

AAAI'25 | 快手LEARN:使用LLM做特征增强用于电商广告推荐

原创 州懂学习笔记 州懂学习笔记 2024年12月28日 10:30 广东



州懂学习笔记

分享大模型推荐系统相关知识和学习笔记

54篇原创内容

公众号

AAAI'25 | 快手LEARN:使用LLM做特征增强用于电商广告推荐

标题: LEARN: Knowledge Adaptation from Large Language Model to Recommendation for Practical Industrial Application

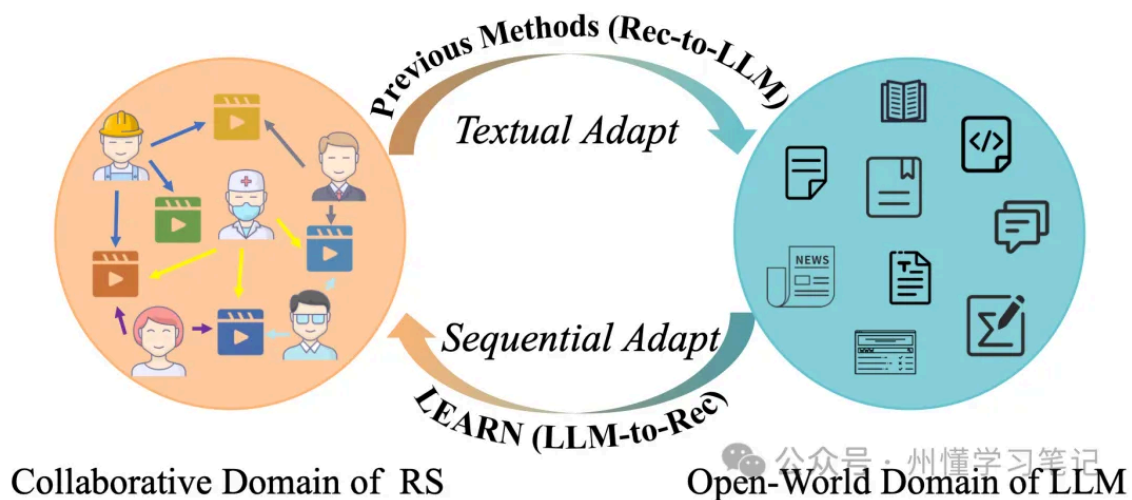
地址: <https://arxiv.org/pdf/2405.03988>

公司: 快手

期刊: AAAI'25

1、前言

现有推荐系统主要还是依赖ID Embed, 这导致在长尾内容和冷启动场景中效果欠佳。随着近年来LLM的突破性进展, 利用LLM强大的内容理解能力辅助推荐被认为是缓解该问题的一个有效途径。



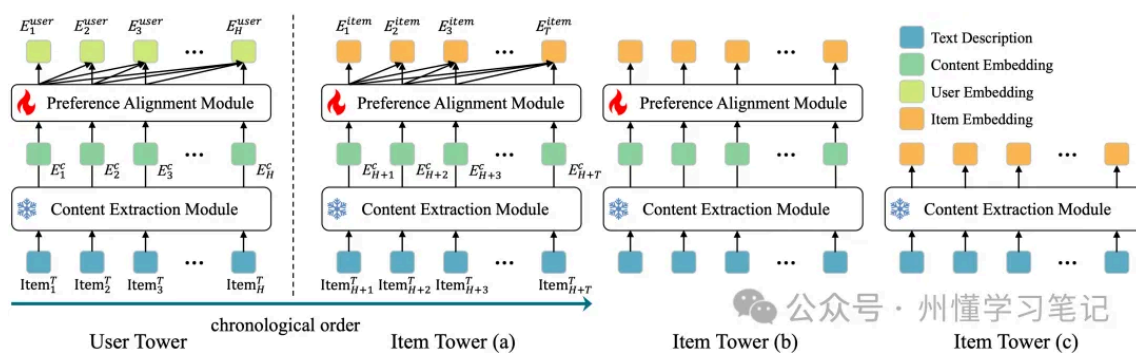
作者提到, 现阶段很多文献都是基于"Rec-to-LLM"的思路来实现的,这种方法通常将推荐域(目标域)的User-Item交互数据转换成LLM开放世界(源域)的文本格式, 并设计特定任务的提示, 将推荐数据转换为会话格式, 以兼容LLM的处理模式。但是, 这种方法存在比较多的缺点:

- **计算效率太低:** 推荐场景下用户的行为序列是非常长的, LLM在处理这么长的用户行为序列非常低效, 无法满足业务落地应用的性能要求。
- **灾难性遗忘:** 推荐系统是User-Item的协同数据主导的, 而LLM学习的是开放世界知识, 这两类知识信息存在非常大的差异, 这样基于用户行为数据去微调LLM常会导致灾难性遗忘开放世界知识。
- **性能下降:** LLM的预训练目标是下一词元预测, 而推荐系统依赖User-Item的协同, 这种训练目标上的不一致性, 使得LLM不能很好的适应推荐任务。

为了克服这些问题, 作者提出了基于"LLM-to-Rec"的思路的LEARN方法(Llm-driven knowlEdge Adaptive RecommEndation)。这种方法使用LLM做特征抽取, 让抽取出来的信息去适应推荐系统本身的训练目标, 更好的兼容推荐系统并满足实际业务落地的性能要求。

2. 方法

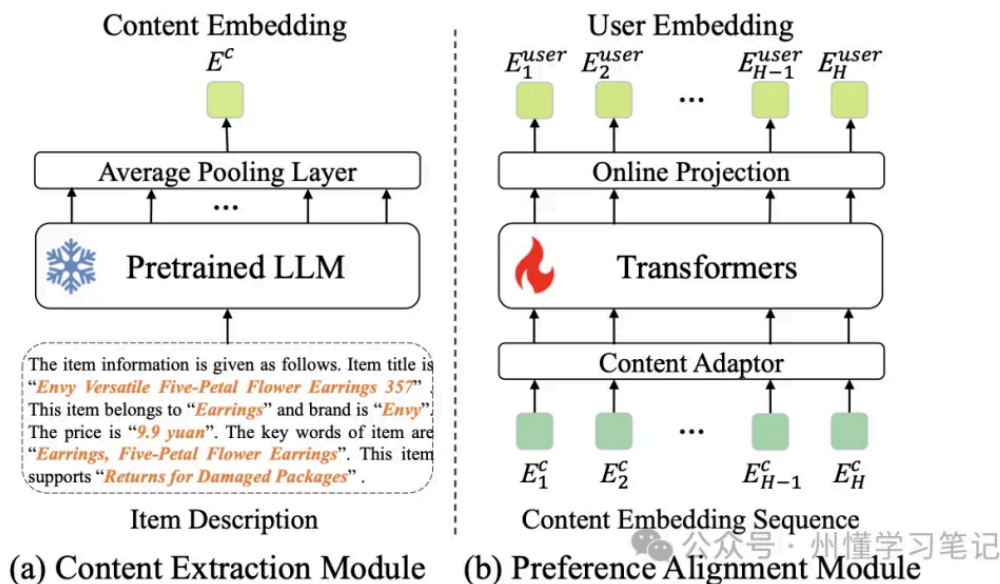
作者所提LEARN方法的整体框架如下图所示, 它是一个双塔结构的模型。



可以看到, 这里会将用户历史行为序列将时间排序后再截断成两部分, 前面部分称为历史交互序列 $U_i^{hist} = \{Item_{i,1}, Item_{i,2}, \dots, Item_{i,H}\}$, 作为用户塔的输入, 后面部分作者把它称为 target 交互序列 $U_i^{tar} = \{Item_{i,H+1}, Item_{i,H+2}, \dots, Item_{i,H+T}\}$, 会作为Item塔的输入。

2.1 User塔

用户塔由两个模块组成, 分别是内容抽取模块和偏好对齐模块, 如下图所示:



2.1.1 内容抽取模块(简称CEX)

对于用户行为序列 U^{hist} 中的每个Item, 先按下图的提示组织其文本描述(包括标题、类别、品牌、价格、关键词和属性):

Item Prompt:

The item information is given as follows. Item title is "{**Title**}". This item belongs to "{**Category**}" and brand is "{**Brand**}". The price is "{**Price**}". The key words of item are "{**Keywords**}". This item supports "{**Attribute**}".

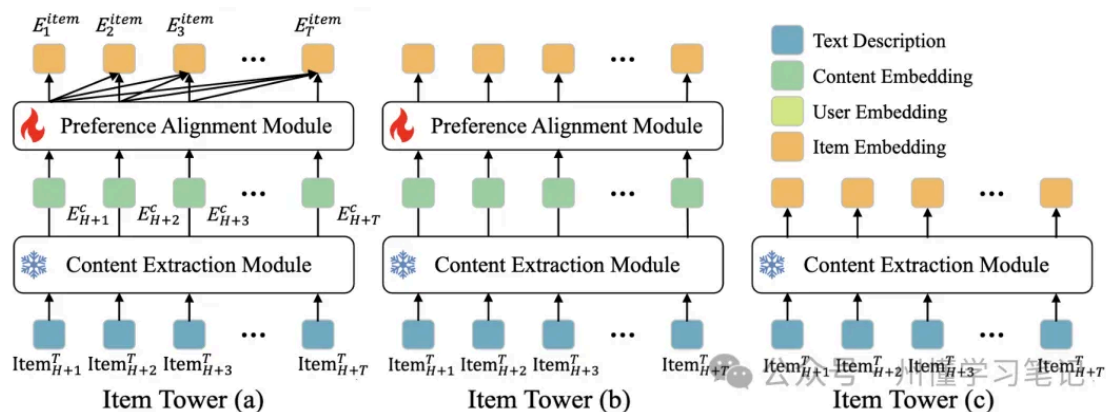
然后, 将这些Item描述输入到参数冻结的预训练LLM(论文使用了Baichuan2-7B)中, 然后再将最后一层的隐含向量做AvgPooling后得到该Item最后的内容表征 E^c 。

2.1.2 偏好对齐模块(简称PAL)

使用前面处理的用户历史行为内容表征序列 $\hat{U}^{\text{hist}} = E_0^c, E_1^c, \dots, E_H^c$ 为输入, 先通过内容映射层做维度变换, 再将它们输入到类似因果注意力机制的Transformer编码器中, 最后使用线性映射降维(至64维)得到User Embed $E^{user} \in \mathbf{R}^{64}$ 。

2.2 Item塔

Item塔以 U_i^{tar} 为输入, 同样也是先过前面的内容抽取模块(CEX), 只是后面这里有3个变种:



- **变体1:** 采用与User塔相同的架构和权重, 都使用了因果注意力机制, 只是这里不是处理用户历史交互序列, 而是Target交互序列
- **变体2:** 同样也是采用与User塔相同的架构和权重, 只是把因果注意力机制替换成只关注自己的内容本身的自注意力机制
- **变体3:** 使用内容Embed作为Item Embed, 直接跳过对齐模块

这样, 变体1相当于让User塔与Item塔去做sequence-to-sequence的对齐, 变体2相当于让User塔与Item塔去做sequence-to-item的对齐, 而变体3相当于要让可学习的User塔去匹配冻结LLM输出的Item塔, 属于Rec-to-LLM的模式。

在训练时, 变体1会将整个用户Target交互序列 U^{tar} 作为输入, 而变体2和变体3则是独立处理Target交互序列的每个Item。而在推理时, 这3个变体都只输入单个Item去生成Item Embedding。

后面的实验效果上, 变体1>变体2>变体3。变体3最差挺好理解的, 毕竟是Rec-to-LLM的范式, 而变体1比变体2好, 作者的解释是"We believe that LEARN with ItemTower(a), which uses sequence-to-sequence alignment, allows the model to better capture long-term user interests compared to the sequence-to-item alignment used in ItemTower(b)". 坦白说, 笔者并不认可这种解释, 笔者认为变体1和变体2在现有数据集方式下的比较本身就不公平。作者的数据集是直接拆分出了前后两个seq, 那变体1使用了因果注意力机制自然是能更好的适配这种seq-2-seq的数据, 而变体2更接近next item prediction的模式, 本身就不是很兼容, 如果把数据集处理成能兼容next item prediction的方式, 变体2的效果不一定会比变体1差, 只能说是, 在作者设定的训练数据集的处理方式下, 变体1会比变体2更适配训练数据。而之所以使用拆分成前后两个序列, 主要是因为下面作者的训练Loss(Pinterest提出的dense all action loss)是用了从前后两个序列采样去做对比学习的训练方式。

2.3 训练目标

在Loss上, 作者使用PinnerFormer的dense all action loss。具体的, 作者会从同一用户的历史交互序列 U^{hist} 中采样了 N_h 个Item(默认为10), 再从该用户的Target交互序列 U^{tar} 中采样了 N_t 个Item(默认为10), 总共构造出 $N_h \times N_t$ 个正样本pair, 再与同一批次中其他用户的Target Item Embedding构造出数量为共 $N_h \times (bs - 1)$ 的负样本pair。这里的采样策略下面会介绍。

再使用对比学习计算对应的Loss:

$$\mathcal{L} = - \sum_{i=1}^{N_u} \sum_{j=1}^{N_h} \sum_{k=1}^{N_h} \log \frac{e^{s(E_{i,j}^{\text{user}}, E_{i,k}^{\text{item}})}}{e^{s(E_{i,j}^{\text{user}}, E_{i,k}^{\text{item}})} + \sum_{z \neq i} \sum_k e^{s(E_{i,j}^{\text{user}}, E_{z,k}^{\text{item}})}},$$

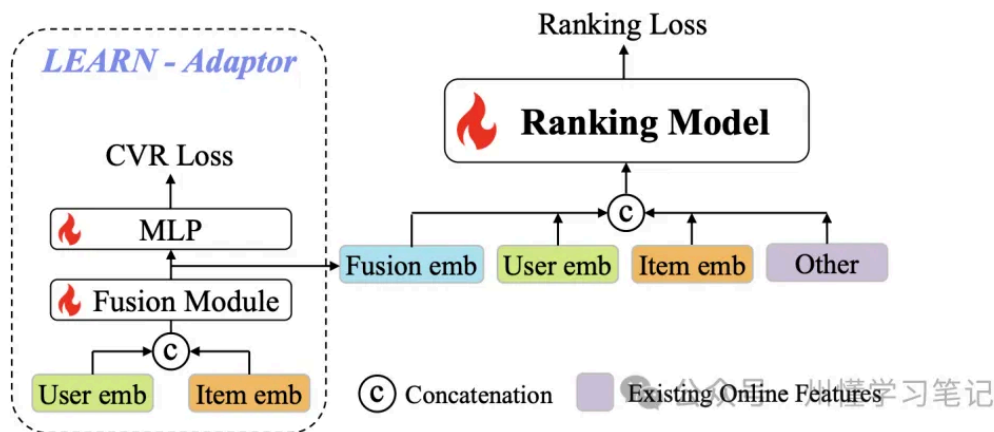
2.4 采样策略

在短视频场景下, 用户交互序列是非常长的, 作者做了前面两阶段采样:

- **Stage1:** 从完整的用户历史/目标交互中执行随机抽样, 作为用户塔的输入, 确保用于建模用户兴趣的数据是无偏的。
- **Stage2:** 在构造正负样本pairs时, 使用样本加权采样方式以提高近期交互Item的采样概率, 因为这更能反应用户的当前兴趣并且与用户偏好的Target Item更相关。这里第*i*个Item的采样概率设为: $\tilde{w}_i = \frac{w_i}{\max(w)}$ 。其中, $w_i = \log\left(\alpha + i \cdot \frac{\beta - \alpha}{N - 1}\right)$, 超参 α 和 β 分别设置为10和10000, 而*N*是从Stage1采样后的序列长度。

2.5 如何在精排中使用

前面的双塔框架是用于召回, 作者还将这些User Embed和Item Embed用于精排模型做特征增强, 不过这里不是直接拿它去使用, 而是在前置环节做了适配, 用了CVR任务来辅助学习这里的融合机制, 具体见下图:



这里有意思的地方是, 为什么是CVR任务, 而不是CTR任务呢。这里笔者的理解是, 由于这里的输入主要是商品详情页的内容信息, 这些内容信息与转化肯定是更相关, 但是如果是图片这种容易骗点击的模态信息的话, 那监督信号就应该用CTR了, 个人理解, 有不同看法欢迎评论区留言。

3. 实验部分

相比于5月份在arxiv第1次提的版本, 这次AAAI的版本在实验对比上更详实了, 但没再和HSTU做对比了, 有些耐人寻味。

3.1 整体效果

与各baseline模型的对比

Dataset	Metric	ID-Only Methods				ID-Text Methods			Text-Only Methods			Improv.
		GRU4Rec	SASRec	BERT4Rec	RecGURU	FDSA	S ³ -Rec	ZESRec	UniSRec	RecFormer	LEARN	
Scientific	NDCG@10	0.0826	0.0797	0.0790	0.0575	0.0716	0.0451	0.0843	0.0862	0.1027	0.1060	+ 3.21%
	Recall@10	0.1055	0.1305	0.1061	0.0781	0.0967	0.0804	0.1260	0.1255	<u>0.1448</u>	0.1594	+10.08%
Instruments	NDCG@10	0.0633	0.0634	0.0707	0.0468	0.0731	0.0797	0.0694	0.0785	0.0830	0.0878	+ 5.78%
	Recall@10	0.0969	0.0995	0.0972	0.0617	0.1006	0.1110	0.1078	<u>0.1119</u>	0.1052	0.1240	+17.87%
Arts	NDCG@10	0.1075	0.0848	0.0942	0.0525	0.0994	0.1026	0.0970	0.0894	0.1252	0.1225	-
	Recall@10	0.1317	0.1342	0.1236	0.0742	0.1209	0.1399	0.1349	0.1333	<u>0.1614</u>	0.1701	+ 5.39%
Office	NDCG@10	0.0761	0.0832	0.0972	0.0500	0.0922	0.0911	0.0865	0.0919	0.1141	0.1167	+ 2.28%
	Recall@10	0.1053	0.1196	0.1205	0.0647	0.1285	0.1186	0.1199	0.1262	<u>0.1403</u>	0.1549	+10.41%
Games	NDCG@10	0.0586	0.0547	0.0628	0.0386	0.0600	0.0532	0.0530	0.0580	0.0684	0.0798	+16.67%
	Recall@10	0.0988	0.0953	0.1029	0.0479	0.0931	0.0879	0.0844	0.0923	<u>0.1039</u>	0.1345	+29.45%
Pet	NDCG@10	0.0648	0.0569	0.0602	0.0366	0.0673	0.0742	0.0754	0.0702	0.0972	0.0990	+ 1.85%
	Recall@10	0.0781	0.0881	0.0765	0.0415	0.0949	0.1039	0.1018	0.0933	<u>0.1162</u>	0.1284	+10.50%

Table 3: Performance comparison of different recommendation models. The best and second best performances are bold and underlined, respectively. Improv. denotes the relative improvement over the SOTA method RecFormer.

NDCG指标

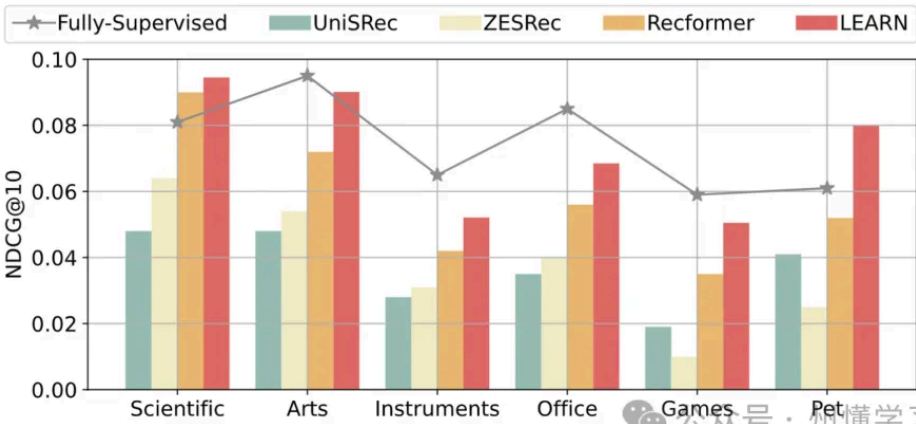


Figure 4: Performance comparison with text-only SOTA methods under zero-shot setting.

LLM embedding对比ID embedding以及bert生成的Embedding

Ablation	Params	H@50	R@50	H@100	R@100
ID-emb	2.3B	0.0312	0.0499	0.0503	0.0754
BERT-emb	86M	0.0357	0.0552	0.0576	0.0843
LLM-emb (Ours)	89M	0.0477	0.0663	0.0751	0.0970

Table 6: Performance comparison of input embedding types in the LEARN framework on the industry dataset. The number of trainable parameters is termed "Params".

PAL模块不同规模参数的LLM的表现

Ablation	Finetune	Params.	H@100	R@100
w/ LLM	LoRA	134M	0.0376	0.0560
		286M	0.0504	0.0709
		572M	0.0513	0.0720
LEARN (Ours)	Full	89M	0.0751	0.0970

Table 7: Ablation studies of the backbone of PAL module. The "w/ LLM" variant uses a pretrained LLM as the backbone of the PAL instead of 12 transformer layers. The number of trainable parameters is termed "Params".

3.2 消融实验

是否有偏好对齐模块, 以及是否加Item tower的消融实验。

Dataset	Metric	w/o Align	w/ ItemTower(c)	LEARN
Scientific	N@10	0.0504	0.0978	0.1060 (+ 8.38%)
	R@10	0.0813	0.1389	0.1594 (+14.76%)
Instruments	N@10	0.0164	0.0679	0.0878 (+29.31%)
	R@10	0.0332	0.0940	0.1240 (+31.91%)
Arts	N@10	0.0308	0.0900	0.1225 (+36.11%)
	R@10	0.0633	0.1337	0.1701 (+27.23%)
Office	N@10	0.0166	0.0890	0.1167 (+31.12%)
	R@10	0.0312	0.1144	0.1549 (+35.40%)
Games	N@10	0.0175	0.0555	0.0798 (+43.78%)
	R@10	0.0361	0.0891	0.1345 (+50.95%)
Pet	N@10	0.0312	0.0812	0.0990 (+21.92%)
	R@10	0.0427	0.1010	0.1284 (+27.13%)

Table 4: Ablation studies of alignment strategy on Amazon Review. *w/o Align* averages the content embeddings of all interacted items to create the user embedding. *w/ ItemTower(c)* take the content embedding of items to supervise the user embedding learning.

不同对齐策略以及采样策略的消融实验

Ablation	H@50	R@50	H@100	R@100
w/o Align	0.0069	0.0154	0.0101	0.0210
w/ ItemTower(c)	0.0292	0.0416	0.0468	0.0626
w/ ItemTower(b)	0.0313	0.0488	0.0505	0.0675
w/ RandomSample	0.0440	0.0610	0.0701	0.0905
LEARN (ours)	0.0477	0.0663	0.0751	0.0970

Table 5: Ablation studies of alignment strategy and sampling strategy on industry dataset.

3.3 在线AB实验

精排模型的AUC指标

Method	UAUC	WUAUC
Baseline	0.6885	0.7002
LEARN (Ours)	0.6969 (+0.84pp)	0.7078 (+0.76pp)

Table 8: AUC results on our e-commerce platform.

拆分冷启和长尾的提升

Level	Type	Proportion	Revenue	AUC
User	cold-start	7.16%	+1.56%	+0.17pp
	long-tail	27.54%	+5.79%	+0.68pp
	others	65.30%	+0.32%	+0.021pp
Item	cold-start	3.15%	+8.77%	+0.29pp
	long-tail	26.47%	+4.63%	+0.21pp
	others	70.38%	+0.35%	+0.01pp

Table 9: Revenue improvement for three user and item types. "Proportion" represents the percentage of users/items in each category.

整体CVR和收入提升

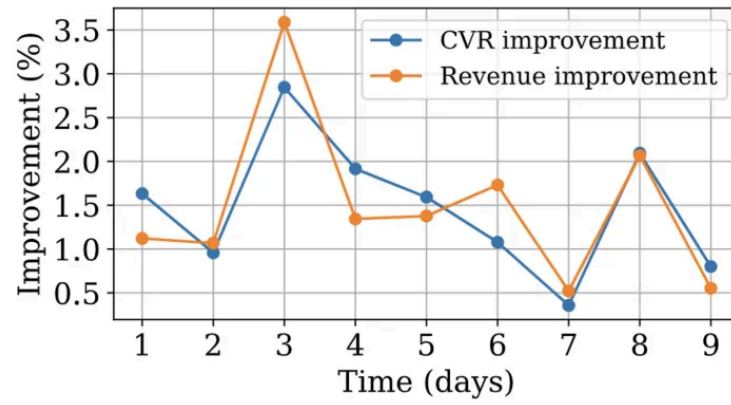


Figure 6: CVR and revenue improvements during online A/B testing. Compared to the baseline method, LEARN achieves a steady and significant increase.

