simple and complex relationships in the data. DCN is proposed to keep the benefits of a DNN model while introducing a cross network that is more efficient in learning certain bounded-degree feature interactions. It applies feature crossing at each layer, and it doesn't require manual feature engineering, adding minimal extra complexity to the DNN model. ESMM estimates post-click conversion rates for recommendation systems by using sequential user actions and a feature representation transfer learning strategy to alleviate sample selection bias and data sparsity. MMoE is a multi-task learning approach that learns to model task relationships from data. It adapts the Mixture-of-Experts (MoE) structure to multi-task learning by sharing the expert submodels across all tasks, while also having a gating network trained to optimize each task. For a clear comparison, the input features for all baselines are kept the same as those for our model.

A.5 Additional Ablation Study

To gain a deeper understanding of the impact of each component design of our model, we also conduct ablation studies with a particular focus on the effects of the group behavior pattern features and the spatiotemporal context features. When removing the group behavior pattern features from the model input, the performance decreased by 1.06%. When removing the spatiotemporal context features from the model input, the performance decreased by 1.46%. These results provide compelling evidence of the pivotal role that these two features play in accurately predicting users' daily living needs.

A Hybrid Approach of Statistics and Embeddings for Multilingual and Multi-Locale Recommendation

Weijia Zhang*†
zhangweijia@meituan.com
Meituan
Shanghai, China

Jin Zhan*
senkin13@gmail.com
DataRobot
Tokyo, Japan

Zhongshan Huang*
huang4211819@126.com
No affilation
Beijing, China

Lu Wang
wlu@microsoft.com
Microsoft
Beijing, China

Qiang Wang
qiang.wang@meituan.com
Meituan
Shanghai, China

ABSTRACT

To encourage the development of multilingual recommendation systems, Amazon pushlished a multilingual and multi-locale shopping session dataset, and KDD Cup 2023 challenge on Multilingual Session Recommendation Challenge was hosted based on this dataset.

In this paper, we present our solution for this competition. Following a widely-used setting in recommender system, our solution consists of two stages: recalling and ranking. In the first stage, we utilize various recalling strategies to retrieve a set of candidate products, including covisit matrix based collective filtering, graph embedding based I2I searching, text transformer based I2I searching and BPR based U2I searching. In the second stage, we develop a model to predict the probability of each user engaging with the candidate products. This model is an ensemble of two Catboost models, which include various statistical features and embedding similarity features. Finally, we achieved 4th place in Task1 and 3rd place in Task2.

CCS CONCEPTS

• **Information systems → Personalization**.

KEYWORDS

recommender systems, sequential recommendation

ACM Reference Format:
Weijia Zhang, Jin Zhan, Zhongshan Huang, Lu Wang, and Qiang Wang. 2018. A Hybrid Approach of Statistics and Embeddings for Multilingual and Multi-Locale Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

*Equivalent contributions.
†Corresponding Author

Unpublished working draft. Not for distribution.

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.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

1.1 Background

Recommender systems, also known as RecSys, play a crucial role in various online services, including e-commerce, social media, news platforms, and online video streaming. Among the different tasks performed by these systems, session-based recommendation, which utilizes customer session data to predict their next interaction, stands out as an essential function.

However, session-based recommendation in the realm of real-world scenarios involving multilingual and imbalanced data has received limited attention in previous studies.

To bridge this research gap, Amazon introduced the "Multilingual Multi-locale Shopping Session Dataset"[3], which comprises millions of user sessions originating from six distinct locales, and hosted KDD Cup 2023 on Multilingual Session Recommendation Challenge based on this dataset.

1.2 Datasets And Tasks

The dataset encompasses products in 6 languages, namely English, German, Japanese, French, Italian, and Spanish. Notably, the dataset exhibits an imbalanced distribution, with a lower number of French, Italian, and Spanish products compared to English, German, and Japanese, as shown in Table 1.

With this dataset, three tasks were introduced:

- *Next Product Recommendation*: Given a session data and the attributes of each product, the goal of this task is to predict the next product that a customer is likely to engage with. The test set for Task 1 comprises data from popular locales, i.e., English, German, and Japanese locales.
- *Next Product Recommendation for Underrepresented Languages and Locales*: The goal of this task is similar to Task1 while the test set is constructed from the underepresented locales, i.e., French, Italian, and Spanish.
- *Next Product Title Generation*: Given a session data and the attributes of each product, the goal of this task is to predict the title of the next product that a customer will engage with. Unlike Tasks 1 and 2, which focus on recommending existing products, predicting new or "cold-start" products presents a unique challenge.

Our primary focus was on Task1 and Task2. Section 2-5 of this paper will provide a detailed description of our solution for Task1

	Locale	# of session	# of items	# of interactions
Task1	English	1,182,181	500,180	4,872,506
	German	1,111,416	518,327	4,836,983
	Japanese	979,119	395,009	4,388,790
Task2	French	117,561	44,577	416,797
	Italian	126,925	50,461	464,851
	Spanish	89,047	42,503	326,256

Table 1: Details of the Dataset

and Task2, encompassing preprocessing methods, recall strategies and ranking models. Furthermore, the experimental results will be presented in Section 6.

2 OVERVIEW

As depicted in Figure 1, our solution is composed of two main stages: recalling and ranking. Although these stages are distinct, they share a significant amount of data. Therefore, we first generate multiple matrices and embeddings, which serve as the foundation for both recalling and ranking.

The first stage is candidate item recalling, which involves retrieving a set of potential products that each user may be interested in from the entire pool of products. This is achieved by utilizing the preprocessed matrices and embeddings. Approximately 200 items are retrieved for each user.

The second stage is candidate item ranking, where we predict the probability of a user engaging with each retrieved item. The items are then sorted based on these probabilities, and the top 100 items with the highest probability are recommended.

3 PREPROCESS

To achieve effective recall and ranking, we utilize a variety of features and models in both processes. In this section, we will elaborate on the generation of these commonly used data, which form the fundamental basis for our recall and ranking methods.

3.1 Covisit Matrix

One of the key features we utilize is the fact that items visited within the same user session often exhibit similarities. To capture this, we constructed a co-visit matrix where each entry represents the number of times item i and item j were visited together within the same user session.

Furthermore, we devised three types of weights to address biases that may arise from different user behaviors. The formula for calculating each entry of the co-visit matrix is as follows:

$$e(i,j) = \sum_{u \in U} \sum_{(i,j) \in u} \frac{1}{W_p(P_{i,j}) + W_c(C_i) + W_c(C_j) + W_l(L_s)} \quad (1)$$

$$W(x) = x^{\alpha} \quad (2)$$

- i and j represent item i and item j, respectively, while s and S denote user session u and the full session set, respectively.
- The closer the items are in the session, the higher their similarity. We use $P_{i,j}$ to denote the positional difference between item i and item j and we use different α values in $W_p(\cdot)$ to promote diversity, like 0.7, 1.0 and so on;
- Popular items, which are visited more frequently, have greater opportunities to calculate similarity with other items. To address this, we introduce a weight factor. C_i and C_j denote the total visited count of item i and item j in the training data, and we use different α values in $W_c(\cdot)$ to promote diversity, like 0.4, 1.0 and so on;
- Covisit in the long sessions also need to be penalized. To address this, we introduce another weight factor. L_s denotes the length of $session_s$ and we use different α values in $W_l(\cdot)$ to promote diversity, like 0.7, 1.0 and so on.

3.2 Text Transformer

In this dataset, there are several text features associated with items, such as title, description, color, material, and more. To leverage these textual information, we employed transformers using two approaches:

- Pretrained sentence transformer[6]. We use a pretrained sentence transformer for multilingual text [1] to encode each attribute of an item into a vector.
- Fine-tuned transformer with contrastive loss. To improve generalization in the competition dataset, we fine-tuned the transformer by referring to the method described in [4]. First, we formulated an item as a "sentence" by flattening its key-value attributes described by text. Then, we used a transformer to encode this "sentence" into a vector, denoted as h_s. Finally, a contrastive loss is utilized to fine-tune the transformer:

$$L = -log \frac{e^{sim(h_s, h_i^+)/\tau}}{\sum_{i \in \beta} e^{sim(h_s, h_i)}} \quad (3)$$

where sim is the cosine similarity function; h_i^+ is the embedding of the next item to $item_s$ in the user session; β is all items in the batch and τ is a temperature parameter.

3.3 Graph Embedding

Because of the ability to utilize high order connection, graph neural network has been widely used in the recommender system[2, 5, 8]. In our approach, we first build a large item graph, where the nodes represent items and an edge consisting of <item i, item j> denotes a successive visit of item i and item j. We then use ProNE[9] to extract a 2000-dimensional graph embedding for each item. The larger the embedding dimension, the better the performance, but it also slows down the process. We chose ProNE for several reasons:

- ProNE uses Sparse Randomized tSVD for fast embedding, which makes the process faster.
- ProNE uses Spectral Propagation for embedding enhancement, which makes the embeddings more accurate.

3.4 Item2vec

We treat all the Items in the same session as a sentence, and then train a word2vec skip-gram model to extract item embeddings.

[1] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

Unpublished working draft. Not for distribution.

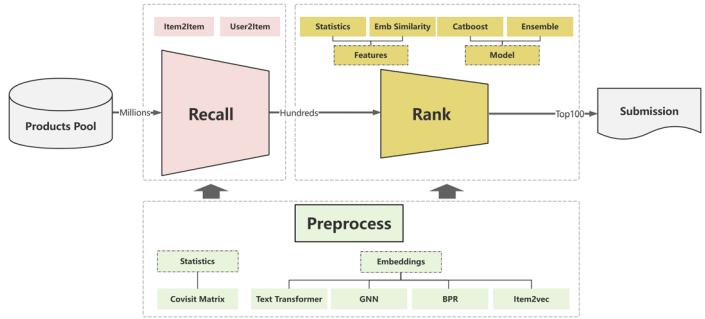


Figure 1: Architecture of our solution.

3.5 Bayesian Personalized Ranking

Bayesian Personalized Ranking[7], a.k.a BPR, is a commonly used matrix factorization method for learning personalized rankings from implicit feedback. It uses stochastic gradient descent (SGD) to optimize a pairwise ranking objective function, as shown in the Equation 4:

$$argmax_\theta l(\theta) = \sum ln\sigma(x_{ui} - x_{uj}) + \lambda||\theta||^2 \quad (4)$$

where i is the postive item and j is the random sampled negative item; θ is the embedding table of users and items and x_{ui} is the similarity of user embedding and item embedding; σ is the sigmoid function and λ is the regularization parameter.

4 RECALLING

In this section, we explain the process of recalling candidate products, which is based on the co-visit matrix and various embeddings described in Section 3.

4.1 Covisit Matrix Based Collaborative Filtering

The top N items that are most similar to the ones that the user has interacted with will be recalled, where the covisit matrix based similarity between users and items is as follows:

$$Sim(u, i) = \sum_{j \in u} W_p(p_j)e(i, j) \quad (5)$$

where u and i represent user u and item i, respectively; $W_p(\cdot)$ is defined in the Equation 2, with a α value of 1.0; p_j represents the position of item j in the user session, with larger values indicating later positions; $e(i, j)$ is defined in the Equation 2.

4.2 Item Embedding Based I2I searching

The recall strategy employed here is similar to collaborative filtering based on covisit matrices. The only difference lies in the calculation of similarity, which is based on the cosine similarity between the embeddings of item i and item j. These embeddings are generated using techniques decribed in the previous sections, such as text transformer embeddings, graph embeddings, and item2vec embeddings.

4.3 BPR Embedding Based U2I searching

This recall strategy involves retrieving items whose embeddings, generated using BPR, are similar to the user's embedding, also generated using BPR.

5 RANKING

We use a popular GBDT model, Catboost [1], to rank the recalled candidate items. In this section, we will provide details about the ranking model.

5.1 Statistical Features

We create statistical features in three aspects: item, user, and user crossing item.

- Item Statistics: Count of interactions in each locale, target locale and all the locales; price in each locale and target locale.
- User Statistic: Nunique and count of interacted items and their brand, color, size, model, and author; mean, median, max, and min price of interacted items.
- User Item Cross Statistics:
  - Count of candidate item's brand, color, size, model and author in user-interacted items.
  - Max, mean, min, sum, last and position-weighted sum of covisit score of session's items and candidate item.
  - Min and last of character-level ASID difference of user-interacted items and candidate item.
  - Max, mean, min, sum, last and position-weighted sum of character-level LCS (Longest contiguous matching subsequence) distance of user-interacted items and candidate item.
  - The rank number of the candidate item under each recalling strategies.

5.2 Embedding Similarity Features

The similarity between the user and candidate item, based on the embeddings described in Section 3, show a very impressive effect.

Unpublished working draft. Not for distribution.

- I2I embedding similarity. This refers to the similarity between the candidate item and the user interacted items. Equation 6 represents the process of calculating I2I similarity.

$$Sim(u, i) = Aggr(cos_sim(e_i, e_j)|j \in u) \quad (6)$$

  where $Aggr$ is the aggregation function, such as max, mean, min, and sum. e_i and e_j are the embeddings of item i and item j, which can be graph embeddings, text transformer embeddings, and item2vec embeddings.
- U2I embedding similarity. As we factorize user and item with BPR, it's natural to use the similarity of user embedding and item embedding as a feature.

5.3 Model

The ranking model in our final submission is an ensemble of two models from different teammates. One model is from Jin Zhan and Zhongshan Huang, and the other is from Weijia Zhang. To ensure diversity, we did not collaborate much until the end of the competition. Below are the details of our individual models and the ensemble method.

5.3.1 Single Model. The first aspect of the model is the training samples. Recalled candidate items that the user did not interact with are labeled with 0, while those that the user interacted with are labeled with 1. Due to significant class imbalance, downsampling is applied. The negative samples are kept at a quantity of n times the length of the positive samples. In Jin Zhan and Zhongshan Huang's model, n is 10, while in Weijia Zhang's model, n is 20. As the data for Task2 is much smaller than that of Task1, we found that training Task2's model with all the samples and training Task1's model with only Task1's data gave the best score.

5.3.2 Ensemble. For more diversity, our models have different features and make predictions on candidate items from different recall strategies. Regarding the ensemble method, we first obtain the top 100 items of each user from the prediction of each model. Then, we apply a weighted blending on the scores, where the weight of Jin Zhan and Zhongshan Huang's model is 7 and that of Weijia Zhang's model is 3.

6 EXPERIMENTS

6.1 Overall Performance

The performance of our models is listed in Table 2, where model A is trained by Jin Zhan and Zhongshan Huang, and model B is trained by Weijia Zhang. Model A also includes some of Weijia Zhang's features. In task 1, model A scores 0.4030, model B scores 0.3968, and the merged model scores 0.4047. In task 2, model A scores 0.4580, model B scores 0.4468, and the merged model scores 0.4601.

Models	Task1	Task2
A	0.4030	0.4580
B	0.3968	0.4468
Ensemble	0.4047	0.4601

Table 2: Overall Performance

6.2 Feature Importance

Model A is trained with all of Jin Zhan, Zhongshan Huang, and Weijia Zhang's features. Therefore, we present the importance of the top 30 features of model A in Figure 2.

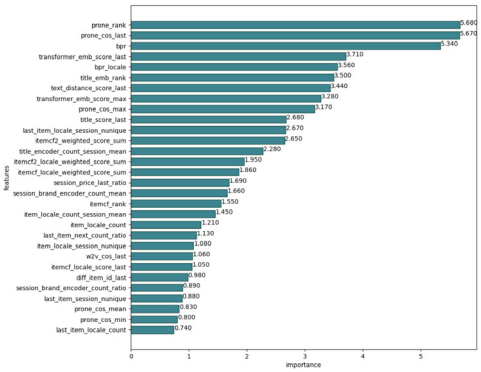


Figure 2: Top 30 Features' Importance.

REFERENCES

[1] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).

[2] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[3] Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, Zhen Li, Monica Xiao Cheng, Rahul Goutam, Haiyang Zhang, Karthik Subbian, Suhang Wang, Yizhou Sun, Jiliang Tang, Bing Yin, and Xianfeng Tang. 2023. Amazon-M2: A Multilingual Multi-locale Shopping Session Dataset for Recommendation and Text Generation. (2023).

[4] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. arXiv:2305.13731 [cs.IR]

[5] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.

[6] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[7] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*.

[8] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[9] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. Prone: Fast and scalable network representation learning.. In *IJCAI*, Vol. 19. 4278–4284.

Unpublished working draft. Not for distribution.

Divide and Adapt- Active Domain Adaptation via Customized Learning

Duojun Huang[1,2] Jichang Li[3] Weikai Chen[4] Junshi Huang[5] Zhenhua Chai[5] Guanbin Li[1,2†]
[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]Research Institute, Sun Yat-sen University, Shenzhen, China
[3]The University of Hong Kong [4]Tencent America [5]Meituan
huangdj9@mail2.sysu.edu.cn, csjcli@connect.hku.hk, chenwk891@gmail.com
{huangjunshi,chaizhenhua}@meituan.com, liguanbin@mail.sysu.edu.cn

Abstract

Active domain adaptation (ADA) aims to improve the model adaptation performance by incorporating active learning (AL) techniques to label a maximally-informative subset of target samples. Conventional AL methods do not consider the existence of domain shift, and hence, fail to identify the truly valuable samples in the context of domain adaptation. To accommodate active learning and domain adaption, the two naturally different tasks, in a collaborative framework, we advocate that a customized learning strategy for the target data is the key to the success of ADA solutions. We present Divide-and-Adapt (DiaNA), a new ADA framework that partitions the target instances into four categories with stratified transferable properties. With a novel data subdivision protocol based on uncertainty and domainness, DiaNA can accurately recognize the most gainful samples. While sending the informative instances for annotation, DiaNA employs tailored learning strategies for the remaining categories. Furthermore, we propose an informativeness score that unifies the data partitioning criteria. This enables the use of a Gaussian mixture model (GMM) to automatically sample unlabeled data into the proposed four categories. Thanks to the "divide-and-adapt" spirit, DiaNA can handle data with large variations of domain gap. In addition, we show that DiaNA can generalize to different domain adaptation settings, such as unsupervised domain adaptation (UDA), semi-supervised domain adaptation (SSDA), source-free domain adaptation (SFDA), etc.

1. Introduction

Domain adaptation (DA) approaches strive to generalize model trained on a labeled source domain to a target domain with rare annotation [5, 13, 16, 24] by coping with the domain disparity. Nevertheless, DA methods are significantly outperformed by their supervised counterparts due to the scarceness of annotation as demonstrated in [4, 14, 27]. In practice, it is cost-effective to get a moderate amount of target samples labeled to boost the performance of domain adaptation. Active learning (AL) approaches seek to select samples with uncertainty [9, 12, 29, 30] and diversity [19, 25] to best benefit the model, which properly matches the demand. However, previous AL methods assume that both the labeled and unlabeled data follow the same distribution, such a strategy may become ineffective to the DA scenarios where the target data suffer from domain shift. The recently proposed active domain adaptation (ADA) [4, 22, 31] aims to resolve this issue by actively selecting the maximally-informative instances such that the performance of the transferred model can be best boosted with a limited annotation budget.

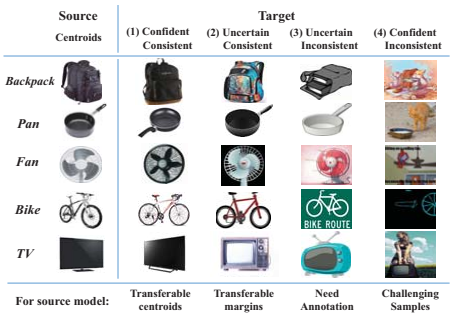


Figure 1. The illustration of our proposed Divide-and-Adapt mechanism to divide target samples into different data subsets for customized learning.

The key to the success of ADA is to strike a good balance between the highly coupled yet inherently different tasks: active learning and domain adaptation. The real-world target data typically exhibit either of the two characteristics:

†Corresponding author is Guanbin Li.

source-like or *target-specific*. While the former has similar feature distribution with the source data, the latter tends to be the unique part of target domain and deviates greatly from the source distribution [4, 26, 38]. On one hand, to achieve domain adaptation, applying the same adaptation strategy to all target data equally cannot generalize well to scenarios with varying degrees of domain shift. This is particularly true when the gap between the source-like and target-specific data is unknown. On the other hand, in active learning tasks, the samples with a learning difficulty will be more likely to be selected for labeling. Nonetheless, with large domain gap, incorporating such difficult samples in the adaptation task would hamper the learning of the adaptation model, making the training highly unstable. However, despite the impressive progress that has been made, none of the existing works has fully addressed the above issues.

In this work, we propose Divide-And-Adapt (DiaNA), a novel ADA framework that can scale to large variations of domain gaps while achieving cost-effective data labeling with a significant performance boost for domain adaptation. Our key observation is that customized learning strategies are vital for target data with different characteristics. In particular, DiaNA divides the target data into four subsets with different levels of transferable properties (see Figure 1), each of which is handled with a customized learning strategy. Unlike traditional AL methods that would simply label the most uncertain data [29, 30, 37], we propose to withhold the most challenging samples (Figure 1 category (4)) for training the domain adaption models. Instead, the selected samples for active annotation would maintain a proper stimulus for the source model, providing informative domain knowledge without jeopardizing the training stability. The subdivision of target data is dynamically updated as the domain disparity is gradually mitigated with more labeled data. Hence, the previous challenging samples could be classified as transferable in the later stage and exploited in the network training.

We introduce a novel protocol for subdividing the target samples for customized learning. In addition to the uncertainty of model prediction, we advocate that the consistency with the learned prior distribution, *i.e.* the domainness, is another key criterion for active domain adaptation [4, 26]. To this end, we divide the target data into four categories as shown in Figure 1 according to the domainness and uncertainty of the instances. We further propose that the samples with 1) being *uncertain* to the model and 2) having *inconsistent* prediction with the label of its closest category prototype in the learned feature space (*i.e.* high domainness) are the most "profitable" instances for bringing informative knowledge of target domain if annotated. Thereby, we identify the *uncertain inconsistent* samples for labeling while applying tailored learning strategies for the remaining categories to boost the selectivity of the sampling.

To avoid heuristic thresholding for data subdivision, we propose an automatic data sampling mechanism based on Gaussian mixture model (GMM). In particular, we propose an *informativeness function* that incorporates the domainness and uncertainty in a unified scoring system. The computed informativeness score of the labeled data is used to train a four-component GMM model, which is then applied to sample the unlabeled target data into four categories.

We evaluate DiaNA over a large variety of domain shift scenarios on DomainNet [21], Office-Home [28] and CIFAR-10 [11]. Furthermore, the proposed sampling strategy of DiaNA can be generalized to various domain adaption problems with different supervision settings, including unsupervised domain adaptation (UDA), semi-supervised domain adaptation (SSDA), and source-free domain adaptation (SFDA).

We summarize our contributions as follows:

- A general "divide-and-adapt" framework, coded DiaNA, for active domain adaptation that can handle diversified degrees of domain gaps while being able to generalize to different domain adaptation problems, including UDA, SSDA, and SFDA.
- A new target data partition strategy based on domainness and uncertainty that enables stratified learning to achieve more stable training, superior adaptation performance, and better generality.
- A novel informativeness scoring system and the corresponding sampling paradigm based on GMM model for automatic data partitioning.
- New state-of-the-art performance over the mainstream public datasets in the task of active domain adaptation.

2. Related Work

Domain adaptation (DA). Most of the prior methods involving domain adaptation follow the unsupervised domain adaptation (UDA) paradigm that only unlabeled data is accessible in the target domain. Early UDA approaches [16, 18, 32] utilized maximum mean discrepancy (MMD) to minimize the discrepancy of features from different domains to address the domain shift, while recent adversarial learning based methods [5, 17] have become popular solutions for UDA. Despite effectiveness, UDA is still significantly outperformed by its supervised variant as indicated in [4, 14, 27]. Semi-supervised domain adaptation (SSDA) [13, 24, 35] allows a few target instances to be labeled and generally achieves improved performance. However, SSDA assumes that the labeled instances are passively provided in advance, depriving the choice of informative samples beneficial to the model. In this work, we propose an active domain adaption method to select the most informative samples, which can be incorporated into other DA methods to further boost their performance.

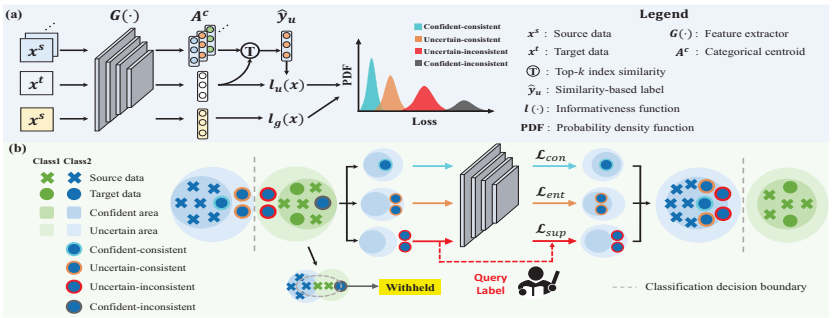


Figure 2. We propose a general "divide-and-adapt" framework, named DiaNA, for active domain adaptation. (a) A new target data partition strategy is presented to divide target data into four categories via building an informativeness scoring function that incorporates both domainness and uncertainty metrics. Besides, an automatic data partitioning function based on a learned four-component GMM model is applied to adaptively separate four sample categories from unlabeled target data. (b) To achieve customized learning, different auxiliary techniques are integrated into model fine-tuning w.r.t the four categories of unlabeled target data.

Active Learning (AL). Active learning is proposed to select a maximally-informative subset of unlabeled data for annotation via a query function, in an effort to best improve the model performance with a limited annotation budget. There exist several efforts to tackle the challenges involving active learning. For example, the uncertainty-based AL methods mainly focus on designing query functions based on prediction confidence [30], entropy [9, 29], or margin [10, 23] of the posterior probabilities. On the other hand, the diversity-based approaches [6, 19, 25] concentrate on selecting samples that can well represent the entire dataset. Meanwhile, expected model change-based methods [3, 15] aim to query the instances that would lead to a significant change to the current model. Recently proposed active learning approaches attain better performance by utilizing hybrid strategies which taking multiple metrics into consideration [1, 20]. Notwithstanding, the efficacy of previous AL resources may falter in DA scenarios as their selection strategy fails to meticulously consider the implications of the domain gap.

Active Domain Adaptation (ADA). Early ADA researches [4, 26] propose to measure the uncertainty and domainness of each target instance via a domain discriminator with adversarial training. CLUE [22] presents an entropy-weighted clustering algorithm to query uncertain and diverse samples in the target domain. Recently, SDM [33] is proposed to optimize a margin loss function for exploring target instances similar to potential hard samples in the source domain. Nevertheless, the majority of existing ADA methods deliberately design hand-crafted query functions to evaluate sample's annotation value and apply the same adaptation strategy to all target data equally [4, 26, 31]. The rigid criteria make them easily overfit to certain domain adaption scenarios, which limits the generalization of the method. In contrast, we propose to integrate uncertainty and domainness metrics within a unified scoring function to select informative samples. Moreover, we apply tailored learning strategies for target data with different characteristics, constructing an effective framework capable of generalizing across large variations of domain shifts.

3. Divide-and-Adapt Framework

Problem formulation. Active domain adaptation aims at seeking an optimal model for a target domain when given labeled source data $\mathcal{S} = \{(x_s, y_s)\}$ as well as unlabeled target data $\mathcal{U} = \{x_u\}$, assisted by the iteratively queried labeled data $\mathcal{T} = \{(x_t, y_t)\}$ of the target domain. Here, $\mathcal{S}$, $\mathcal{U}$, and $\mathcal{T}$ share the same label space over C categories. Initially, $\mathcal{T}$ is an empty set. The model is iteratively trained using $\mathcal{S}$, $\mathcal{T}$, and $\mathcal{U}$ for R active learning loops until a given annotation budget B is reached. In each active learning loop, $b = B/R$ samples from $\mathcal{U}$ would be selected, labeled by human experts, and then moved into $\mathcal{T}$. The proposed Divide-and-Adapt framework consists of a feature extractor and a classifier. Our goal is to design a query function to identify informative samples to annotate and utilize limited amount of labeled samples to best enhance the classification performance for the target domain.

Overview. In this paper, we propose Divide-And-Adapt, coded DiaNA, to tackle the problem of active domain adaptation. In detail, we first propose to design an informativeness function for subdividing the target samples into four categories, one of which is the informative target samples used for actively annotating. Such a function is a unified scoring system that incorporates the domainness and the uncertainty. To avoid heuristic thresholding for data subdivision, we propose an automatic data sampling mechanism to divide the unlabeled target data into four

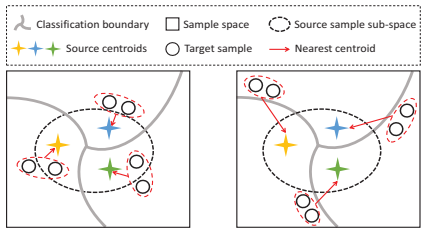


Figure 3. (a) Source-like target samples are more likely to be close to the source centroid in the feature space with respect to its predicted category. (b) Target-specific target samples are prone to be inconsistent between model predictions and similarity-based labels.

sample categories. This is achieved by learning a four-component Gaussian mixture model using semi-supervised Expectation-Maximization algorithm. Finally, we design tailored learning strategies for different categories of target samples during model fine-tuning, thus further improving the model performance. The overview of the proposed algorithm has been summarized in Figure 2.

3.1. Informativeness Scoring Mechanism

Following previous ADA works [4, 26, 31], we aim to search for the target instances with high domainness and uncertainty. To achieve this goal, we propose to first estimate category-wise centroids based on the labeled samples, and then define a similarity-based label for each unlabeled target sample by computing its top-k feature similarity with the categorical centroids. Finally, we integrate both the domainness and the uncertainty metrics into an informativeness function, which is formulated as the consistency between model prediction and similarity-based label.

Categorical centroids. We propose to average the features of labeled samples to estimate the category-specific data distribution, i.e., category prototype. Here, we formulate the centroid with respect to the c-th category as follows,

$$A^c = \frac{\sum_{(x,y)\in\mathcal{S}} \mathbb{1}\{y=c\}\cdot G(x)}{\sum_{(x,y)\in\mathcal{S}} \mathbb{1}\{y=c\}}, \tag{1}$$

where $\mathbb{1}\{\cdot\}$ is an indicator function, and $G(x)$ is a feature extractor that outputs the sample feature of x. Note that, as the active learning loops proceed, labeled target samples would be incorporated into the calculation of categorical centroids to mitigate the impact of distribution change.

Similarity-based data label. As introduced in the works [7, 13], samples share more top-k indices in their respective lists of rank-ordered feature elements would have a higher probability of being the same category. To obtain the similarity-based label for each target instance, we first measure pairwise similarity of the indices of the sorted feature elements based on the magnitude. Then, we can formulate a similarity-based label for a sample x from $\mathcal{U}$ as follows,

$$\ddot{y}(x) = \underset{1\le c\le C}{\operatorname{argmax}} \ \mathbf{IoU}(\operatorname{top}k(G(x)), \operatorname{top}k(A^c)), \tag{2}$$

where top$k(\cdot)$ computes the top-k feature element indices ranked according to their magnitudes. In addition, $\mathbf{IoU}(\cdot,\cdot)$ denotes an Intersection-over-Union function, which returns the extent of overlap of the two respective index lists.

We set the value of k to be significantly smaller than the full dimension of the feature vector. In this way, the function top$k(\cdot)$ only extracts the principle components of the feature of sample. Therefore, the IoU function in Eq. 2 is equivalent to measuring pairwise image sample similarity under a low-resolution condition. In ADA tasks, the training of model is inevitably dominated by the source domain especially in the early stage [33]. Therefore, source-like target samples that are similar to the source domain would naturally have more accurate representation than the target-specific ones that are distinct from the source. Under the low-resolution condition, i.e., k set to be small, source-like samples tend to have the same similarity-based label as the model predicted class thanks to their reliable and discriminative features extracted by the model. In contrast, target-specific samples are prone to produce inconsistent results since they are typically underfitted by the current model as shown in Figure 3. Based on the analysis above, we utilize the consistency of the model predicted class and similarity-based label to evaluate the domainness of each target sample. Detailed theoretical and experimental supports are demonstrated in our supplementary.

Informativeness function. To explore the most informative target samples, we integrate the domainness and uncertainty into a unified informativeness function (InfoF). The informativeness score for a sample x from unlabeled target domain can be formulated as follows,

$$\ell_u(x) = -\sum_{c=1}^{C} \mathbb{1}\{c=\ddot{y}(x)\}\cdot\log P_c(x), \tag{3}$$

where $P(x)$ denotes the probabilistic prediction of a sample x from the network model and $P_c(x)$ is the c-th element of the vector $P(x)$. According to Eq. 3, we can divide unlabeled target samples into two subsets depending on whether the model predicted class and the similarity-based label are identical: consistent and inconsistent subsets, while both sets have confident samples and uncertain samples depending on whether the predicted probability is greater than a given confidence threshold. We name these four categories of unlabeled target data as *confident-consistent (CC)*, *uncertain-consistent (UC)*, *uncertain-inconsistent (UI)* and

Algorithm 1: The training procedure of four-component Gaussian Mixture Model (GMM) using semi-supervised Expectation-Maximization

Input: $\mathcal{D}_L$; $\mathcal{D}_U$; α

Output: Optimal GMM parameters $\mathcal{W}$

Initialize $\mathcal{W}=\{(\pi_k,\mu_k,\sigma_k)|1\leq k\leq K\}$ using $\mathcal{D}_L$ via Bayes estimation.

while $\mathcal{W}$ *is not converged* **do**

$\{(\ell_g^i,p_g^i)\}_{i=1}^{|\mathcal{D}_L|}\leftarrow\mathcal{D}_L$

$\{(\ell_u^j,)\}_{j=1}^{|\mathcal{D}_U|}\leftarrow\mathcal{D}_U$

// E step:

$\gamma_{gk}(\ell_g^i)=\mathbb{1}\{p_g^i=k\}$,

$\gamma_{uk}(\ell_u^j)=\frac{\pi_k\mathcal{N}(\ell_u^j;\mu_k,\sigma_k)}{\sum_{k'=1}^{K}\pi_{k'}\mathcal{N}(\ell_u^j;\mu_{k'},\sigma_{k'})}$,

// M step:

$\pi_k=\frac{\alpha\sum_i\gamma_{gk}(\ell_g^i)+(1-\alpha)\sum_j\gamma_{uk}(\ell_u^j)}{\alpha|\mathcal{D}_L|+(1-\alpha)|\mathcal{D}_U|}$;

$\mu_k=\frac{\alpha\sum_i\gamma_{gk}(\ell_g^i)\cdot\ell_g^i+(1-\alpha)\sum_j\gamma_{uk}^j\cdot\ell_u^j}{\alpha\sum_i\gamma_{gk}(\ell_g^i)+(1-\alpha)\sum_j\gamma_{uk}(\ell_u^j)}$;

$\sigma_k=\frac{\alpha\sum_i\gamma_{gk}(\ell_g^i)\cdot(\ell_g^i-\mu_k)^2+(1-\alpha)\sum_j\gamma_{uk}\cdot(\ell_u^j)(\ell_u^j-\mu_k)^2}{\alpha\sum_i\gamma_{gk}(\ell_g^i)+(1-\alpha)\sum_j\gamma_{uk}(\ell_u^j)}$;

confident-inconsistent (CI), respectively. In this way, samples that are both uncertain to the model and contain target-specific knowledge can be extracted based on the InfoF score. In next section, we devise a strategy for informative sampling to adapatively distinguish those subsets from unlabeled target samples.

3.2. Informative Sampling Function

During training, the varying scale of InfoF scores obtained above would hinder accurate data categorization using manual data division, leading to a biased sampling strategy and degraded performance. Therefore, our work here formulates an Informative Sampling Function to achieve the goal of adaptively selecting informative target samples. In general, it can be observed that the four categories of unlabeled target data as stated above are significantly separable in the distributions of the InfoF scores, making it possible to learn mixture models to separate four sample partitioning from unlabeled target data. Thereafter, by learning the distribution of InfoF scores using a four-component Gaussian mixture model (GMM), the target data can be adaptively mapped to the probability distribution of four categories and then be separated, thereby avoiding sampling bias that may be created by manual division.

In specific, we first take inspiration from [34] that labeled samples can be used to construct supervision information, therefore building a more trustworthy GMM model with K Gaussian components (namely $K=4$ in this work) via a semi-supervised Expectation-Maximization algorithm. To this end, we then calculate the InfoF score for a labeled sample with (x,y) from $\mathcal{S}$ or $\mathcal{T}$ as follows,

$$\ell_g(x,y)=-\sum_{c=1}^{C}\mathbb{1}\{c=y\}\cdot\log P_c(x). \tag{4}$$

The similarity-based label is determined using categorical centroids of the labeled data, resulting in the totally identical similarity-based label and model prediction for labeled samples. In order to obtain reliable supervision and build a better GMM model, we have replaced the similarity-based label with the ground-truth class label here.

Afterwards, to obtain the supervision information for training GMM, we here define a piecewise function to obtain the observation labels that assign all labeled samples across domains to the corresponding Gaussian components. Thus, the observation label of a candidate instance x can be computed as,

$$q(x,y)=\begin{cases}1, & \max P(x)\geq\tau \text{ and } y=\arg\max P(x),\\ 2, & \max P(x)<\tau \text{ and } y=\arg\max P(x),\\ 3, & \max P(x)\geq\tau \text{ and } y\neq\arg\max P(x),\\ 4, & \max P(x)<\tau \text{ and } y\neq\arg\max P(x),\end{cases} \tag{5}$$

where the numbers, i.e., $k=1,\cdots,4$, indicate the indices of different Gaussian components corresponding to the CC, UC, UI and CI sample subsets, respectively. As well, τ is a confidence threshold to classify labeled samples into the confident subset and the uncertain subset. Furthermore, the comparison between the ground-truth and predicted class labels is used to divide them into the consistent subset and the inconsistent subset of labeled training samples.

Once we obtain $\{\ell_g(x,y)|(x,y)\in\mathcal{S}\cup\mathcal{T}\}$, $\{\ell_u(x)|x\in\mathcal{U}\}$, and $q(x,y)$, we incorporate them into building training instances for learning the GMM model as follows,

$$\mathcal{D}_L\leftarrow\{(\ell_g(x,y),q(x,y))|(x,y)\in\mathcal{S}\cup\mathcal{T}\}, \tag{6}$$

$$\mathcal{D}_U\leftarrow\{(\ell_u(x),)|x\in\mathcal{U}\}. \tag{7}$$

For convenience, we denote the GMM model by $\mathcal{W}=\{(\pi_k,\mu_k,\sigma_k)|1\leq k\leq K\}$. Afterwards, given a training instance ℓ_x from $\mathcal{D}_L$ or $\mathcal{D}_U$, the output of the probability density functions of the GMM model regarding ℓ_x can be computed as

$$p(\ell_x)=\sum_{k=1}^{K}\pi_k\mathcal{N}(\ell_x;\mu_k,\sigma_k), \tag{8}$$

where π_k represents the weight of the k-th Gaussian component subject to $\sum_{k=1}^{K}\pi_k=1$ and $0\leq\pi_k\leq 1$ for $k=1,\cdots,K$, while μ_k and σ_k are vectors denoting the mean and variance of such a component. Here, $\mathcal{N}(\cdot;\mu_k,\sigma_k)$ is used to model the Gaussian distribution of the k-th component. Note that a given coefficient $\alpha=|\mathcal{D}_L|/|\mathcal{D}_U|$ is

used for the weighted integration of the labeled and unlabeled training instances. Thus, we can summarize the training procedure of the GMM model using a semi-supervised EM algorithm in Algorithm 1 according to [34].

After obtaining the trained GMM parameters $\mathcal{W}$, the probability of a sample x from $\mathcal{U}$ being classified into the k-th component can be computed as follows,

$$\Pr(z=k|x,\mathcal{W}) = \frac{\pi_k \mathcal{N}(\ell_u(x);\mu_k,\sigma_k)}{\sum_{k'=1}^{K}\pi_{k'}\mathcal{N}(\ell_u(x);\mu_{k'},\sigma_{k'})}, \quad (9)$$

where z is a discrete variable to infer which component the sample x belongs to.

3.3. Candidate Selection and Training Objectives

After acquiring the informativeness sampling function, it is utilized for carrying out data partitioning. For ADA, the main focus should be on annotating samples that are uncertain to the model and can effectively represent the target dataset, which is termed as *uncertain-inconsistent* samples within our subdivision framework. Therefore, during each sampling step, we first rank the unlabeled target samples in descending order according to the scores predicted by $\Pr(z=3|\cdot,\mathcal{W})$, and then the top b examples are selected to annotate and moved into labeled target data. Following previous ADA works, we apply the standard cross-entropy loss to the labeled data as follows:

$$\mathcal{L}_{sup} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{S}\cup\mathcal{T}} \big[-\sum_{c=1}^{C}\mathbb{1}\{c=y\}\cdot\log P_c(x)\big]. \quad (10)$$

To boost the selectivity of the sampling strategy, we further design tailored training techniques for the other data subsets with varied transferable properties. We perform data partitioning on the remaining unlabeled target samples, i.e., $\mathcal{U} \leftarrow \mathcal{U}\backslash\mathcal{T}$, according to the posterior probabilities regarding the rest of Gaussian components.

$$\hat{\mathcal{U}}_k = \{x|\arg\max \mathbf{Pr}(x)=k, \forall(x,)\in\mathcal{U}\}, \text{for } k=1,2,4, \quad (11)$$

where $\hat{\mathcal{U}}_1$, $\hat{\mathcal{U}}_2$ and $\hat{\mathcal{U}}_4$ refer to $\hat{\mathcal{U}}_{\text{CC}}$, $\hat{\mathcal{U}}_{\text{UC}}$, $\hat{\mathcal{U}}_{\text{CI}}$. Here, $\mathbf{Pr}(x)$ is the concatenated probability vector corresponding to different components, i.e., $\mathbf{Pr}(x) = [\Pr(z=1|x,\mathcal{W}), \Pr(z=2|x,\mathcal{W}), \Pr(z=3|x,\mathcal{W}), \Pr(z=4|x,\mathcal{W})]$ w.r.t the sample x.

As shown in Figure 1(4), the source-oriented model could produce confident but unreliable predictions for some target samples having large domain gap with the source. Samples from $\hat{\mathcal{U}}_{\text{CI}}$ have extremely controversial results between model prediction and similarity-based label, which are most likely to be some excessively challenging instances. Therefore, we withhold them in the current stage to avoid jeopardizing the training stability.

The samples in $\hat{\mathcal{U}}_{\text{CC}}$ have confident and reliable predictions, which tend to be some class centroid in the target feature space. Therefore, we apply consistency regularization to them to enforce the prediction consistency under different perturbations, which can be formulated as follows:

$$\mathcal{L}_{con} = \mathop{\mathbb{E}}_{(x,)\sim\hat{\mathcal{U}}_{\text{CC}}} \big[-\sum_{c=1}^{C}\mathbb{1}\{c=\ddot{y}(x)\}\cdot\log P_c(\mathbf{Aug}(x))\big], \quad (12)$$

where $\mathbf{Aug}(\cdot)$ is a function to create perturbations for the sample x using classic data augmentation techniques such as AutoAugment [2].

For data group $\hat{\mathcal{U}}_{\text{UC}}$ whose prediction tend to be reliable but uncertain, we directly minimize the entropy of the predictions to using the following conditional entropy loss:

$$\mathcal{L}_{ent} = \mathop{\mathbb{E}}_{(x,)\sim\hat{\mathcal{U}}_{UC}} \big[-\sum_{c=1}^{C} P_c(x)\log P_c(x)\big]. \quad (13)$$

In summary, the overall loss functions used to further optimize the model can be formulated as follows,

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda_c\mathcal{L}_{con} + \lambda_e\mathcal{L}_{ent} \quad (14)$$

where λ_c and λ_e are the weights to trade off different loss terms during the training process. As shown in Figure 2, $\mathcal{L}_{sup}$ can calibrate the potentially underfitting of uncertain and target-specific samples. $\mathcal{L}_{con}$ can guide the model to learn global cross-domain clustering while $\mathcal{L}_{ent}$ serves to minimize the distance of data from the same class. As demonstrated in Sec. 4.3, these training objectives are complementary to each other to best adapt the model to the target domain.

3.4. Compatibility with UDA/SSDA/SFDA

The proposed sampling strategy of DiaNA requires no additional network modules like a domain discriminator or multiple classifiers [4, 26], making it easy to be integrated into existing DA frameworks, such as unsupervised domain adaptation (UDA), semi-supervised domain adaptation (SSDA), and source-free domain adaptation (SFDA); see supplementary document for the implementation details. It should be noted that we further design a variant of DiaNA for the SFDA settings, due to the unavailability of source data. As evidenced by Sec. 4.3, when integrated into diverse DA techniques, the proposed sampling strategy can deliver more performance improvement than random sample selection, thereby demonstrating its compatibility in various DA settings. To the best of our knowledge, we are the first to propose an active selection strategy that can generalize to UDA, SSDA, and SFDA methods.

Method	R → C			C → S			S → P			C → Q			R → S			R → P			AVG		
	1k	2k	5k	1k	2k	5k	1k	2k	5k	1k	2k	5k	1k	2k	5k	1k	2k	5k	1k	2k	5k
Random	50.9	55.2	61.5	42.6	44.8	49.4	41.0	44.4	50.1	26.3	33.2	43.0	38.2	40.7	46.9	46.5	48.8	53.7	40.9	44.5	50.8
CONF	48.6	54.2	61.3	41.9	43.6	48.0	41.6	44.2	48.5	25.5	32.7	42.8	36.9	39.4	45.3	47.1	47.8	51.2	40.3	43.6	49.5
Entropy [29]	48.0	54.2	61.9	42.7	45.2	48.9	41.1	43.7	48.7	24.6	31.0	42.1	37.0	40.4	45.7	46.5	48.4	52.2	40.0	43.8	49.9
Coreset [25]	49.1	52.5	59.8	40.6	42.2	45.8	39.1	41.5	44.8	24.7	30.3	39.7	36.7	38.3	43.3	45.1	46.8	50.3	39.2	41.9	47.3
BADGE [1]	51.7	55.4	62.5	43.6	45.7	50.1	42.3	45.5	49.7	26.0	34.3	43.9	37.8	41.5	47.8	48.0	50.1	54.2	41.6	45.4	51.4
Alpha [20]	52.0	56.3	62.8	44.3	46.8	50.5	41.9	44.9	49.7	27.2	34.6	44.4	38.6	41.3	47.9	47.6	50.4	53.2	41.9	45.7	51.4
AADA [26]	50.8	56.3	64.7	42.8	45.7	51.0	41.8	45.6	50.8	25.2	31.3	40.8	38.5	41.7	49.5	46.5	49.7	55.1	40.9	45.1	52.0
SDM [33]	51.0	55.4	62.3	43.1	45.2	49.6	41.9	44.9	49.3	26.9	35.0	43.8	37.3	40.2	46.5	47.4	49.7	53.2	41.3	45.1	50.8
EADA [31]	51.5	56.1	63.9	44.2	46.3	51.7	43.8	46.1	50.3	27.5	34.8	42.9	39.0	41.5	46.4	48.6	50.4	54.1	42.4	45.9	51.2
CLUE [22]	54.1	59.4	66.0	45.3	48.7	53.7	44.0	48.0	53.2	28.8	35.5	43.2	40.4	44.4	51.1	50.0	52.7	56.4	43.8	48.1	53.9
DiaNA(Ours)	**55.6**	**60.4**	**68.7**	**47.0**	**50.8**	**56.6**	**44.2**	**49.0**	**55.4**	**30.2**	**39.5**	**52.1**	**42.4**	**47.9**	**55.0**	**50.5**	**53.6**	**59.1**	**45.0**	**50.2**	**57.8**

Table 1. Comparison results (Accuracy: %) on DomainNet with 1k, 2k and 5k labeling budgets. "Random" and "CONF" correspond to the classic AL approaches "Random Sampling" and "Least-Confidence Sampling".

Method	Office-Home												
	A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	AVG
Random	52.5	74.3	77.4	56.3	69.7	68.9	57.7	50.9	75.8	70.0	54.6	81.3	65.8
Entropy [29]	51.3	72.7	76.4	61.7	74.1	72.9	57.6	51.1	76.6	69.8	57.1	82.3	67.0
Coreset [25]	52.8	73.3	75.5	59.8	73.4	70.8	58.1	52.6	75.8	69.3	56.5	82.5	66.7
BADGE [1]	57.2	75.2	76.9	61.5	77.2	71.9	60.4	53.6	78.0	70.8	61.1	84.3	69.0
Alpha [20]	57.0	79.4	78.2	61.6	78.0	74.1	58.9	54.2	78.1	71.7	61.1	84.6	69.7
AADA [26]	56.6	78.1	79.0	58.5	73.7	71.0	60.1	53.1	77.0	70.6	57.0	84.5	68.3
CLUE [22]	58.0	79.1	77.0	61.3	78.0	73.1	60.4	55.9	77.9	70.9	60.3	84.1	69.7
TQS [4]	58.6	81.1	81.5	61.1	76.1	73.3	61.2	54.7	79.7	73.4	58.9	86.1	70.5
SDM [33]	61.2	82.2	82.7	66.1	77.9	76.1	66.1	58.4	81.0	76.0	62.5	87.0	73.1
EADA [31]	63.6	84.4	83.5	70.7	83.7	80.5	73.0	63.5	85.2	78.4	65.4	88.6	76.7
DiaNA(Ours)	**64.5**	**86.0**	**84.9**	**72.3**	**84.6**	**82.5**	**73.3**	**63.7**	**85.6**	**78.5**	**67.2**	**89.5**	**77.7**

Table 2. Comparison results (Accuracy: %) on Office-Home with 5% labeling budget.

4. Experiments

4.1. Dataset

We validate the effectiveness of the proposed approach DiaNA on three standard benchmark datasets: DomainNet [21], Office-Home [28] and CIFAR-10 [11]. On DomainNet, we select five domains namley Real (**R**), Clipart (**C**), Sketch (**S**), Painting (**P**), and Qucikdraw (**Q**), while we hire these domains to construct 6 adaptation scenarios. As in [4, 31, 33], we report 12 different adaptation scenarios on Office-Home, constructed from four domains: Real (**R**), Clipart (**C**), Art (**A**), and Product (**P**). To further evaluate the generalization of the proposed sample selection strategy, we additionally report the classification results on CIFAR-10 with a standard active learning setting.

4.2. Implementation

Here, we mainly focus on the detailed descriptions of the settings involving active learning on DomainNet, Office-Home, and CIFAR-10, while more implementation details can be found in the supplementary material. In each adaptation scenario, we report the average accuracy over 3 trials.

DomainNet. Similar to CLUE [22], we consider ResNet-34 [8] as the network backbone, where such network model is pre-trained on all labeled source samples over 50 training epochs through standard supervision. In all adaptation cases, we set $B = 5000$ and $R = 10$.

Office-Home. To be fair, we follow the experimental settings of existing ADA works [4, 31, 33] to conduct performance comparison. In particular, we also use ResNet-50 [8] as the backbone model, while we choose a 5% proportion of unlabeled target data to set the labeling budget B, and R is set to 5.

CIFAR-10. On this dataset, we select ResNet-34 [8] as the backbone model. We randomly choose select 10% training samples at random as the initially labeled data set, and we then annotate 4% at each sampling step with a total annotation budget 30%.

4.3. Main Results

Results on DomainNet and Office-Home. We summarize the comparison with baselines in Table 1 and Table 2 on both datasets respectively. Note that we consider varying label budgets, namely i.e., 1k, 2k, and 5k, on DomainNet. We can see that DiaNA achieves the highest average accuracy in all cases on both datasets for solving the ADA tasks. Especially in the most difficult adaptation scenarios, e.g., C→A and R→C, on Office-Home, the proposed method can still exceed all the other methods by more than 1.6% and 1.8% respectively, demonstrating the effectiveness of DiaNA on the commonly used DA benchmark datasets.

Results on CIFAR-10. We also extend the proposed method to standard AL tasks on CIFAR-10, where the domain gap between labeled and unlabeled ones is minor. Figure 4(a) lists the comparisons of DiaNA against classic AL strategies involving [1, 20, 25, 29]. It can be observed that compared to classic AL algorithms, with fewer labeling budgets, the effect of the proposed method is still notice-

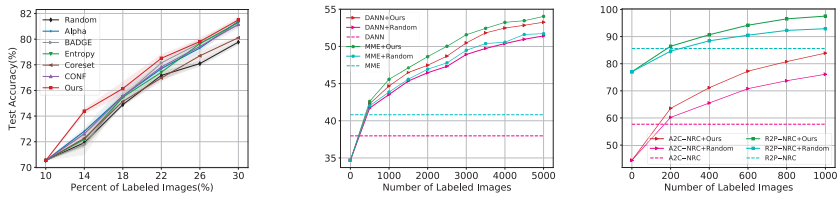


Figure 4. (a) Comparison results on CIFAR-10. (b) Combination with UDA/SSDA methods. (c) Combination with SFDA methods.

Method	AL	$\mathcal{L}_{con}$	$\mathcal{L}_{ent}$	GMM	AVG
DiaNA	UI	✓	✓	✓	77.7±0.2
DiaNA w/o AL	R	✓	✓	✓	74.8±0.4
DiaNA w/o $\mathcal{L}_{con}$	UI		✓	✓	76.9±0.3
DiaNA w/o $\mathcal{L}_{ent}$	UI	✓		✓	75.8±0.3
DiaNA w/o GMM	UI	✓	✓		74.0±0.2
DiaNA-UI	UI			✓	72.1±0.4
DiaNA-UC	UC			✓	68.3±0.6
DiaNA-CI	CI			✓	66.4±0.8
DiaNA-CC	CC			✓	63.0±0.3
Random (Baseline)	R				65.8±0.5

Table 3. Ablation study results of DiaNA. The "AL" column indicates which type of data subset should be selected to be annotated during each sampling step. We evaluated the performance of all adaptation scenarios on Office-Home by averaging their accuracy.

able, resulting from its significant performance improvement. This also indicates that our sampling strategy can be effectively applied to general active learning.

Combination with UDA/SSDA/SFDA methods. To validate the compatibility of the proposed DiaNA with existing UDA, SSDA, and SFDA algorithms, we conduct the experiments for UDA and SSDA in the adaptation C→S on DomainNet, and for SFDA in C→A and R→C on Office-Home. DANN [5], MME [24], and NRC [36] correspond to the SOTA methods towards UDA, SSDA, and SFDA, respectively. As shown in Figure 4(b) and Figure 4(c), combining the selection strategy of DiaNA with other DA methods, i.e., DANN+Ours, MME+Ours, and NRC+Ours, significantly increases the performance of existing DA algorithms that select labeled target samples via random selection, namely DANN+Random, MME+Random and NRC+Random. This suggests that, adaptively considering both the domainness and uncertainty, DiaNA is beneficial for the selection of informative target sample searching, while combined into diverse domain adaptation techniques.

4.4. Ablation study

Efficacy of the targeted training strategies. To illustrate the effect of each component of the proposed DiaNA, we conduct ablation study by removing the corresponding individual components. As displayed in Table 3, the proposed method exceeds all of the model variants by a large accuracy margin on average, demonstrating the effectiveness of each component. By comparing our proposed entire approach with DiaNA w/o GMM, manual data division strategy might considerably cause performance degradation, thereby in turn emphasizing the need for adaptive data subdivision protocol. Furthermore, as demonstrated, disrupting the correspondence between targeted data subsets provides empirical evidence supporting the validity of our proposed customized learning techniques; see supplementary document for more details.

Efficacy of the sampling strategy. To provide insights into how effectively the domainness-based metric and the uncertainty-based metric work in the sampling step, we use different data subsets to replace the UI subset as active samples for annotation. To eliminate the influence of $\mathcal{L}_{con}$ and $\mathcal{L}_{ent}$, we remove these two terms but only retain $\mathcal{L}_{sup}$ during training. As shown in Table 3, both DiaNA-UC and DiaNA-CI achieve better classification performance than baseline selection, thereby showing the effect of the proposed uncertainty metric and domainness metric. Moreover, active sampling on UI samples (DiaNA-UI) obtains the best performance among all the variants, further illustrating the superiority of the integration of both metrics.

5. Conclusions

In this paper, we introduce a "divide-and-adapt" framework, named DiaNA, to address the problem of active domain adaptation. Specifically, we first design an informativeness function that jointly captures sample's uncertainty and domainness. Moreover, we devise an automatic data subdivision protocol to partition the target instances into four categories with different characteristics. While selecting the most informative samples to annotate, we also design tailored learning strategies for the other target data subsets with stratified transferable properties. Extensive experimental results, as well as ablation studies, have confirmed the superiority of the proposed approach.

Acknowledgments

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation (NO. 2020B1515020048), in part by the National Natural Science Foundation of China (NO. 61976250), in part by the Shenzhen Science and Technology Program (NO. JCYJ20220530141211024) and in part by the Fundamental Research Funds for the Central Universities under Grant 22lgqb25.

References

[1] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2019.

[2] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

[3] Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In *European conference on computer vision*, pages 562–577. Springer, 2014.

[4] Bo Fu, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. Transferable query selection for active domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7272–7281, 2021.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[6] Yuhong Guo. Active instance sampling via matrix partition. *Advances in Neural Information Processing Systems*, 23, 2010.

[7] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *International Conference on Learning Representations*, 2020.

[8] 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.

[9] Sheng-Jun Huang, Jia-Wei Zhao, and Zhao-Yang Liu. Cost-effective training of deep cnns with active model adaptation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1580–1588, 2018.

[10] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pages 2372–2379. IEEE, 2009.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.

[13] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2505–2514, 2021.

[14] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021.

[15] Zhuoming Liu, Hao Ding, Huaping Zhong, Weijia Li, Jifeng Dai, and Conghui He. Influence selection for active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9274–9283, 2021.

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[17] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[18] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[19] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79, 2004.

[20] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Gholamreza Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12237–12246, 2022.

[21] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

[22] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8505–8514, 2021.

[23] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning*, pages 413–424. Springer, 2006.

[24] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[25] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[26] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 739–748, 2020.

[27] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[28] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[29] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE, 2014.

[30] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[31] Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, Xinjing Cheng, and Guoren Wang. Active learning for domain adaptation: An energy-based approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8708–8716, 2022.

[32] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[33] Ming Xie, Yuxi Li, Yabiao Wang, Zekun Luo, Zhenye Gan, Zhongyi Sun, Mingmin Chi, Chengjie Wang, and Pei Wang. Learning distinctive margin toward active domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7993–8002, 2022.

[34] Heng-Chao Yan, Jun-Hong Zhou, and Chee Khiang Pang. Gaussian mixture model using semisupervised learning for probabilistic fault diagnosis under new data categories. *IEEE Transactions on Instrumentation and Measurement*, 66(4):723–733, 2017.

[35] Zizheng Yan, Yushuang Wu, Guanbin Li, Yipeng Qin, Xiaoguang Han, and Shuguang Cui. Multi-level consistency learning for semi-supervised domain adaptation. *arXiv preprint arXiv:2205.04066*, 2022.

[36] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in Neural Information Processing Systems*, 34:29393–29405, 2021.

[37] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019.

[38] Ziyi Zhang, Weikai Chen, Hui Cheng, Zhen Li, Siyuan Li, Liang Lin, and Guanbin Li. Divide and contrast: Source-free domain adaptation via adaptive contrastive learning. *arXiv preprint arXiv:2211.06612*, 2022.

Efficient Second-Order Plane Adjustment

Lipu Zhou
Meituan, Beijing, China
zhoulipu@meituan.com

Abstract

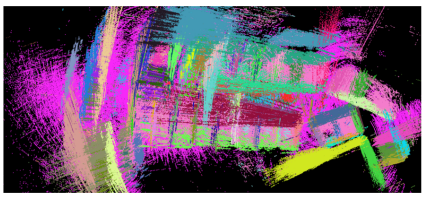
Planes are generally used in 3D reconstruction for depth sensors, such as RGB-D cameras and LiDARs. This paper focuses on the problem of estimating the optimal planes and sensor poses to minimize the point-to-plane distance. The resulting least-squares problem is referred to as plane adjustment (PA) in the literature, which is the counterpart of bundle adjustment (BA) in visual reconstruction. Iterative methods are adopted to solve these least-squares problems. Typically, Newton's method is rarely used for a large-scale least-squares problem, due to the high computational complexity of the Hessian matrix. Instead, methods using an approximation of the Hessian matrix, such as the Levenberg-Marquardt (LM) method, are generally adopted. This paper adopts the Newton's method to efficiently solve the PA problem. Specifically, given poses, the optimal plane have a close-form solution. Thus we can eliminate planes from the cost function, which significantly reduces the number of variables. Furthermore, as the optimal planes are functions of poses, this method actually ensures that the optimal planes for the current estimated poses can be obtained at each iteration, which benefits the convergence. The difficulty lies in how to efficiently compute the Hessian matrix and the gradient of the resulting cost. This paper provides an efficient solution. Empirical evaluation shows that our algorithm outperforms the state-of-the-art algorithms.

1. Introduction

Planes ubiquitously exist in man-made environments, as demonstrated in Fig. 1. Thus they are generally used in simultaneous localization and mapping (SLAM) systems for depth sensors, such as RGB-D cameras [6, 9, 12–14] and LiDARs [16, 21, 22, 24, 26]. Just as bundle adjustment (BA) [3, 8, 20, 25] is important for visual reconstruction [1, 5, 18, 19], jointly optimizing planes and depth sensor poses, which is called plane adjustment (PA) [24, 26], is critical for 3D reconstruction using depth sensors. This paper focuses on efficiently solving the large-scale PA problem.

The BA and PA problems both involve jointly optimizing 3D structures and sensor poses. As the two problems are similar, it is straightforward to apply the well-studied solutions for BA to PA, as done in [23, 26]. However, planes in PA can be eliminated, so that the cost function of the PA problem only depends on sensor poses, which significantly reduces the number of variables. This property provides a promising direction to efficiently solve the PA problem. However, it is difficult to compute the Hessian matrix and the gradient vector of the resulting cost. Although this problem was studied in several previous works [10, 16], no efficient solution has been proposed. This paper seeks to solve this problem.

The main contribution of this paper is an efficient PA solution using Newton's method. We derive a closed-form solution for the Hessian matrix and the gradient vector for the PA problem whose computational complexity is inde-



(a) Point cloud from perturbed poses



(b) Point cloud from our algorithm

Figure 1. We use Gaussian noises to perturb the poses of dataset D in Fig. 3. The standard deviations for rotation and translation are 3° and $0.3m$, respectively. The resulting point cloud (a) is in a mess. Fig. (b) shows the result from our algorithm. Our algorithm can quickly align the planes, as shown in Fig. 5.

pendent of the number of points on the planes. Our experimental results show that the proposed algorithm converges faster than the state-of-the-art algorithms.

2. Related Work

The PA problem is closely related to the BA problem. In BA, points and camera poses are jointly optimized to minimize the reprojection error. Schur complement [3, 20, 25] or the square root method [7, 8] is generally used to solve the linear system of the iterative methods. The keypoint is to generate a reduced camera system (RCS) which only relates to camera poses.

In PA, planes and poses are jointly optimized. Planes are the counterparts of points in BA. Thus, the well-known solutions for the BA problem can be applied to the PA problem [23, 24]. In the literature, two cost functions are used to formulate the PA problem. The first one is the plane-to-plane distance which measures the difference between two plane parameters [12, 13]. The value of the plane-to-plane distance is related to the choice of the global coordinate system, which means the selection of the global coordinate system will affect the accuracy of the results. The second one is the point-to-plane distance, whose value is invariant to the choice of the global coordinate system. The solutions of different choices of the global coordinate system are equivalent up to a rigid transformation. Zhou *et al.* [23] show that the point-to-plane distance can converge faster and lead to a more accurate result. But unlike BA where each 3D point has only one 2D observation at a pose, a plane can generate many points at a pose as demonstrated in Fig. 2. This means the point-to-plane distance probably leads to a very large-scale least-squares problem. Directly adopting the BA solutions is computationally infeasible for a large-scale PA problem. Zhou *et al.* [23] propose to use the QR decomposition to accelerate the computation.

For a general least-squares problem with M variables, the computational complexity of the Hessian matrix is $O(M^2)$. Thus, in the computer vision community, it is ingrained that Newton's method is infeasible for a large-scale optimization problem, as calculating the Hessian matrix is computationally demanding. Instead, Gauss-Newton-based iterative methods are generally adopted. Suppose that $\mathbf{J}$ is the Jacobian matrix of the residuals. The Gauss-Newton method actually approximates the Hessian matrix by $\mathbf{H} \approx \mathbf{J}^T\mathbf{J}$. In theory, Newton's method can lead to a better quadratic approximation to the original cost function, which means the Newton's step probably yields a better result. This in turn may reduce the number of iterations.

The PA problem has a special property that the optimal plane parameters are determined by the poses. That is to say the point-to-plane cost actually only depends on the poses. This property is attractive, as it significantly reduces the number of variables, which makes using the Newton's method possible. In traditional iterative methods, the correlation between the plane parameters and the poses are ignored. Thus, after one iteration, there is no guarantee that the new plane parameters are optimal for the new poses. Using the property of the PA, it is possible to overcome this drawback, which may lead to faster convergence. Several previous works seek to exploit this property of PA. Ferrer [10] explored an algebraic point-to-plane distance and provided a closed-form gradient for the resulting cost. The algebraic cost may result in a suboptimal solution [4], and the first-order optimization generally leads to slow convergence [20]. Liu *et al.* [16] provided analytic forms of the Hessian matrix and the gradient of the genuine point-to-plane cost. Assume that N points are captured from a plane. The computational complexity of the Hessian matrix related to these points is $O(N^2)$. Since N can be large as shown in Fig. 2, this algorithm is computationally demanding and infeasible for a large-scale problem.

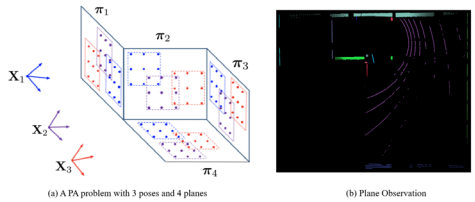


Figure 2. A schematic of the PA problem and the planes detected in a LiDAR scan. Unlike BA where each 3D point only has one observation, many points can be captured from a plane in PA. Assume that N points are captured from a plane. The computational complexity of the Hessian matrix related to these points using BALM [16] is $O(N^2)$. Thus, this method is infeasible for a large-scale problem. In contrast, the computational complexity of our algorithm is independent of N.

In summary, the potential benefits of the special property of the PA problem have not been manifested in previous works. The bottleneck is how to efficiently compute the Hessian matrix and the gradient vector. This paper focuses on solving this problem.

3. Problem Formulation

In this paper we use italic, boldfaced lowercase and boldfaced uppercase letters to represent scalars, vectors and matrices, respectively.

3.1. Notations

Planes and Poses A plane can be represented by a four-dimensional vector $\boldsymbol{\pi} = [\boldsymbol{n}; d]$. We denote the rotational and translational components from a depth sensor coordinate system to the global coordinate system as $\mathbf{R} \in SO(3)$

and $\boldsymbol{t} \in \mathbb{R}^3$, respectively. To simplify the notation in the following description, we also use the following two matrices to represent a pose:

$$\mathbf{X} = \begin{bmatrix} \mathbf{R}, & \boldsymbol{t} \\ \mathbf{0}, & 1 \end{bmatrix} \in SE(3) \text{ and } \mathbf{T} = \begin{bmatrix} \mathbf{R}, & \boldsymbol{t} \end{bmatrix}. \quad (1)$$

As $\mathbf{R} \in SO(3)$, a certain parameterization is usually adopted in the optimization [20]. In this paper, we use the Cayley-Gibbs-Rodriguez (CGR) parameterization [11] to represent $\mathbf{R}$

$$\mathbf{R} = \frac{\bar{\mathbf{R}}}{1 + \boldsymbol{s}^T\boldsymbol{s}}, \bar{\mathbf{R}} = \left(1 - \boldsymbol{s}^T\boldsymbol{s}\right)\mathbf{I}_3 + 2[\boldsymbol{s}]_\times + 2\boldsymbol{s}\boldsymbol{s}^T, \quad (2)$$

where $\boldsymbol{s} = [s_1; s_2; s_3]$ is a three-dimensional vector. We adopt the CGR parameterization as it is a minimal representation for $\mathbf{R}$. Furthermore, unlike the angle-axis parameterization that is singular at $\mathbf{I}_3$, the CGR parameterization is well-defined at $\mathbf{I}_3$, and equals to $[0; 0; 0]$ which can accelerate the computation, as described in Section 4.3. We parameterize $\mathbf{X}$ as a six-dimensional vector $\boldsymbol{x} = [\boldsymbol{s}; \boldsymbol{t}]$.

Newton's method This paper adopts the damped Newton's method in the optimization. For a cost function $f(\boldsymbol{z})$, the damped Newton's method seeks to find its minimizer from an initial point. Assume that $\boldsymbol{z}_n$ is the solution at the nth iteration. Given the Hessian matrix $\mathbf{H}_f(\boldsymbol{z}_n)$ and the gradient $\boldsymbol{g}_f(\boldsymbol{z}_n)$ at $\boldsymbol{z}_n$, $\boldsymbol{z}_n$ is updated by $\boldsymbol{z}_{n+1} = \boldsymbol{z}_n + \Delta\boldsymbol{z}$. Here $\Delta\boldsymbol{z}$ is from

$$(\mathbf{H}_f(\boldsymbol{z}_n) + \mu\mathbf{I}))\Delta z = -\boldsymbol{g}_f(\boldsymbol{z}_n), \quad (3)$$

where μ is adjusted at each iteration to make the value of $f(\boldsymbol{z})$ reduce, as done in the Levenberg-Marquardt (LM) algorithm [17].

Matrix Calculus In the following derivation, we will use vector-by-vector, vector-by-scalar, scalar-by-vector derivatives. Here we provide their definitions.

Assume $\boldsymbol{a} = [a_1; \cdots; a_N] \in \mathbb{R}^N$ is a vector function of $\boldsymbol{b} = [b_1, \cdots, b_M] \in \mathbb{R}^M$. The first-order partial derivatives of vector-by-vector $\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{b}}$, vector-by-scalar $\frac{\partial \boldsymbol{a}}{\partial b_j}$, and scalar-by-vector $\frac{\partial a_i}{\partial \boldsymbol{b}}$ are defined as

$$\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{b}} = \begin{bmatrix} \frac{\partial a_1}{\partial b_1} & \cdots & \frac{\partial a_N}{\partial b_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{a}_1}{\partial b_M} & \cdots & \frac{\partial \boldsymbol{a}_N}{\partial \boldsymbol{b}_M} \end{bmatrix}, \frac{\partial \boldsymbol{a}}{\partial b_j} = \begin{bmatrix} \frac{\partial a_1}{\partial b_j} \\ \vdots \\ \frac{\partial a_N}{\partial b_j} \end{bmatrix}, \frac{\partial a_i}{\partial \boldsymbol{b}} = \begin{bmatrix} \frac{\partial a_i}{\partial b_1} \\ \vdots \\ \frac{\partial a_i}{\partial b_M} \end{bmatrix} \quad (4)$$

where $\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{b}}$ is an $M \times N$ matrix with $\frac{\partial a_j}{\partial b_i}$ as the ith row jth column element, $\frac{\partial \boldsymbol{a}}{\partial b_j}$ is an N-dimensional vector whose ith term is $\frac{\partial a_i}{\partial b_j}$, and $\frac{\partial a_i}{\partial \boldsymbol{b}}$ is an M-dimensional vector whose ith term is $\frac{\partial a_i}{\partial b_i}$.

3.2. Optimal Plane Estimation

Given a set of K points $\{\boldsymbol{p}_i\}$, the optimal plane $\hat{\boldsymbol{\pi}}$ can be estimated by minimizing the sum of squared point-to-plane distances

$$\hat{\boldsymbol{\pi}} = \arg\min_{\boldsymbol{\pi}} \sum_i^K \left(\boldsymbol{n}^T\boldsymbol{p}_i + d\right)^2, \ s.t. \ \|\boldsymbol{n}\|_2^2 = 1. \quad (5)$$

There is a closed-form solution for $\hat{\boldsymbol{\pi}}$. Let us define

$$\mathbf{M} = \sum_{i=1}^K \left(\boldsymbol{p}_i - \bar{\boldsymbol{p}}\right)\left(\boldsymbol{p}_i - \bar{\boldsymbol{p}}\right)^T = \mathbf{S} - K\bar{\boldsymbol{p}}\bar{\boldsymbol{p}}^T, \quad (6)$$

where $\mathbf{S} = \sum_{i=1}^K \boldsymbol{p}_i\boldsymbol{p}_i^T$ and $\bar{\boldsymbol{p}} = \frac{1}{K}\sum_i^K \boldsymbol{p}_i$. Assume that $\lambda_3(\mathbf{M})$ and $\boldsymbol{\xi}_3(\mathbf{M})$ are the smallest eigenvalue of $\mathbf{M}$ and the corresponding eigenvector, respectively. Using these notations, we can write the optimal plane $\hat{\boldsymbol{\pi}} = [\hat{\boldsymbol{n}}; \hat{d}]$ as

$$\hat{\boldsymbol{n}} = \boldsymbol{\xi}_3(\mathbf{M}), \ \hat{d} = -\hat{\boldsymbol{n}}^T\bar{\boldsymbol{p}}. \quad (7)$$

Furthermore, the cost of (5) at $\hat{\boldsymbol{\pi}}$ equals to $\lambda_3(\mathbf{M})$, *i.e.*,

$$\lambda_3(\mathbf{M}) = \sum_{i=1}^K \left(\hat{\boldsymbol{n}}\boldsymbol{p}_i + \hat{d}\right)^2 = \min_{\boldsymbol{\pi}} \sum_{i=1}^K \left(\boldsymbol{n}\boldsymbol{p}_i + d\right)^2. \quad (8)$$

The above property will be used to eliminate planes in PA.

3.3. Plane Adjustment

Assume that there are M planes and N poses. According to section 3.1, the ith plane can be represented by a four-dimensional vector $\boldsymbol{\pi}_i = [\boldsymbol{n}_i; d_i]$. The jth pose is denoted as $\boldsymbol{x}_j$. The observation of $\boldsymbol{\pi}_i$ at $\boldsymbol{x}_j$ is a set of N_{ij} points $\mathbb{Q}_{ij} = \{\boldsymbol{p}_{ijk} \in \mathbb{R}^3\}_{k=1}^{N_{ij}}$. For a 3D point $\boldsymbol{p}_{ijk}$, we use $\tilde{\boldsymbol{p}}_{ijk} = [\boldsymbol{p}_{ijk}; 1]$ to represent the homogeneous coordinates of $\boldsymbol{p}_{ijk}$. Then, the transformation from the local coordinate system at $\boldsymbol{x}_j$ to the global coordinate system can be represented as

$$\boldsymbol{p}_{ijk}^g = \mathbf{R}_j\boldsymbol{p}_{ijk} + \boldsymbol{t}_j = \mathbf{T}_j\tilde{\boldsymbol{p}}_{ijk}, \quad (9)$$

where $\mathbf{T}_j$ is defined in (1). Then the distance d_{ijk} from $\boldsymbol{p}_{ijk}$ to $\boldsymbol{\pi}_i$ has the form

$$\begin{aligned} d_{ijk}(\boldsymbol{\pi}_i, \mathbf{x}_j) &= \boldsymbol{n}_i^T\left(\mathbf{R}_j\boldsymbol{p}_{ijk} + \boldsymbol{t}_j\right) + d_i \\ &= \boldsymbol{n}_i^T\mathbf{T}_j\tilde{\boldsymbol{p}}_{ijk} + d_j = \boldsymbol{\pi}_i^T\tilde{\boldsymbol{p}}_{ijk}^g. \end{aligned} \quad (10)$$

The PA problem is to jointly adjust the M planes $\{\boldsymbol{\pi}_i\}$ and the N sensor poses $\{\boldsymbol{x}_j\}$ to minimize the sum of squared point-to-plane distances. Specifically, using (10), we can formulate the cost function of the PA problem as

$$\min_{\{\boldsymbol{\pi}_i\},\{\mathbf{x}_j\}} \sum_{i=1}^M \underbrace{\sum_{j \in obs(\boldsymbol{\pi}_i)} \sum_{k=1}^{N_{ij}} d_{ijk}^2(\boldsymbol{\pi}_i, \mathbf{x}_j)}_{C_i(\boldsymbol{\pi}_i, \mathbb{X}_i),\ \mathbb{X}_i = \{\boldsymbol{x}_j | j \in obs(\boldsymbol{\pi}_i)\}} = \min_{\{\boldsymbol{\pi}_i\},\{\boldsymbol{x}_j\}} \sum_{i=1}^M C_i\left(\boldsymbol{\pi}_i, \mathbb{X}_i\right). \quad (11)$$

where $obs(\boldsymbol{\pi}_i)$ represents the indexes of poses where $\boldsymbol{\pi}_i$ can be observed, and $C_i(\boldsymbol{\pi}_i, \mathbb{X}_i)$ accumulates the errors from $N_i = \sum_{j \in obs(\boldsymbol{\pi}_i)} N_{ij}$ points captured at the poses in $\mathbb{X}_i$. According to (6) and (9), we get

$$\mathbf{M}_i(\mathbb{X}_i) = \sum_{j \in obs(\boldsymbol{\pi}_i)} \mathbf{S}_{ij} - N_i \bar{\boldsymbol{p}}_i \bar{\boldsymbol{p}}_i^T, \tag{12}$$

where $\bar{\boldsymbol{p}}_i = \frac{1}{N_i} \sum_{j \in obs(\boldsymbol{\pi}_i)} \sum_{k=1}^{N_{ij}} \boldsymbol{p}_{ijk}^g$ and $\mathbf{S}_{ij} = \sum_{k=1}^{N_{ij}} \boldsymbol{p}_{ijk}^g \left(\boldsymbol{p}_{ijk}^g\right)^T$. Here the elements in $\mathbf{M}_i$, $\mathbf{S}_{ij}$ and $\bar{\boldsymbol{p}}_i$ in (12) are all functions of the poses in $\mathbb{X}_i$. Substituting $\boldsymbol{p}_{ijk}^g$ in (9) into $\mathbf{S}_{ij}$ and $\bar{\boldsymbol{p}}_i$ in (12), we have

$$\begin{aligned} \mathbf{S}_{ij} &= \mathbf{T}_j \underbrace{\sum_{k=1}^{N_{ij}} \tilde{\boldsymbol{p}}_{ijk} \tilde{\boldsymbol{p}}_{ijk}^T}_{\mathbf{U}_{ij}} \mathbf{T}_j^T = \mathbf{T}_j \mathbf{U}_{ij} \mathbf{T}_j^T, \\ \bar{\boldsymbol{p}}_i &= \frac{1}{N_i} \sum_{j \in obs(\boldsymbol{\pi}_i)} \mathbf{T}_j \underbrace{\sum_{k=1}^{N_{ij}} \tilde{\boldsymbol{p}}_{ijk}}_{\tilde{\boldsymbol{p}}_{ij}} = \frac{1}{N_i} \sum_{j \in obs(\boldsymbol{\pi}_i)} \mathbf{T}_j \tilde{\boldsymbol{p}}_{ij}. \end{aligned} \tag{13}$$

Here $\mathbf{U}_{ij}$ and $\tilde{\boldsymbol{p}}_{ij}$ in (13) are constants. We only need to compute them once, and reuse them in the iteration.

According to (7), given poses in $\mathbb{X}_i$, the optimal solution for $\boldsymbol{\pi}_i$ has a closed-form expression $\hat{\boldsymbol{\pi}}_i = [\hat{\boldsymbol{n}}_i; \hat{d}_i]$, where $\hat{\boldsymbol{n}}_i = \boldsymbol{\xi}_3(\mathbf{M}_i(\mathbb{X}_i))$ and $\hat{d}_i = -\hat{\boldsymbol{n}} \bar{\boldsymbol{p}}_i$. As $\mathbf{M}_i$ and $\bar{\boldsymbol{p}}_i$ are functions of the poses in $\mathbb{X}_i$, $\hat{\boldsymbol{\pi}}_i$ is also a function of the poses in $\mathbb{X}_i$. That is to say $\hat{\boldsymbol{\pi}}_i$ is completely determined by the poses in $\mathbb{X}_i$. To simplify the notation, let us define

$$\lambda_{i,3}(\mathbb{X}_i) = \lambda_3(\mathbf{M}_i(\mathbb{X}_i)), \tag{14}$$

which represents the smallest eigenvalue of $\mathbf{M}_i(\mathbb{X}_i)$.

Substituting the optimal plane estimation $\hat{\boldsymbol{\pi}}_i$ into $C_i(\boldsymbol{\pi}_i, \mathbb{X}_i)$ in (11) and using (8), we have

$$\lambda_{i,3}(\mathbb{X}_i) = C_i\left(\hat{\boldsymbol{\pi}}_i, \mathbb{X}_i\right). \tag{15}$$

Using (15), we can formulate the PA problem in (11) as

$$\{\hat{\mathbf{x}}_j\} = \arg\min_{\{\mathbf{x}_j\}} \boldsymbol{\tau}, \ \boldsymbol{\tau} = \sum_{i=1}^{M} \lambda_{i,3}(\mathbb{X}_i). \tag{16}$$

Table 1 summarizes the notations for PA.

The cost function (16) only depends on poses, which significantly reduces the number of variables. However, as it is the sum of squared point-to-plane distances, we cannot apply the Gauss-Newton-based methods to minimize it, where the Jacobian matrix of residuals are required. Here we adopt the Newton's method to solve it. The crux for applying the Newton's method to minimize (16) is how to compute the gradient and the Hessian matrix of (16) efficiently. In the following sections, we provide a closed-form solution for them. Note that we can assign a weight to each point to punish outliers, which also has a close-form solution for planes using the **weighted PCA**. Our algorithm can be extended to this situation. To simplify the notation, we omit the variables of functions in the following description (*e.g.*, $\lambda_{i,3}(\mathbb{X}_i) \to \lambda_{i,3}$).

Notation	Description
$\boldsymbol{\pi}_i$	The ith plane.
$obs(\boldsymbol{\pi}_i)$	The set of indexes of poses which can see $\boldsymbol{\pi}_i$.
$\mathbb{X}_i$	The set of poses which can see $\boldsymbol{\pi}_i$.
$\mathbf{M}_i$	The scatter matrix for $\boldsymbol{\pi}_i$.
$\lambda_{3,i}$	The smallest eigenvalue of $\mathbf{M}_i$.
$\boldsymbol{x}_j, \boldsymbol{x}_k$	The jth and kth poses.
x_{jm}, x_{kn}	The mth and nth elements of $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$.
$\mathbb{P}_j$	The set of planes that are visible to $\boldsymbol{x}_j$.
$\mathbb{P}_{jk}$	The set of planes that are visible to $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$.

Table 1. Table of notations.

4. Newton's Iteration for Plane Adjustment

Let us denote the gradient and the Hessian matrix of τ in (16) as $\boldsymbol{g}$ and $\mathbf{H}$, and denote the 6-dimensional gradient vector for $\boldsymbol{x}_j$ as $\boldsymbol{g}_j$ and the 6×6 Hessian matrix block for $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ as $\mathbf{H}_{jk}$ (note that here j can equal to k). Then $\boldsymbol{g}$ and $\mathbf{H}$ can be written in the block form as $\boldsymbol{g} = (\boldsymbol{g}_j) \in \mathbb{R}^{6N}$ and $\mathbf{H} = (\mathbf{H}_{jk}) \in \mathbb{R}^{6N \times 6N}$.

The ith plane $\boldsymbol{\pi}_i$ is observed by the poses in $\mathbb{X}_i$. Assume $\boldsymbol{x}_j \in \mathbb{X}_i$ and $\boldsymbol{x}_k \in \mathbb{X}_i$. Let us define

$$\boldsymbol{g}_j^i = \frac{\partial \lambda_{i,3}}{\partial \boldsymbol{x}_j}, \ \mathbf{H}_{jk}^i = \frac{\partial^2 \lambda_{i,3}}{\partial \boldsymbol{x}_j \partial \boldsymbol{x}_k}. \tag{17}$$

According to (16), we have

$$\boldsymbol{g}_j = \sum_{i \in \mathbb{P}_j} \boldsymbol{g}_j^i, \ \mathbf{H}_{jk} = \sum_{i \in \mathbb{P}_{jk}} \mathbf{H}_{jk}^i, \tag{18}$$

where $\mathbb{P}_j$ is the set of planes observed by $\boldsymbol{x}_j$, and $\mathbb{P}_{jk}$ is the set of planes observed by $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ simultaneously. If $j = k$, here $\mathbb{P}_{jk}$ equals to $\mathbb{P}_j$. From (18), we know that the key point to get $\boldsymbol{g}$ and $\mathbf{H}$ is to compute $\boldsymbol{g}_j^i$ and $\mathbf{H}_{jk}^i$ in (17).

4.1. Partial Derivatives of Eigenvalue

According to (15), $\lambda_{i,3}$ is a function of poses in $\mathbb{X}_i$. Assume that x_{jm} and x_{kn} are the mth and nth elements of $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, respectively. We consider the first- and second-order partial derivation $\frac{\partial \lambda_{i,3}}{\partial x_{jm}}$ and $\frac{\partial^2 \lambda_{i,3}}{\partial x_{jm} \partial x_{kn}}$, respectively.

$\lambda_{i,3}$ is a root of the equation $|\mathbf{M}_i(\mathbb{X}_i) - \lambda_i \mathbf{I}_3| = 0$, where $|\cdot|$ denotes the determinant of a matrix. Assume m_{ef} is the eth row fth column term of $\mathbf{M}_i(\mathbb{X}_i)$. $|\mathbf{M}_i(\mathbb{X}_i) - \lambda_i \mathbf{I}_3| = 0$ is a cubic equation with the following form

$$-\lambda_{i,3}^3 + a_i \lambda_{i,3}^2 + b_i \lambda_{i,3} + c_i = 0, \tag{19}$$

where $a_i = m_{11} + m_{22} + m_{33}$, $b_i = m_{12}^2 + m_{13}^2 + m_{23}^2 - m_{11}m_{22} - m_{11}m_{33} - m_{22}m_{33}$, and $c_i = -m_{33}m_{12}^2 + 2m_{12}m_{13}m_{23} - m_{22}m_{13}^2 - m_{11}m_{23}^2 + m_{11}m_{22}m_{33}$. Here a_i, b_i and c_i are all functions of the poses in $\mathbb{X}_i$. It is known that the root of a cubic equation has a closed form. One solution to compute $\frac{\partial \lambda_{i,3}}{\partial x_{jm}}$ and $\frac{\partial^2 \lambda_{i,3}}{\partial x_{jm}\partial x_{kn}}$ is to directly differentiate the root. However, the formula of the root is too complicated. Here we introduce a simple way to compute them. Briefly, we employ the implicit function theorem [15] to compute them. Let us define

$$\boldsymbol{\chi}_i = \begin{bmatrix} \lambda_{i,3}^2 \\ \lambda_{i,3} \\ 1 \end{bmatrix}, \ \boldsymbol{\eta}_i = \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \boldsymbol{\kappa}_i = \begin{bmatrix} -3 \\ 2a_i \\ b_i \end{bmatrix}, \boldsymbol{\delta}_{jm}^i = \frac{\partial \boldsymbol{\eta}_i}{\partial x_{jm}}. \tag{20}$$

Using the above notations, we present $\frac{\partial \lambda_{i,3}}{\partial x_{jm}}$ and $\frac{\partial^2 \lambda_{i,3}}{\partial x_{jm}\partial x_{kn}}$ in Lemma 1 and 2. The proofs of the following lemmas and theorems are in the **supplementary material**.

Lemma 1 *Using the notations in (20), we have*

$$\frac{\partial \lambda_{i,3}}{\partial x_{jm}} = -\varphi_i \boldsymbol{\delta}_{jm}^i \cdot \boldsymbol{\chi}_i, \tag{21}$$

where $\cdot$ represents the dot product and $\varphi_i = (\boldsymbol{\kappa}_i \cdot \boldsymbol{\chi}_i)^{-1}$.

Lemma 2 *Using the notations in (20) and (21), we have*

$$\frac{\partial^2 \lambda_{i,3}}{\partial x_{jm}\partial x_{kn}} = -\varphi_i \left(\boldsymbol{\delta}_{jm}^i \cdot \frac{\partial \boldsymbol{\chi}_i}{\partial x_{kn}} + \boldsymbol{\chi}_i \cdot \frac{\partial \boldsymbol{\delta}_{jm}^i}{\partial x_{kn}} - \frac{\partial \lambda_{i,3}}{\partial x_{jm}} \frac{\partial \varphi_i^{-1}}{\partial x_{kn}} \right). \tag{22}$$

Let us define

$$\begin{aligned} &\boldsymbol{\alpha}_j^i = \frac{\partial a_i}{\partial \boldsymbol{x}_j}, \boldsymbol{\beta}_j^i = \frac{\partial b_i}{\partial \boldsymbol{x}_j}, \boldsymbol{\gamma}_j^i = \frac{\partial c_i}{\partial \boldsymbol{x}_j}, \boldsymbol{\Delta}_j^i = [\boldsymbol{\alpha}_j^i, \boldsymbol{\beta}_j^i, \boldsymbol{\gamma}_j^i], \\ &\boldsymbol{\alpha}_k^i = \frac{\partial a_i}{\partial \boldsymbol{x}_k}, \boldsymbol{\beta}_k^i = \frac{\partial b_i}{\partial \boldsymbol{x}_k}, \boldsymbol{\gamma}_k^i = \frac{\partial c_i}{\partial \boldsymbol{x}_k}, \boldsymbol{\Delta}_k^i = [\boldsymbol{\alpha}_k^i, \boldsymbol{\beta}_k^i, \boldsymbol{\gamma}_k^i], \\ &\mathbf{H}_{jk}^{a_i} = \frac{\partial^2 a_i}{\partial \boldsymbol{x}_j \partial \boldsymbol{x}_k}, \mathbf{H}_{jk}^{b_i} = \frac{\partial^2 b_i}{\partial \boldsymbol{x}_j \partial \boldsymbol{x}_k}, \mathbf{H}_{jk}^{c_i} = \frac{\partial^2 c_i}{\partial \boldsymbol{x}_j \partial \boldsymbol{x}_k}. \end{aligned} \tag{23}$$

Using the above lemmas and notations, we can derive $\boldsymbol{g}_j^i$ and $\mathbf{H}_{jk}^i$.

Theorem 1 *Using the notations in (20), (21) and (23), $\boldsymbol{g}_j^i$ and $\mathbf{H}_{jk}^i$ have the forms*

$$\begin{aligned} \boldsymbol{g}_j^i &= -\varphi_i \boldsymbol{\Delta}_j^i \boldsymbol{\chi}_i, \\ \mathbf{H}_{jk}^i &= \varphi_i \left(\mathbf{K}_{jk}^i - \lambda_{3,i}^2 \mathbf{H}_{jk}^{a_i} - \lambda_{3,i} \mathbf{H}_{jk}^{b_i} - \mathbf{H}_{jk}^{c_i} \right), \end{aligned} \tag{24}$$

where $\mathbf{K}_{jk}^i = \boldsymbol{g}_j^i \boldsymbol{u}^T - \boldsymbol{v}(\boldsymbol{g}_k^i)^T$, $\boldsymbol{u} = 2\lambda_{i,3}\boldsymbol{\alpha}_k^i + \boldsymbol{\beta}_k^i + (2a - 6\lambda_{i,3})\boldsymbol{g}_k^i$, and $\boldsymbol{v} = 2\lambda_{i,3}\boldsymbol{\alpha}_j^i + \boldsymbol{\beta}_j^i$, and similar to $\boldsymbol{g}_j^i$, $\boldsymbol{g}_k^i = -\varphi_i \boldsymbol{\Delta}_k^i \boldsymbol{\chi}_i$ is the gradient block for $\boldsymbol{x}_k$.

The formula of $\mathbf{H}_{jk}^i$ in (24) is applicable to the case that $j = k$. From Theorem 1, we know that the key point to get $\boldsymbol{g}_j^i$ and $\mathbf{H}_{jk}^i$ is to get the derivatives of a_i, b_i and c_i in (23).

4.2. Partial Derivatives of a_i, b_i and c_i

According to (19), a_i, b_i and c_i are functions of the elements in $\mathbf{M}_i$. Using this relationship, we can easily derive the partial derivatives in (23). For instance, as $a_i = m_{11} + m_{22} + m_{33}$, we have

$$\frac{\partial a_i}{\partial \boldsymbol{x}_j} = \frac{\partial m_{11}}{\partial \boldsymbol{x}_j} + \frac{\partial m_{22}}{\partial \boldsymbol{x}_j} + \frac{\partial m_{33}}{\partial \boldsymbol{x}_j}. \tag{25}$$

Thus, to get the first- and second-order partial derivatives of a_i, b_i and c_i with respect to $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ in (23), we need to derive the form of $\mathbf{M}_i$ with respect to $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$.

Lemma 3 *In terms of $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, $\bar{\boldsymbol{p}}_i$ in (13) has the form*

$$\bar{\boldsymbol{p}}_i(\boldsymbol{x}_j, \boldsymbol{x}_k) = \mathbf{T}_j \boldsymbol{q}_{ij} + \mathbf{T}_k \boldsymbol{q}_{ik} + \boldsymbol{c}_{ijk}, \tag{26}$$

where $\boldsymbol{q}_{ij} = \frac{1}{N_i}\tilde{\boldsymbol{p}}_{ij}$, $\boldsymbol{q}_{ik} = \frac{1}{N_i}\tilde{\boldsymbol{p}}_{ik}$, and $\boldsymbol{c}_{ijk} = \frac{1}{N_i}\sum_{n \in \mathbb{O}_{jk}} \mathbf{T}_n \tilde{\boldsymbol{p}}_{in}$. Here $\mathbb{O}_{jk} = obs(\boldsymbol{\pi}_i) - \{j, k\}$ represents the set of indexes of the poses that can observe $\boldsymbol{\pi}_i$, excluding the $\boldsymbol{j}$th and $\boldsymbol{k}$th poses.

In terms of $\boldsymbol{x}_j$, $\bar{\boldsymbol{p}}_i$ has the form

$$\bar{\boldsymbol{p}}_i(\boldsymbol{x}_j) = \mathbf{T}_j \mathbf{q}_{ij} + \boldsymbol{c}_{ij}, \tag{27}$$

where $\boldsymbol{c}_{ij} = \mathbf{T}_k \mathbf{q}_{ik} + \boldsymbol{c}_{ijk}$.

Using Lemma 3, we can have the following theorem for $\mathbf{M}_i$ in (12).

Theorem 2 *In terms of $\boldsymbol{x}_j$, $\mathbf{M}_i$ in (12) can be written as*

$$\mathbf{M}_i(\boldsymbol{x}_j) = \mathbf{T}_j \mathbf{Q}_j^i \mathbf{T}_j^T + \mathbf{T}_j \mathbf{K}_j^i + (\mathbf{K}_j^i)^T \mathbf{T}_j^T + \mathbf{C}_j^i, \tag{28}$$

where $\mathbf{Q}_j^i = \mathbf{U}_{ij} - N_j \boldsymbol{q}_{ij}\boldsymbol{q}_{ij}^T$ and $\mathbf{K}_j^i = -N_i \boldsymbol{q}_{ij}\boldsymbol{c}_{ij}^T$. Here $\mathbf{U}_{ij}$ and $\boldsymbol{q}_{ij}$ are defined in (13) and (26), respectively.

In terms of $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, $\mathbf{M}_i$ can be written as

$$\mathbf{M}_i(\boldsymbol{x}_j, \boldsymbol{x}_k) = \mathbf{T}_j \mathbf{O}_{jk}^i \mathbf{T}_k^T + \mathbf{T}_k (\mathbf{O}_{jk}^i)^T \mathbf{T}_j^T + \mathbf{C}_{jk}^i. \tag{29}$$

where $\mathbf{O}_{jk}^i = -N_i \boldsymbol{q}_{ij}\boldsymbol{q}_{ik}^T$.

Here we do not provide the detailed formulas for $\mathbf{C}_j^i$ and $\mathbf{C}_{jk}^i$, as they will be eliminated in the partial derivative. Actually, only $\mathbf{Q}_j^i$, $\mathbf{K}_j^i$, and $\mathbf{O}_{jk}^i$ are required to compute the partial derivatives in (23). Equation (28) is used to compute the first- and second-order partial derivatives of a_i, b_i and c_i with respect to $\boldsymbol{x}_j$. Equation (29) is used to compute the second-order partial derivatives of a_i, b_i and c_i with respect to $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$.

4.3. Efficient Iteration

From Theorem 2, we can easily derive the elements of $\mathbf{M}_i(\boldsymbol{x}_j)$ and $\mathbf{M}_i(\boldsymbol{x}_j, \boldsymbol{x}_i)$. Assume $a(\boldsymbol{x}_j)$ and $b(\boldsymbol{x}_j, \boldsymbol{x}_k)$ are one of the elements in $\mathbf{M}_i(\boldsymbol{x}_j)$ and $\mathbf{M}_i(\boldsymbol{x}_j, \boldsymbol{x}_k)$, respectively. Substituting $\mathbf{T}_j$ and $\mathbf{T}_k$ defined in (1) into (28) and

(29) and expanding them, we can obtain that $a(\boldsymbol{x}_j)$ and $b(\boldsymbol{x}_j, \boldsymbol{x}_k)$ are second-order polynomials in terms of the elements in $\mathbf{T}_j$ and $\mathbf{T}_k$. Using the CGB parameterization in (2), we can write $a(\boldsymbol{x}_j)$ and $b(\boldsymbol{x}_j, \boldsymbol{x}_k)$ as

$$\begin{aligned} a(\boldsymbol{x}_j) &= \boldsymbol{c} \cdot \boldsymbol{h}(\boldsymbol{x}_j) + c_0, \\ b(\boldsymbol{x}_j, \boldsymbol{x}_k) &= \boldsymbol{d} \cdot \boldsymbol{g}(\boldsymbol{x}_j, \boldsymbol{x}_k) + d_0, \end{aligned} \tag{30}$$

where $\boldsymbol{c}$ is determined by $\mathbf{Q}_j^i$ and $\mathbf{K}_j^i$ in (28), $\boldsymbol{d}$ is determined by $\mathbf{O}_{jk}^i$ in (29), $\boldsymbol{h}(\boldsymbol{x}_j)$ and $\boldsymbol{g}(\boldsymbol{x}_j, \boldsymbol{x}_k)$ are two vector functions, c_0 and d_0 are two scalars independent on $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$. Let us first consider the first-order partial derivative of $a(\boldsymbol{x}_j)$ with respect to $\boldsymbol{x}_j$. It has the form

$$\frac{\partial a(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j} = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j} \boldsymbol{c}, \tag{31}$$

where the vector-by-vector derivative $\frac{\partial \boldsymbol{h}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}$ is defined in (4). To efficiently compute (31), we consider a special pose $\mathbf{T}_0 = [\mathbf{R}_0, \boldsymbol{t}_0]$ where $\mathbf{R}_0 = \mathbf{I}_3$ and $\boldsymbol{t}_0 = [0; 0; 0]$. Let us denote the parameterization of $\mathbf{T}_0$ as $\boldsymbol{x}_0$. As the CGR parameterization for $\mathbf{I}_3$ is $[0; 0; 0]$, we have $\boldsymbol{x}_0 = [0; 0; 0; 0; 0; 0]$. For $\boldsymbol{x}_j = \boldsymbol{x}_0$, the matrix $\frac{\partial \boldsymbol{h}(\mathbf{x}_j)}{\partial \boldsymbol{x}_j}$ can be easily computed. Similarly, the second-order partial derivatives of $\boldsymbol{h}(\boldsymbol{x}_j)$ and $\boldsymbol{g}(\boldsymbol{x}_j, \boldsymbol{x}_k)$ at $\boldsymbol{x}_j = \boldsymbol{x}_0$ and $\boldsymbol{x}_k = \boldsymbol{x}_0$ are also simple. As $\boldsymbol{h}(\boldsymbol{x}_j)$ and $\boldsymbol{g}(\boldsymbol{x}_j, \boldsymbol{x}_k)$ only depend on $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, we can compute the partial derivatives at $\boldsymbol{x}_0$ once, and then reuse them in the following iterations. Here we introduce a method to make the iteration stay at $\mathbf{x}_0$ for each pose.

Assume that $\{\mathbf{X}_j^n\}$ are the poses after the nth iteration. Then we can update $\mathbf{U}_{ij}$ and $\tilde{\boldsymbol{p}}_{ij}$ in (13) by

$$\mathbf{U}_{ij}^{n+1} = \mathbf{X}_j^n \mathbf{U}_{ij} (\mathbf{X}_j^n)^T \text{ and } \tilde{\boldsymbol{p}}_{ij}^{n+1} = \mathbf{X}_j^n \tilde{\boldsymbol{p}}_{ij}. \tag{32}$$

Substituting $\mathbf{U}_{ij}^{n+1}$ and $\tilde{\boldsymbol{p}}_{ij}^{n+1}$ into (12), we get a new matrix $\mathbf{M}_i(\mathbb{X}_i)^{n+1}$, which can finally lead to a new cost τ^{n+1} in (16). As the points have been transformed by $\{\mathbf{X}_j^n\}$, each pose should start with $\mathbf{X}_0$ for τ^{n+1}. Assume that $\Delta \boldsymbol{x}_j^{n+1}$ is the result from the $(n+1)$th iteration for the jth pose. We can compute the corresponding transformation matrix $\Delta \mathbf{X}_j^{n+1}$ using (1) and (2). Then we can update $\mathbf{X}_j^n$ by

$$\mathbf{X}_j^{n+1} = \Delta \mathbf{X}_j^{n+1} \mathbf{X}_j^n. \tag{33}$$

Furthermore, the update steps in (32) will not introduce additional computation. This is because the damped Newton's method requires to compute the cost τ in (16) to adjust μ in (3) after each iteration, which requires to perform the computation in (32).

4.4. Algorithm Summary

We first construct $\mathbf{H}$ and $\boldsymbol{g}$. For each plane $\boldsymbol{\pi}_i$, we solve the cubic equation (19), and select the smallest root $\lambda_{i,3}$. For $\boldsymbol{x}_j$, we construct $\mathbf{M}(\boldsymbol{x}_j)$ in (28), and calculate the partial derivatives of a_i, b_i and c_i with respect to $\boldsymbol{x}_j$ in (23). Then, we use (24) to compute $\boldsymbol{g}_j^i$ and $\mathbf{H}_{jj}^i$ and use (18) to update $\boldsymbol{g}_j$ and $\mathbf{H}_{jj}$. For $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, $\mathbf{M}_i(\boldsymbol{x}_j, \boldsymbol{x}_k)$ is generated, and then the partial derivatives of a_i, b_i and c_i with respect to $\boldsymbol{x}_j$ and $\boldsymbol{x}_j$ in (23) are computed. Then, $\mathbf{H}_{jk}^i$ can be easily obtained from (24), and $\mathbf{H}_{jk}$ in (18) is computed accordingly. Using $\mathbf{H}$ and $\boldsymbol{g}$, we conduct the damped Newton's step in (3). After each iteration, $\mathbf{U}_{ij}$ and $\tilde{\boldsymbol{p}}_{ij}$ are updated by (32). The proposed algorithm is summarized in Algorithm 1. Let us denote the mean and variance of the number of observations per plane as $\bar{K}$ and σ^2, respectively. According to [8], the computational complexity of the Hessian matrix is $O(M(\bar{K}^2 + \sigma^2))$, which is of the same order as the Schur complement trick.

Algorithm 1: Second-order plane adjustment for N poses and M planes

while *not converge* **do**
  $\mathbf{H} = zeros(6N, 6N)$, $\boldsymbol{g} = zeros(6N, 1)$;
  for $i \in [1, M]$ **do**
    `/* Compute` $\boldsymbol{g}$ `and the diagonal terms of` $\mathbf{H}$. `*/`
    for $j \in obs(\boldsymbol{\pi}_i)$ **do**
      Compute $\mathbf{M}_i(\boldsymbol{x}_j)$ using (28);
      Compute $\boldsymbol{\alpha}_j^i, \boldsymbol{\beta}_j^i, \boldsymbol{\gamma}_j^i, \mathbf{H}_{jj}^{a_i}, \mathbf{H}_{jj}^{b_i}, \mathbf{H}_{jj}^{c_i}$ using (23);
      Compute $\mathbf{H}_{jj}^i$ and $\boldsymbol{g}_j^i$ using (24);
      $\mathbf{H}_{jj} = \mathbf{H}_{jj} + \mathbf{H}_{jj}^i$, $\boldsymbol{g}_j = \boldsymbol{g}_j + \boldsymbol{g}_j^i$;
    `/* Compute other terms of` $\mathbf{H}$. `*/`
    for $j \in obs(\boldsymbol{\pi}_i)$ **do**
      for $k \in obs(\boldsymbol{\pi}_i)$ *and* $k > j$ **do**
        Compute $\mathbf{H}_{jk}^i$ using (24);
        $\mathbf{H}_{jk} = \mathbf{H}_{jk} + \mathbf{H}_{jk}^i$;
        $\mathbf{H}_{kj} = \mathbf{H}_{kj} + (\mathbf{H}_{jk}^i)^T$;
  Conduct the damped Newton's step in (3) ;
  Update $\mathbf{U}_{ij}$ and $\tilde{\boldsymbol{p}}_{ij}$ using (32);

5. Experiments

5.1. Setup

In this section, we compare our algorithm with **EF** [10], **BALM** [16] and the LM solution [23] with plane fitting after a successful LM step (named **LM+PF**). Our damped Newton's method was implemented according to the LM algorithm in Ceres [2]. The damped Newton's method and the LM algorithm are with the same parameters. Specifically, the initial value of the damping factor μ in (3) is set to 10^{-4}. The early stopping tolerances (such as the cost value change and the norm of gradient) are set to 10^{-7}. The

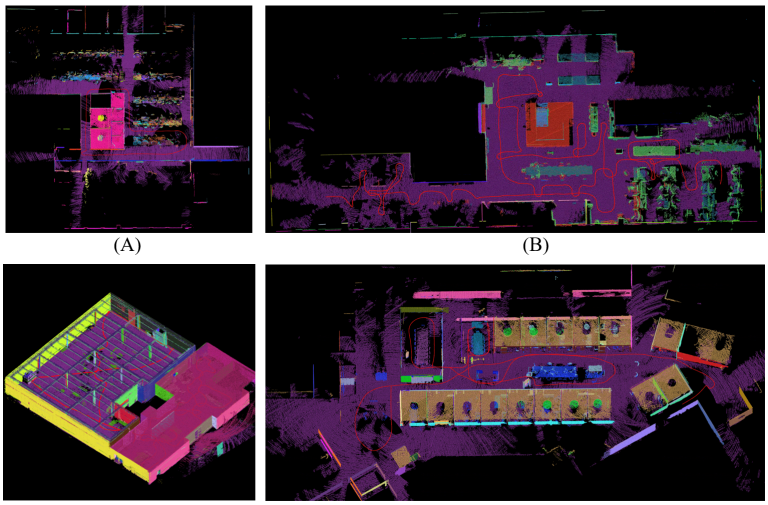


Figure 3. The four datasets used in this paper. There are 4.5×10^6, 16.6×10^6, 16.7×10^6, and 10.5×10^6 points in the 4 datasets. The 4 datasets have 339, 369, 856, and 589 planes, and 472, 1355, 1606, 1184 poses, respectively. Roofs are removed to show the trajectories.

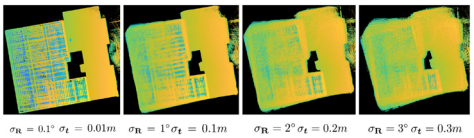


Figure 4. The point clouds of dataset C after the poses were perturbs by the four noise levels.

maximum number of iterations is set to 200 for our algorithm, BALM, and LM+PF, and 10^5 for EF, as EF uses the first-order minimization which requires more iterations to converge. All the experiments were conducted on a desktop with an Intel i7 cpu and 64G memory.

5.2. Datasets

We collected four datasets using a VLP-16 LiDAR. We used the LiDAR SLAM algorithm [24] to detect the planes and establish the plane association. Fig. 3 shows the four datasets. Similar to the evaluation of BA algorithms [3, 8, 25], we perturb the pose, and compare the PA cost in (11) for different algorithms. Specifically, we directly add Gaussian noises to the translation, and randomly generate an angle-axis vector from a Gaussian distribution to perturb the rotation. After the poses are perturb, we use (7) to get the initial plane parameters for LM+PF.

We evaluate the performance of different algorithms under different noise levels. Let us denote the standard deviation (std) of the Gaussian noises for rotation and translation as $\sigma_{\mathbf{R}}$ and $\sigma_{\boldsymbol{t}}$, respectively. We consider four noise levels: $\sigma_{\mathbf{R}} = 0.1^\circ$ and $\sigma_{\boldsymbol{t}} = 0.01m$, $\sigma_{\mathbf{R}} = 1^\circ$ and $\sigma_{\boldsymbol{t}} = 0.1m$, $\sigma_{\mathbf{R}} = 2^\circ$ and $\sigma_{\boldsymbol{t}} = 0.2m$, and $\sigma_{\mathbf{R}} = 3^\circ$ and $\sigma_{\boldsymbol{t}} = 0.3m$. Fig. 4 demonstrates the point clouds of dataset C after the poses are perturbed by the four noise levels. To evaluate the performance of different algorithms suffering from large measurement noises, we also test the case that additional Gaussian noises with the std $\sigma_{pt} = 0.05m$ are added to the LiDAR point cloud, and the poses are perturbed by Gaussian noises with $\sigma_{\mathbf{R}} = 3^\circ$ and $\sigma_{\boldsymbol{t}} = 0.3m$.

5.3. Results

Fig. 5 and Fig. 6 illustrates the results. It is clear that our algorithm converges faster than other algorithms. LM+PF works well at small noises (such as $\sigma_{\mathbf{R}} = 0.1^\circ$ and $\sigma_{\boldsymbol{t}} = 0.01m$). As the noise level increases, LM+PF tends to converge slower. Constructing $\mathbf{H}$ and $\mathbf{g}$ using BALM [16] is computationally demanding. For efficiency, BALM only keeps the centroid of each plane observation (*i.e.*, only 1 point is kept), which results in bad performance. In Fig. 5 and Fig. 6, all points are used to compute the cost in (11) after an iteration of BALM. BALM generally converges slow, as it minimizes a reduced PA problem. EF [10] adopts the first-order optimization method as it only provides a method

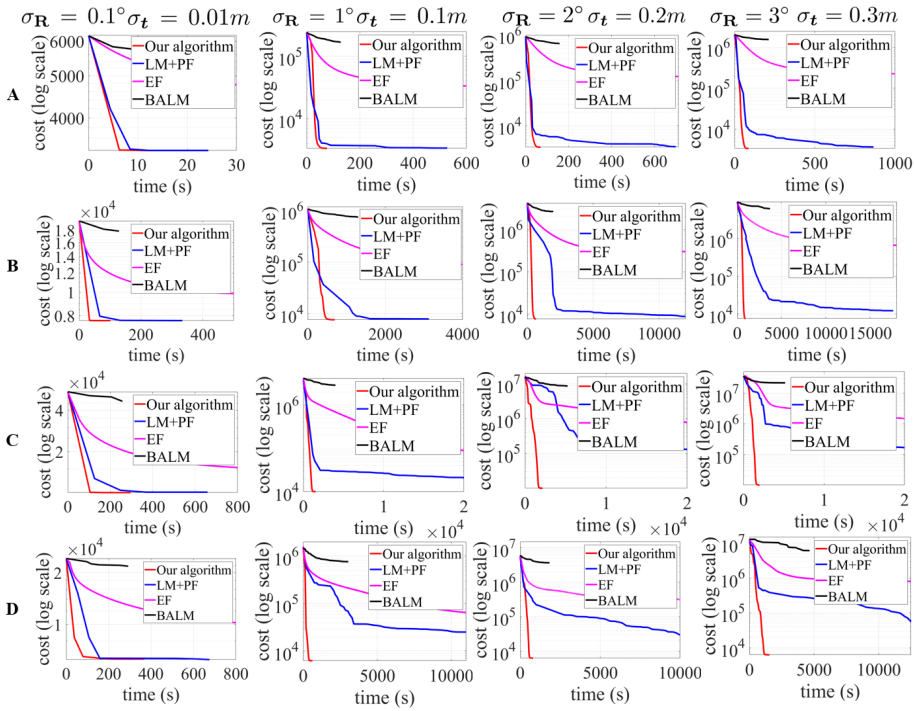


Figure 5. The results of different algorithms using different initial noise levels. It is clear that our algorithm converges significantly faster than other algorithms. The y-axis represents the logarithmic scale of the cost in (11).

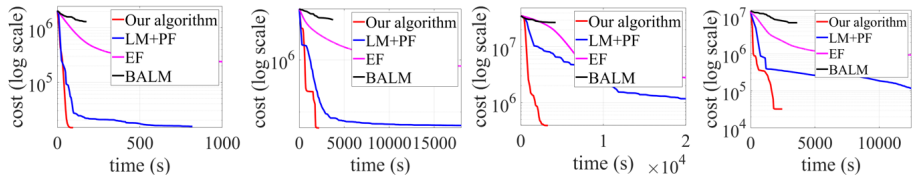


Figure 6. The results of different algorithms for point clouds perturbed by Gaussian noise with std $\sigma_{pt} = 0.05m$. The rotations and translations of the poses are perturbed by Gaussian noise with std$\sigma_{\mathbf{R}} = 3°$ and $\sigma_{\boldsymbol{t}} = 0.3m$, respectively.

to compute the gradient, which results in slow convergence.

6. Conclusion

In the computer vision community, Newton's method is generally considered too expensive for a large-scale least-squares problem. This paper adopts the Newton's method to efficiently solve the PA problem. Our algorithm takes advantage of the fact that the optimal planes are determined by the poses, so that the number of unknowns can be significantly reduced. Furthermore, this property can ensure to obtain the optimal planes when we update the poses. The difficulty lies in how to efficiently compute the Hessian matrix and the gradient vector. The key contribution of this paper is to provide a closed-form solution for them. The experimental results show that our algorithm outperforms the state-of-the-art algorithms.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 1

[2] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022. 6

[3] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *European conference on computer vision*, pages 29–42. Springer, 2010. 1, 2, 7

[4] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001. 2

[5] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 2021. 1

[6] Danpeng Chen, Shuai Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Hujun Bao, and Guofeng Zhang. Vip-slam: An efficient tightly-coupled rgb-d visual inertial planar slam. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5615–5621. IEEE, 2022. 1

[7] Nikolaus Demmel, David Schubert, Christiane Sommer, Daniel Cremers, and Vladyslav Usenko. Square root marginalization for sliding-window bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13260–13268, 2021. 2

[8] Nikolaus Demmel, Christiane Sommer, Daniel Cremers, and Vladyslav Usenko. Square root bundle adjustment for large-scale reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11723–11732, 2021. 1, 2, 6, 7

[9] Hakim Elchaoui Elghor, David Roussel, Fakhreddine Ababsa, and El Houssine Bouyakhf. Planes detection for robust localization and mapping in rgb-d slam systems. In *2015 International Conference on 3D Vision*, pages 452–459. IEEE, 2015. 1

[10] Gonzalo Ferrer. Eigen-factors: Plane estimation for multi-frame and time-continuous point cloud alignment. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1278–1284. IEEE, 2019. 1, 2, 6, 7

[11] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390. IEEE, 2011. 3

[12] Ming Hsiao, Eric Westman, Guofeng Zhang, and Michael Kaess. Keyframe-based dense planar slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5110–5117. IEEE, 2017. 1, 2

[13] Michael Kaess. Simultaneous localization and mapping with infinite planes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4605–4611. IEEE, 2015. 1, 2

[14] Pyojin Kim, Brian Coltin, and H Jin Kim. Linear rgb-d slam for planar environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 333–348, 2018. 1

[15] Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002. 5

[16] Zheng Liu and Fu Zhang. BALM: Bundle Adjustment for LiDAR Mapping. *IEEE Robotics and Automation Letters*, 6(2):3184–3191, 2021. 1, 2, 6, 7

[17] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978. 3

[18] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[19] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[20] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999. 1, 2, 3

[21] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, 2014. 1

[22] Lipu Zhou, Guoquan Huang, Yinian Mao, Jincheng Yu, Shengze Wang, and Michael Kaess. Plc-lislam: Lidar slam with planes, lines and cylinders. *IEEE Robotics and Automation Letters*, 2022. 1

[23] Lipu Zhou, Daniel Koppel, Hul Ju, Frank Steinbruecker, and Michael Kaess. An efficient planar bundle adjustment algorithm. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 136–145, 2020. 1, 2, 6

[24] Lipu Zhou, Daniel Koppel, and Michael Kaess. Lidar slam with plane adjustment for indoor environment. *IEEE Robotics and Automation Letters*, 6(4):7073–7080, 2021. 1, 2, 7

[25] Lei Zhou, Zixin Luo, Mingmin Zhen, Tianwei Shen, Shiwei Li, Zhuofei Huang, Tian Fang, and Long Quan. Stochastic bundle adjustment for efficient and scalable 3d reconstruction. In *European Conference on Computer Vision*, pages 364–379. Springer, 2020. 1, 2, 7

[26] Lipu Zhou, Shengze Wang, and Michael Kaess. π-lsam: Lidar smoothing and mapping with planes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5751–5757. IEEE, 2021. 1