PLATE: A Prompt-Enhanced Paradigm for Multi-Scenario Recommendations

Yuhao Wang
City University of Hong Kong
yhwang25-c@my.cityu.edu.hk

Xiangyu Zhao ✉
City University of Hong Kong
xianzhao@cityu.edu.hk

Bo Chen
Huawei Noah's Ark Lab
chenbo116@huawei.com

Qidong Liu
City University of Hong Kong
qidongliu2-c@my.cityu.edu.hk

Huifeng Guo
Huawei Noah's Ark Lab
huifeng.guo@huawei.com

Huanshuo Liu
Sun Yat-sen University
liuhsh35@mail2.sysu.edu.cn

Yichao Wang
Huawei Noah's Ark Lab
wangyichao5@huawei.com

Rui Zhang
ruizhang.info
rayteam@yeah.net

Ruiming Tang ✉
Huawei Noah's Ark Lab
tangruiming@huawei.com

ABSTRACT

With the explosive growth of commercial applications of recommender systems, multi-scenario recommendation (MSR) has attracted considerable attention, which utilizes data from multiple domains to improve their recommendation performance simultaneously. However, training a unified deep recommender system (DRS) may not explicitly comprehend the commonality and difference among domains, whereas training an individual model for each domain neglects the global information and incurs high computation costs. Likewise, fine-tuning on each domain is inefficient, and recent advances that apply the prompt tuning technique to improve fine-tuning efficiency rely solely on large-sized transformers. In this work, we propose a novel prompt-enhanced paradigm for multi-scenario recommendation. Specifically, a unified DRS backbone model is first pre-trained using data from all the domains in order to capture the commonality across domains. Then, we conduct prompt tuning with two novel prompt modules, capturing the distinctions among various domains and users. Our experiments on Douban, Amazon, and Ali-CCP datasets demonstrate the effectiveness of the proposed paradigm with two noticeable strengths: (i) its great compatibility with various DRS backbone models, and (ii) its high computation and storage efficiency with only 6% trainable parameters in prompt tuning phase. The implementation code is available for easy reproduction[1,2].

CCS CONCEPTS

• **Information systems → Recommender systems**.

[1]https://gitee.com/mindspore/models/tree/master/research/recommend/PLATE
[2]https://github.com/Applied-Machine-Learning-Lab/PLATE

✉Corresponding authors.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591750

KEYWORDS

Multi-Scenario, Multi-Domain, Cross-Domain Recommendation, Prompt Tuning, Click-Through Rate Prediction

ACM Reference Format:

Yuhao Wang, Xiangyu Zhao ✉, Bo Chen, Qidong Liu, Huifeng Guo, Huanshuo Liu, Yichao Wang, Rui Zhang, and Ruiming Tang ✉. 2023. PLATE: A Prompt-Enhanced Paradigm for Multi-Scenario Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591750

1 INTRODUCTION

Driven by the prevalence of online services and the advances in deep recommender system (DRS) technology, there has been tremendous interest in developing DRS models for various commercial scenarios and domains. However, due to the uneven data distributions in different domains, great efforts have been devoted to cross-domain recommendation (CDR) [2, 20], which utilizes the information in richer domains to enhance the recommendation in sparser domains. This paper focuses on *Multi-Scenario Recommendation* (MSR), which leverages data from multiple domains to simultaneously improve recommendation performance in all domains, *i.e.,* rich or sparse information domains alike. To avoid confusion, it is notable that MSR is exactly *Multi-Target CDR* or *Multi-Domain Recommendation* in research and industry community [30, 38, 42], whereas transferring knowledge from one source domain to promote another target domain, i.e., *Single-Target CDR* [45], is another topic in CDR [10].

Multi-scenario recommendation is faced with challenges in twofold. First, there are a large number of domains with distinctive distributions. Second, there are complex resemblance and exclusion correlations among domains. Existing efforts on MSR can be categorized into three groups: (i) Mix: one common model is built on data from all the domains, (ii) Pre-train&Fine-tune: a common model is first pre-trained using data from all the domains, then fine-tuned for each domain using the data from each individual domain, and (iii) Multi-task learning (MTL): a unified model is built based on multi-task learning. Nevertheless, these conventional approaches suffer from the following shortcomings: (i) The Mix model only models commonality whilst neglecting the distinction among domains. (ii) In Pre-train&Fine-tune, the first drawback is non-trivial model parameters and inefficient fine-tuning. For instance, the pre-train and fine-tune paradigm requires maintaining a domain-specific model for each domain, incurring high storage and computation cost. Additionally, fine-tuning brings about the catastrophic forgetting issue. To be specific, when fine-tuning the pre-trained model, some of the common knowledge acquired from multiple domains will be forgotten. (iii) The deployability of the MTL framework is inadequate. On the one hand, its structure indicates poor compatibility. For example, the star topology based on fully connected networks in STAR [30] may hinder its combination with existing recommendation models such as Factorization Machines (FM) [27]. On the other hand, joint learning becomes infeasible if there exist a large number of domains.

To address the above-mentioned challenges in MSR, we propose a novel prompt-enhanced paradigm, which is inspired by the recent advances in pre-training and prompt tuning [22]. However, since the tokens in recommendation are mostly discrete features (*e.g.*, user ID, item ID, and domain ID), which lack meaningful semantics as in NLP tasks [22, 41], it is challenging to design NLP-style hard prompts in MSR setting. To this end, we propose a novel **P**rompt **L**earning **A**nd **T**uning **E**nhancement (PLATE) paradigm with two types of soft prompts, namely domain prompt and user prompt. The former aims at extracting domain distinctions, and the latter focuses on conducting more accurate personalized recommendations across domains. Notably, we develop two parameter-efficient architectures for user prompt generation, which significantly reduce the amount of soft prompt parameters caused by a large number of users. To capture the commonality and difference among domains, we further propose a two-stage pre-training and prompt tuning framework. To be specific, we first pre-train a unified DRS backbone model based on the data from all the domains, which is able to acquire the global cross-domain commonality. Then, prompt tuning is employed with domain prompt and user prompt modules so as to capture the differences among various domains and users.

As a consequence, the implementation of two novel prompt designs and two-stage optimization has the following merits. First, since most model parameters are frozen after the pre-training stage, the cross-domain common knowledge is retained in prompt tuning stage, which significantly relieves the catastrophic forgetting issue. Second, only 6% of trainable model parameters are updated for each domain in the prompt tuning stage, leading to high tuning and storage efficiency. Third, the proposed PLATE paradigm possesses great compatibility, as both the prompt modules and the two-stage framework are applicable to the vast majority of mainstream DRS models. We validate the effectiveness of PLATE against the state-of-the-art MSR baselines on three benchmark datasets from Douban, Amazon, and Ali-CCP.

The contributions of this work can be summarized as follows:

- We propose a novel multi-scenario recommendation paradigm, PLATE, which applies prompt tuning to multi-scenario recommendation tasks with two novel prompt designs, namely domain prompt and user prompt. To the best of our knowledge, we are the first to introduce prompt tuning techniques into MSR;
- Most importantly, the domain prompt and user prompt provide PLATE great compatibility with various DRS models, tackling their low flexibility and inadequate fine-tune efficiency when dealing with multi-scenario recommendation. Besides, the domain prompt and user prompt enable PLATE to conduct domain-aware parameter update and more personalized recommendation;
- Extensive experiments show that PLATE outperforms the state-of-the-art baselines based on various DRS backbone models with only 6% trainable parameters in the tuning stage.

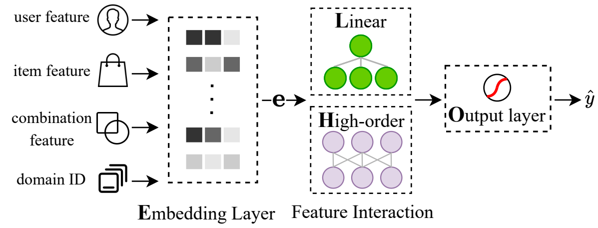


Figure 1: Typical DRS architecture.

2 PRELIMINARY

In this section, we first define the problem formulation of multi-scenario CTR prediction. Then, the typical architecture of deep recommender system (DRS) model is illustrated.

2.1 Multi-Scenario CTR Prediction

In this paper, we focus on the Click-Trough Rate (CTR) prediction task in multi-scenario recommendation setting with the following formulation. The recommender system generally takes the following data $(\boldsymbol{x}, d, y)$ as input, where $\boldsymbol{x}$ represents the concatenation of raw features, including user features, item attributes, contextual information, and combination features. d represents a domain indicator $d \in \{1, 2, \ldots, D\}$ to distinguish samples from D different domains. The ground truth label y indicates click ($y = 1$) or not ($y = 0$). Afterward, the raw features $\boldsymbol{x}$ are mapped into low-dimensional dense embedding vector $\boldsymbol{e}$ through an embedding layer. Finally, the prediction $\hat{y}$ of a user clicking on an item is calculated via $\hat{y} = f_d(\boldsymbol{e})$, where f_d denotes the recommendation model like DeepFM [13] in the d-th domain.

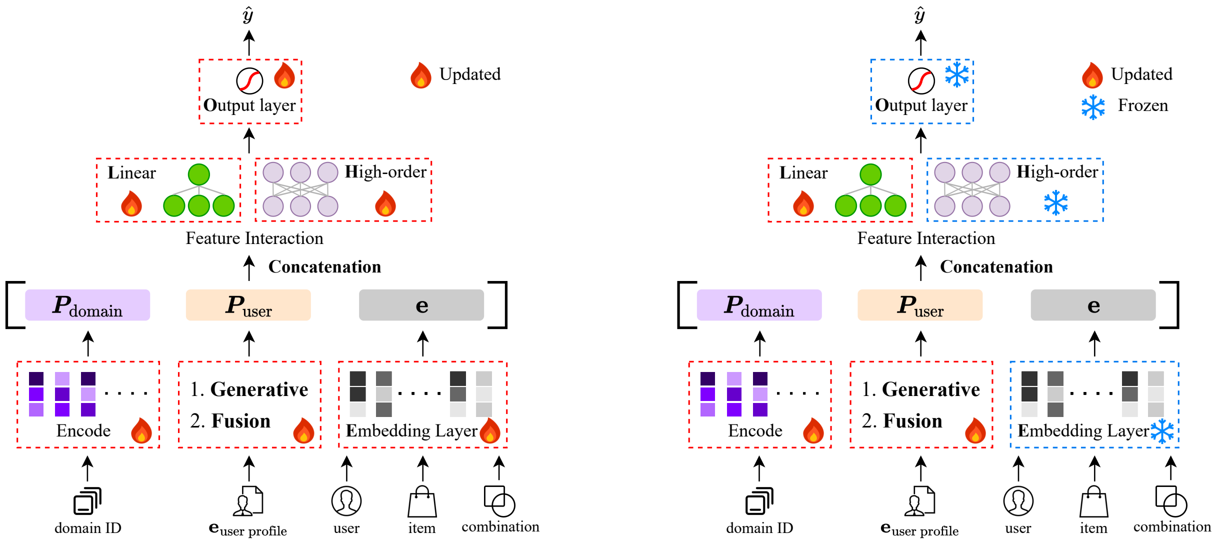
2.2 Typical Architecture of DRS Models

The common architecture of deep recommender system (DRS) mainly consists of three components: embedding layer, feature interaction, and output layer as shown in Figure 1. We denote $\boldsymbol{E}$, $\boldsymbol{L}$, $\boldsymbol{H}$, and $\boldsymbol{O}$ as the parameters of the embedding layer, first-order feature interaction (linear), high-order feature interaction, and output layer, respectively [34, 43].

2.2.1 ***Embedding Layer.*** The raw input features usually consist of both categorical and numerical ones as different feature fields. Suppose there are M categorical and N numerical features, and it can be represented as:

$$\boldsymbol{x} = [\underbrace{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M}_{\text{one-hot encoding}}; \underbrace{x_1, x_2, \ldots, x_N}_{\text{scalars}}], \tag{1}$$

where we denote $\boldsymbol{x}_m$ as the one-hot vector of the m-th categorical field and x_n as the scalar of the n-th numerical field. $\boldsymbol{x}_m$ is usually transformed into a low-dimensional embedding by a look-up operation $\boldsymbol{e}_m = \boldsymbol{E}_m \cdot \boldsymbol{x}_m$, where $\boldsymbol{E}_m \in \mathbb{R}^{u_m \times k}$ is the weight matrix of m-th categorical field, u_m is the number of unique feature values, and k is the pre-defined embedding dimension. For multi-valued



(a) Pre-training stage of PLATE (b) Prompt tuning stage of PLATE

Figure 2: Architecture of PLATE with domain and user prompt in a two-stage pre-training and prompt tuning optimization.

categorical features, we conduct mean pooling on embeddings of all non-zero feature values. For a numerical feature x_n, we obtain its feature embedding $\boldsymbol{e}_n$ following the mainstream representation approaches including no embedding, field-embedding, and discretization [12]. Therefore, the final output of the embedding layer is the concatenation of embeddings of all feature fields where $\|$ denotes the concatenation:

$$\boldsymbol{e} = [\boldsymbol{e}_1 \| \boldsymbol{e}_2 \| \dots \| \boldsymbol{e}_{M+N}]. \tag{2}$$

2.2.2 ***Feature Interaction***. The embedding layer is usually followed by a feature interaction component to grasp both low-order and high-order interactions among different feature fields. For first-order interaction calculation, a generalized linear model like the wide component in Wide&Deep [6] is widely used [13, 21]. In addition, Factorization Machine (FM) [27] is usually implemented to specify second-order feature interaction as inner product of respective feature embeddings [13, 21]. Furthermore, feed-forward neural network [6, 13, 40], Cross Network (CN) [28], and Compressed Interaction Network (CIN) [21] are utilized to capture implicit and explicit high-order interactions. We denote the output of the feature interaction component as $\boldsymbol{h}$.

2.2.3 ***Output Layer***. The output layer of DRS model takes $\boldsymbol{h}$ as input and generates prediction $\hat{y}$:

$$\hat{y} = \sigma(\boldsymbol{W}_o \boldsymbol{h} + \boldsymbol{b}_o), \tag{3}$$

where $\boldsymbol{W}_o$ and $\boldsymbol{b}_o$ are the weight and bias for the output layer. Since our focus is CTR prediction task, the activation function σ is *Sigmoid* and the loss function is Logloss [3]:

$$\min_{\Theta} \mathcal{L} = -\frac{1}{B} \sum_{i=1}^{B} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i), \tag{4}$$

where y_i and $\hat{y}_i$ are the ground truth label and predicted value of the i-th sample, respectively. B is the total number of training samples and $\Theta = \{\boldsymbol{E}, \boldsymbol{L}, \boldsymbol{H}, \boldsymbol{O}\}$ is the set of all trainable parameters in the DRS model.

3 PROPOSED METHOD

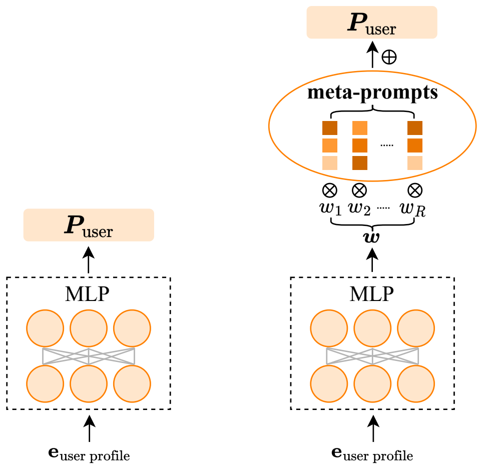
In this section, we first demonstrate the overall framework of PLATE paradigm based on the aforementioned DRS model. Afterward, the details of domain and user prompt modules and two-stage optimization are given. Finally, we discuss the characteristics of PLATE compared with previous MSR methods.

3.1 Overall Framework

To address the limitations of existing MSR methods, we propose PLATE paradigm for multi-scenario CTR prediction, which is compatible with most mainstream DRS models under the aforementioned framework in Section 2.2. Since it is difficult and inefficient to manually design hard prompt for this task, we introduce two parameter-efficient soft prompts namely *domain prompt* and *user prompt* into DRS model. Meanwhile, we introduce a two-stage optimization with pre-training and prompt tuning stages as shown in Figure 2. In pre-training stage, all parameters including two prompts are pre-trained on all-domain data. Afterward, in prompt tuning stage, parameters of prompt and linear feature interaction are fine-tuned on each domain. The choice of learnable parameters are further discussed as Section 4.4.

3.2 Domain Prompt and User Prompt

It is of great significance but challenging to explicitly capture domain distinction and different user preferences across domains in MSR. For one thing, contextual and user features are mostly treated equally as other common features, whose embeddings are simply concatenated and fed into feature interaction. As a result, the information from a unique user representation across domains is neglected. For another, the implicitly learnt domain distinction may be overwhelmed by unimportant features or information from the dominant domain. To address these challenges, we propose two parameter-efficient soft prompt modules, namely domain prompt and user prompt. To be specific, domain prompt aims at extracting domain heterogeneity, whilst user prompt aims at conducting personalized recommendation more accurately. Their combination acts



(a) Generative (b) Fusion

Figure 3: Generative and Fusion architecture for user prompt.

as a bridge connecting global commonality and domain distinction from different perspectives of user.

As for domain prompt, unlike multitudinous user and item features along with their massive combination features, it is common that the only contextual feature of domain is its ID, whose quantity is far less. Thus in PLATE, an explicit domain prompt vector $\boldsymbol{P}_{\text{domain}}$ is simply encoded from the domain indicator d as the unique representation of each doamin. It has the same embedding dimension k as other feature embeddings.

As for user prompt, employing informative user profile usually leads to better personalization [18]. However, solely concatenating embeddings of substantial user attributes to conduct user modeling will bring nontrivial new parameters, thus extremely inefficient. Hence, aiming at establishing a distinctive view of user, especially in a parameter-efficient manner, we propose two architectures namely **Generative** and **Fusion** to extract useful information from user profile and generate soft user prompt. Their detailed architecture are shown in Figure 3.

To be specific, the Generative architecture takes the concatenation of user feature embeddings $\boldsymbol{e}_{\text{user profile}} = [\boldsymbol{e}_{u_1} \| \dots \| \boldsymbol{e}_{u_z}]$ as input, where $\boldsymbol{e}_{u_j}$ is the embedding of the j-th user feature and z is the number of user features. Then it passes on $\boldsymbol{e}_{\text{user profile}}$ to an MLP to automatically learn different user patterns, and directly outputs the distinctive user prompt with dimension k. In contrast, the Fusion architecture also takes $\boldsymbol{e}_{\text{user profile}}$ as input followed by an MLP, but it outputs: (i) a prompt pool with R meta-prompts $\boldsymbol{MP}_r$ and (ii) the weighted values $\boldsymbol{w} = [w_1, \dots, w_r, \dots, w_R]$ to fuse the meta-prompts by attentive pooling to obtain the user prompt. The learnt weight $\boldsymbol{w}$ is normalized by $Softmax$ function with temperature τ as a hyper-parameter.

$$\boldsymbol{P}_{\text{user}} = \sum_{r=1}^{R} w_r \cdot \boldsymbol{MP}_r, \tag{5}$$

$$w_r = \frac{e^{\frac{1}{\tau} w_r}}{\sum_{j=1}^{R} e^{\frac{1}{\tau} w_j}}, \tag{6}$$

Specifically, the individual meta-prompt possesses the capability of grasping the commonality among various user profiles. Furthermore, the pool of meta-prompts focuses on such commonality in different level and granularity, which is learnt by attentive weight $\boldsymbol{w}$ to generate user prompt from distinctive user perspectives [12].

After generating domain prompt and user prompt, they are concatenated as prefix to other feature embeddings as $[\boldsymbol{P}_{\text{domain}} \| \boldsymbol{P}_{\text{user}} \| \boldsymbol{e}]$ and fed into feature interaction component. By introducing these two parameter-efficient soft prompts, the information on domain heterogeneity and personalization context is explicitly injected into behavioral and item-level information.

3.3 Two-stage Optimization

Since it is challenging for the prompt-enhanced model to fully capture distinction across domains in one-step learning on all-domain data, whilst only based on single domain DRS backbone model, our proposed PLATE paradigm consists of two stages: (i) pre-training with prompt learning and (ii) prompt tuning. This two-stage optimization enables PLATE to learn global domain correlation in the former stage, then further capture distinction among domains in an explicit view of users in the latter one. Specifically, in pre-training stage, the parameters of prompt are synchronously updated, *i.e.,* prompt learning, with all other parameters in DRS model using all data from multiple domains. After that, in prompt tuning stage, prompts are further tuned to adapt to each domain. To sum up, domain prompt and user prompt are updated in both pre-training and prompt tuning stage, interacting with other features to update parameters in a domain-aware manner and thoroughly capture the commonality and heterogeneity across domains and users.

In contrast, for transformer-based large-sized pre-trained model in the community of Natural Language Processing and Computer Vision [4, 9], prompt is only introduced in tuning stage when the pre-trained model is applied to downstream tasks, thus being completely irrelevant to pre-training. Consequently, we also investigate another way to incorporate prompt, *i.e.,* freezing the prompt parameters in pre-training stage and fine-tuning them in tuning stage, named as **P**rompt **T**uning **E**nhancement (PTE) paradigm, as a variant of PLATE. To be specific, the loss function of PLATE is the same as Equation (4), where we define the trainable parameters Θ as:

$$\Theta = \begin{cases} \{\boldsymbol{P}, \boldsymbol{E}, \boldsymbol{L}, \boldsymbol{H}, \boldsymbol{O}\}, & \text{pre-training stage} \\ \{\boldsymbol{P}, \boldsymbol{L}\}, & \text{prompt tuning stage} \end{cases} \tag{7}$$

and the loss function of PTE also follows Equation (4), but the trainable parameters Θ is defined as:

$$\Theta = \begin{cases} \{\boldsymbol{E}, \boldsymbol{L}, \boldsymbol{H}, \boldsymbol{O}\}, & \text{pre-training stage} \\ \{\boldsymbol{P}, \boldsymbol{L}\}, & \text{prompt tuning stage} \end{cases} \tag{8}$$

From the Equation (7) and (8), we can observe that the only difference between PLATE and PTE is whether the prompts' parameters are frozen or tuned in pre-training stage. In comparison, previous models for MSR either train all the parameters in one stage or pre-train then fine-tune all parameters in two stages, which suffer from low computing and storage efficiency.

Choice of Learnable Parameters. Apart from tuning the proposed prompts, we also fine-tune the parameters (if applicable in a given DRS model) calculating the first-order feature interaction, *i.e.,* the linear part $\boldsymbol{L}$, because it probably possesses memorization capability and reflects the distinction of domain distribution [6]. Notably, in the context of prompt-tuning, it is a common practice

Algorithm 1: PLATE and PTE paradigm for multi-scenario CTR prediction

Input: combination of user and item features $\boldsymbol{x}$; domain indicator $d \in \{1, 2, \ldots, D\}$; true label of click y
Output: well trained CTR prediction model for all domains
Stage 1: Pre-training with Prompt Learning
1 **while** *not converge* **do**
2   Sample a mini-batch data instances from training data on all domains;
3   Calculate loss for PLATE via Equation (4) with $\Theta = \{\boldsymbol{P}, \boldsymbol{E}, \boldsymbol{L}, \boldsymbol{H}, \boldsymbol{O}\}$, or for PTE $\Theta = \{\boldsymbol{E}, \boldsymbol{L}, \boldsymbol{H}, \boldsymbol{O}\}$;
4   Take the gradient and update respective parameters;
5 **end**
Stage 2: Prompt Tuning
6 **for** $d = 1$ *to* D **do**
7   **while** *not converge* **do**
8   Sample a mini-batch data instances on domain d;
9   Calculate loss via Equation (4), where $\Theta = \{\boldsymbol{P}, \boldsymbol{L}\}$;
10   Take the gradient and update respective parameters;
11   **end**
12 **end**

to update a small number of additional parameters, such as the head in VPT [15]. Therefore, it is reasonable to classify PLATE (updating $\boldsymbol{P}$ with $\boldsymbol{L}$) as a member of the prompt-tuning field, rather than pre-training&fine-tuning. In Section 4.4, we will empirically discuss (i) the performance comparison between PLATE and PTE, and (ii) the choice of learnable parameters in prompt tuning stages, where we observe that updating $\boldsymbol{P}$ and $\boldsymbol{L}$ is the optimal solution.

Optimization Algorithm. Finally, the procedure of PLATE and PTE is summarized in Algorithm 1. Specifically, in the first stage of pre-training, a mini-batch of training data is sampled from all-domain data (line 2); then we obtain loss via Equation (4) (line 3); next, the gradient is taken and PLATE updates all parameters while PTE does not update $\boldsymbol{P}$ (line 4). In the second stage of prompt-tuning, for each domain d the mini-batch is sampled from its own data (line 8); afterward, only $\boldsymbol{P}$ and $\boldsymbol{L}$ are updated (line 9-10) by calculating loss via Equation (4) for both PLATE and PTE.

3.4 Discussion

First, we compare PLATE paradigm with conventional MSR models and discuss their characteristics.

PLATE v.s. Mix. Both of them commonly train on all-domain data. Mix fail to take domain distinction into consideration, while PLATE explicitly models domain heterogeneity with the updated prompt in prompt tuning stage.

PLATE v.s. Pre-train&Fine-tune. Pre-train&Fine-tune is adapted to each domain like PLATE, but it requires fine-tuning all parameters resulting in low efficiency. In comparison, PLATE only tunes a tiny portion (which will be detailed in Section 4.6) with the two parameter-efficient soft prompts, which leads to far lower computation and model storage cost. In addition, Pre-train&Fine-tune is unable to fix the common knowledge learnt from pre-training, thus bringing about catastrophic forgetting. By contrast, in PLATE, the global commonality is frozen in the majority of parameters and domain heterogeneity is further captured in prompt tuning stage.

PLATE v.s. MTL. Both of them maintain shared and domain-specific parameters to model domain commonality and distinction. However, MTL usually adopts inflexible structure [30], where joint learning becomes infeasible for a large number of domains. By contrast, PLATE employs a flexible prompt-enhanced manner possessing great compatibility with mainstream DRS models.

Theoretically, PLATE does not explicitly take the difference of high-order feature interactions across domains into prompt tuning stage, because we argue that: first, in prompt tuning stage, only considering difference across domains through prompt and linear part in feature interaction has achieved state-of-the-art results from the observation of following experiments; second, it enables low cost of computation and model storage, since the domain-specific models need to fine-tune and store the shared majority parameters of the backbone model.

4 EXPERIMENTS

In this section, we conduct extensive experiments on public benchmark datasets to validate the effectiveness of our proposed PLATE paradigm and answer the following questions:

- **RQ1:** How does PLATE paradigm perform compared with the state-of-the-art baseline methods?
- **RQ2:** Is PLATE paradigm compatible and adaptive enough with different DRS models?
- **RQ3:** What is the optimal solution of model parameters to be fixed and updated in the pre-training and prompt tuning stage?
- **RQ4:** What are the effects of the proposed domain prompt and user prompt module?
- **RQ5:** Does PLATE paradigm possess high computation and storage efficiency?

4.1 Experimental Settings

4.1.1 ***Datasets.*** Our experiments are conducted on three datasets, namely Douban [44], Amazon 5-core [26], and Ali-CCP [25], and all these datasets have three domains. The dataset descriptions and statistics, as well as the splitting for training, validation, and test, are discussed in **Appendix A.1**.

4.1.2 ***Evaluation Metrics.*** As for evaluation metric, we apply Area Under the ROC (AUC) and Logloss to evaluate the performance of models on the test set. Specifically, a higher AUC value or a lower Logloss at "0.001" level indicates significant better recommendation performance [13], where the two-tailed unpaired t-test is performed.

4.1.3 ***Baselines.*** We compare PLATE with the following baselines: (i) ***Single*** trained only on each domain separately, (ii) ***Mix*** trained on all data from multiple domains, (iii) ***Pre-train&Fine-tune*** first pre-trained on all data then fine-tuned on each domain, and (iv) ***MTL*** including **Shared Bottom** [5], **OMoE** [14], **MMoE** [24], **AITM** [36], **PLE** [31], and **STAR** [30].

Notably, all existing prompt-based recommendation models (summarized in Section 5.2) are designed for sequential recommendations, which is a different recommendation task with ours. Thus, we exclude them in comparison. In addition, (i) - (iii) are applied to backbone model in different experiments, *e.g.*, DeepFM [13] in Section 4.2. Their detailed descriptions are given in **Appendix A.2**.

4.1.4 ***Implementation Details.*** The implementation details of PLATE are illustrated in **Appendix A.3**. The implementation code is available to ease reproducibility[1, 2].

Table 1: Overall performance comparison. The backbone model is DeepFM [13]. Suffix 'Fusion' and 'Gene' indicate the Fusion and Generative architecture generating user prompt. Boldface denotes the highest score and underline indicates the best result of the baselines. ⋆ represents significance level p-value < 0.05 of comparing PLATE over the best baselines.

Models / **AUC**	Douban			Amazon 5-core			Ali-CCP		
	Music	Book	Movie	Clothing	Beauty	Health	#1	#2	#3
Single	0.7666	0.7483	0.8181	0.6147	0.5989	0.6231	0.5873	0.5403	0.5854
Mix	0.7571	0.7373	0.8221	0.5949	0.6157	<u>0.6382</u>	<u>0.6205</u>	0.5945	<u>0.6150</u>
Pre-train&Fine-tune	0.7602	0.7381	<u>0.8223</u>	0.6259	<u>0.6251</u>	0.6365	0.6181	<u>0.5950</u>	0.6077
Shared Bottom	0.7718	0.7512	0.8191	0.6207	0.6073	0.6327	0.6155	0.5830	0.6075
MMoE	0.7777	0.7562	0.8217	0.6223	0.6023	0.6362	0.6175	0.5763	0.6123
OMoE	0.7763	0.7532	0.8218	0.6221	0.6088	0.6300	0.6153	0.5866	0.6110
AITM	0.7755	0.7540	0.8220	0.6215	0.6075	0.6337	0.6163	0.5635	0.6112
PLE	0.7780	0.7553	0.8220	0.6216	0.6089	0.6350	0.6184	0.5738	0.6101
STAR	<u>0.7808</u>	<u>0.7563</u>	0.8214	<u>0.6272</u>	0.6005	0.6341	0.6176	0.5817	0.6134
PLATE-Fusion (ours)	0.7918	0.7620	**0.8250**⋆	**0.6334**⋆	**0.6376**⋆	**0.6546**⋆	**0.6219**⋆	**0.5955**⋆	**0.6166**⋆
PLATE-Gene (ours)	**0.7920**⋆	**0.7629**⋆	0.8247	0.6327	0.6258	0.6507	0.6218	0.5927	0.6159

Models / **Logloss**	Douban			Amazon 5-core			Ali-CCP		
	Music	Book	Movie	Clothing	Beauty	Health	#1	#2	#3
Single	0.5049	0.5326	0.5008	0.7395	0.6516	0.5491	0.1710	0.2626	0.1620
Mix	0.5028	0.5262	0.4989	0.5342	0.4950	0.4381	0.1656	0.1802	0.1595
Pre-train&Fine-tune	0.4873	0.5189	<u>0.4984</u>	0.6126	0.5329	0.4968	0.1665	<u>0.1797</u>	0.1627
Shared Bottom	0.5069	0.5231	0.5085	<u>0.5163</u>	0.4913	0.4230	0.1663	0.1820	0.1597
MMoE	0.4792	0.5196	0.5012	0.5189	0.4951	**0.4221**⋆	0.1653	0.1886	0.1591
OMoE	0.4991	0.5217	0.4990	0.5172	<u>0.4822</u>	0.4273	0.1658	0.1814	0.1591
AITM	0.4805	0.5192	0.5028	0.5188	0.4877	0.4236	0.1654	0.2000	0.1596
PLE	0.4881	0.5228	0.5008	0.5195	0.4889	<u>0.4231</u>	<u>0.1652</u>	0.1922	<u>0.1590</u>
STAR	<u>0.4876</u>	<u>0.5212</u>	0.5065	0.5473	0.5017	0.4333	0.1677	0.1817	0.1625
PLATE-Fusion (ours)	**0.4762**⋆	0.5206	**0.4935**⋆	0.5041	**0.4521**⋆	0.4397	**0.1651**	0.1791	0.1594
PLATE-Gene (ours)	0.4772	**0.5182**⋆	0.4947	**0.5012**⋆	0.4556	0.4354	**0.1651**	**0.1782**⋆	**0.1588**

4.2 Comparison with Baseline Methods (RQ1)

The overall performance of PLATE and baselines are listed in Table 1. For all three datasets, we choose the widely used DeepFM [13] as the backbone model of PLATE, Single, Mix, and Pre-train&Fine-tune.

From Table 1, two shortcomings of previous MSR methods can be observed. First, pre-train and fine-tune may suffer from disastrous forgetting shown in the #1 and #3 domain in Ali-CCP. Second, in the six MTL frameworks, STAR achieves the best performance since it simultaneously models domain commonality and distinction by maintaining shared centered and domain-specific parameters. By contrast, other five models except Shared Bottom implicitly model domain relations (*e.g.,* adopting a gating network), while their performances reveal its inferiority.

In comparison, PLATE surpasses all state-of-the-art baselines on all three datasets and guarantees improvement on all domains except the Logloss in the Health domain on Amazon 5-core. It is able to address these limitations because: (i) the global information of commonality learned in pre-training stage is fixed and retained in prompt tuning stage; and (ii) the domain heterogeneity and personalization across domains are explicitly modeled, demonstrating the effectiveness of prompt modules and two-stage optimization. In addition, the designed Fusion structure illustrates its superiority over the Generative manner in most cases. This is probably because the pool of meta-prompts explicitly considers commonality in different level and granularity among users, thus generating more informative user prompt.

To summarize, PLATE achieves significantly better performance than the state-of-the-art baselines on all domains and datasets on DeepFM as backbone, demonstrating its effectiveness.

4.3 Compatibility with DRS Models (RQ2)

By design, our proposed PLATE is able to adapt to most DRS models, and enhance their capability of multi-domain modeling. Therefore, apart from DeepFM, we also demonstrate PLATE's compatibility with four other state-of-the-art backbone models, *i.e.,* xDeepFM [21], Wide&Deep [6], DCN [32], and FNN [40] on all three datasets.

To be specific, we compare the prompt enhanced model using two proposed user prompt generation methods with the pre-train and fine-tune paradigm. From Table 2, we can observe that: (i) with the proposed PLATE paradigm, the performance on all CTR prediction backbone models is improved. This demonstrates that PLATE has a remarkable ability to grasp domain difference while fixing common knowledge learned across domains with the introduced prompt modules and two-stage learning. (ii) For Douban dataset, the improvement on the Movie domain on all backbones is far less than that of two other domains, which may result from the large data volume of the Movie domain dominating the model.

In summary, PLATE possesses great compatibility and effectiveness on various DRS backbone models.

Table 2: The compatibility experiments on different backbones on Douban, Amazon 5-core, and Ali-CCP dataset

Backbone	Paradigm	Douban			Amazon 5-core			Ali-CCP		
		Music	Book	Movie	Clothing	Beauty	Health	#1	#2	#3
xDeepFM	Pre-train&Fine-tune	0.7717	0.7451	0.8246	0.6288	0.6151	0.6378	0.6202	0.5914	0.6161
	PLATE-Fusion (ours)	**0.7840**	**0.7585**	**0.8250**	**0.6314**	0.6194	0.6445	**0.6223**	**0.5963**	**0.6165**
	PLATE-Gene (ours)	0.7799	0.7513	0.8241	0.6288	**0.6264**	**0.6448**	0.6217	0.5949	0.6162
Wide&Deep	Pre-train&Fine-tune	0.7693	0.7469	0.8220	0.6261	0.6287	0.6394	0.6202	0.5901	0.6143
	PLATE-Fusion (ours)	0.7842	0.7572	**0.8232**	**0.6321**	**0.6308**	0.6497	**0.6225**	0.5957	**0.6161**
	PLATE-Gene (ours)	**0.7889**	**0.7598**	0.8230	0.6304	0.6306	**0.6498**	0.6220	**0.5990**	**0.6161**
DCN	Pre-train&Fine-tune	0.7711	0.7470	0.8270	0.6283	0.5850	0.6192	0.6212	0.5941	0.6168
	PLATE-Fusion (ours)	0.7722	0.7499	0.8262	**0.6323**	**0.6013**	0.6205	**0.6240**	**0.5986**	**0.6196**
	PLATE-Gene (ours)	**0.7748**	**0.7526**	**0.8271**	0.6301	0.6012	**0.6229**	0.6214	0.5970	0.6165
FNN	Pre-train&Fine-tune	0.7658	0.7443	0.8243	0.6287	0.6177	0.6385	0.6208	0.5940	0.6167
	PLATE-Fusion (ours)	**0.7702**	**0.7542**	0.8243	**0.6323**	**0.6185**	0.6404	**0.6221**	**0.5972**	**0.6169**
	PLATE-Gene (ours)	0.7691	0.7519	**0.8246**	0.6300	0.6117	**0.6408**	0.6217	0.5965	0.6164

Table 3: Comparison of PTE and PLATE on Douban, Amazon 5-core, and Ali-CCP datasets

Paradigm	Douban			Amazon 5-core			Ali-CCP		
	Music	Book	Movie	Clothing	Beauty	Health	#1	#2	#3
PTE-Fusion	0.7877	0.7588	0.8243	0.6304	0.6325	0.6533	0.6217	**0.5962**	0.6158
PTE-Gene	0.7879	0.7610	0.8243	0.6313	0.6297	0.6484	0.6211	0.5909	0.6161
PLATE-Fusion	0.7918	0.7620	**0.8250**	**0.6334**	**0.6376**	**0.6546**	**0.6219**	0.5955	**0.6166**
PLATE-Gene	**0.7920**	**0.7629**	0.8247	0.6327	0.6258	0.6507	0.6218	0.5927	0.6159

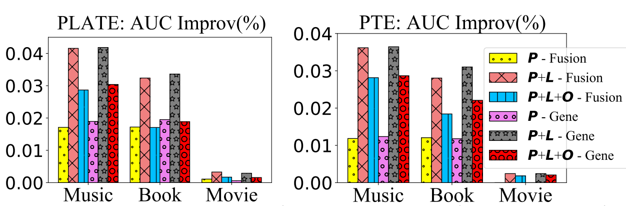


Figure 4: Component study on PLATE and PTE on Douban dataset, + means different combinations of fixed and tuned parameters. Suffix 'Fusion' and 'Gene' indicate the Fusion and Generative architecture generating user prompt. We use the ratio improvement of the AUC of the current model compared to the AUC of the baseline Finetune (DeepFM).

4.4 Component Analysis (RQ3)

We conduct numerous experiments to decide (i) whether to update or freeze prompt in pre-training stage (PLATE v.s. PTE), and (ii) what is the optimal combination of parameters fixed and tuned with prompt in tuning stage. Details are described as follows:

- PLATE (Prompt Learning And Tuning Enhancement): updating all parameters at pre-training stage.
- PTE (Prompt Tuning Enhancement): freezing the prompt parameters in the pre-training stage and only fine-tuning them in the prompt tuning stage.
- $\boldsymbol{P}$, $\boldsymbol{L}$, $\boldsymbol{O}$: exploring different combinations of fixed and tuned parameters where $\boldsymbol{P}$ denotes prompt, $\boldsymbol{L}$ denotes the linear part, and $\boldsymbol{O}$ denotes the output layer.

The results are shown in Table 3 and Figure 4. We can observe that:

First, refering to Table 3, PLATE is better than PTE in most cases only except on domain #2 in Ali-CCP. We attribute the superiority of PLATE to prompt learning in pre-training stage, which better extracts the commonality and distinction of users and items across different domains by learning the introduced domain prompt and user prompt. In contrast, freezing the prompt in pre-training stage may hinder the capability to do so in a domain-aware manner.

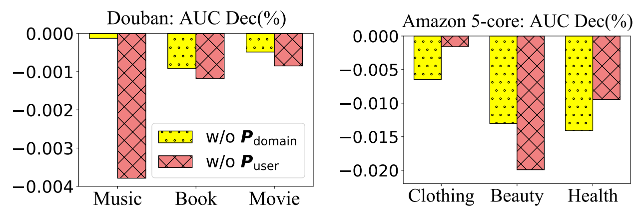


Figure 5: Ablation study on prompt module of PLATE on Douban and Amazon 5-core, w/o means removal of corresponding component. We use the ratio decrease of the AUC of the current model compared to the AUC of the full model.

Second, refering to Figure 4, in tuning stage, updating linear part with prompt parameters performs better than other combinations. A possible explanation is the distribution of linear feature interaction in each domain varies while the majority of common knowledge learned from multiple domains are fixed in other parts of model, like FM. Meanwhile, unlike the prompt tuning methods in other communities [15], it is better not to further fine-tune the output layer with prompt in our multi-scenario CTR prediction problem. We speculate the reason is that the output layer plays the role of generalization and balance on all domains, thus further updating it on each domain may dampen this ability.

4.5 Ablation Study (RQ4)

To verify the effectiveness of the prompt modules and answer **RQ4**, we conduct a series of ablation studies on the three datasets. We also take DeepFM as the backbone with user prompt generated by Fusion structure and the results are shown in Figure 5 for Douban and Amazon 5-core dataset.

We can observe that, both domain prompt and user prompt have significant contribution to our proposed PLATE paradigm only except for the #2 domain on Ali-CCP. In contrast, for PTE the same conclusion holds, except on Ali-CCP that is better to apply user prompt only. Since the domain prompt and user prompt are just concatenated as prefix to the embedding of other features, the combination and interaction of different prompt needs further investigation in the future.

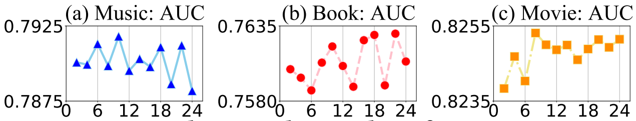


Figure 6: Analysis on the number of meta-prompts.

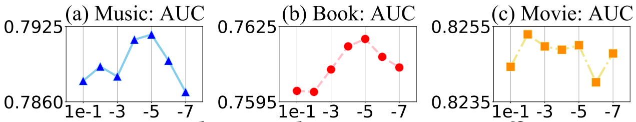


Figure 7: Analysis on the temperature coefficient.

4.6 Efficiency Analysis (RQ5)

For the number of parameters tuned, taking DeepFM as an example: the pre-train and fine-tune paradigm tunes 100% of parameters; in contrast, our proposed PLATE paradigm only needs to tune 6.36%(0.54%), 5.95%(0.10%), and 4.45%(0.17%) of parameters on Douban, Amazon 5-core, and Ali-CCP with better performances, where the number in parenthesis denotes the proportion of introduced prompt of all parameters. We can see that in the tuned parameters, the majority is the linear feature interaction component and the introduced prompt only accounts for a tiny proportion. Consequently, for different domains, a single model with different domain-specific parameters can be retained with ease. In contrast, multi-task framework needs to maintain multiple towers. Thus, it is even no longer applicable for a large number of domains.

On the other hand, PLATE also exhibits competitive or even superior training efficiency, especially when faced with numerous domains in commercial recommender systems. In practice, on the original Amazon[3] dataset with 24 domains and using DeepFM as backbone model, the computing consumption (i.e., GPU hour) of the following models is: Single (24), Pre-train&Fine-tune (9), Shared Bottom (6), MMoE (8), PLE (12), STAR (6), PLATE (6). We can observe that the computational cost of PLATE is the same with Shared Bottom (a simple model) and STAR (a commercial model), and is much less than other models.

To summarize, equipped with high parametric and computation efficiency, PLATE proved to possess the potential to be deployed into real-world commercial systems with numerous domains.

4.7 Hyper-parameter Analysis

Based on Douban dataset, we analyze the hyper-parameters of PLATE, including the number of meta-prompts and the temperature coefficient in Fusion structure generating user prompt. Specifically, the number of meta-prompts is varied from 2 to 24 and the temperature coefficient is ranged from 1e-1 to 1e-7. The results are shown in Figure 6 and 7. It can be observed that, as the number of meta-prompts increases to 10 the performance is improved, since more abundant information can be captured. However, the performance decreases as more meta-embeddings join in, which also brings about higher computation cost. Meanwhile, the temperature coefficient controls the distribution of weight averaging meta-prompts, thus it has a significant impact on capturing user distinction. We find the optimal global temperature is around 1e-5.

5 RELATED WORK

This section provides a brief summary of recent work on multi-scenario recommendation and prompt tuning for recommendation.

5.1 Multi-Scenario Recommendation

Multi-scenario recommendation [19, 33, 42] leverages data from all domains and simultaneously increases the recommendation accuracy in these multiple domains. STAR [30] proposes a star topology architecture with a single shared centered fully connected network multiplied by many domain-specific fully connected networks for multi-domain CTR prediction. SAR-Net [29] proposes two attention layers to capture users' cross-scenario interest transfer along with scenario-specific and scenario-shared experts. M2M [39] designs a Multi-level Meta Unit to explicitly learn domain-related meta-knowledge and utilizes meta mechanism to simultaneously model multi-scenario and multi-task.

Facing the shortcomings of the above methods introduced in Section 1, i.e., ignoring domain distinction, high computation and storage cost, disastrous forgetting, and poor deployability, our proposed prompt learning and prompt tuning enhancement paradigm exploits domain prompt and user prompt in two-stage optimization. Not only can it explicitly capture domain commonality and heterogeneity in a more personalized and domain-aware manner, but it also leads to high computation and storage efficiency with SOTA performance and great compatibility with various DRS models.

5.2 Prompt Tuning with Recommendation

Facing the difficulty of fine-tuning large-sized transformer-based pre-trained model on downstream tasks, the concept of prompt tuning is first raised in [22]. Generally, prompt is a piece of text prepended to original input to guide the downstream tasks to adapt to the pre-trained model. Prompt tuning focuses on learnable embeddings known as soft/continuous prompt, which are directly optimized via gradients during fine-tuning [17, 23]. For example, VPT [15] uses a fixed pre-trained ViT backbone with learnable task-specific prompts into the input space and its performance surpasses the fine-tuning protocol.

Several works try to incorporate prompt with recommendation [37]. One of the solutions is to naturally resort to successful large Pre-trained Language Model (PLM). PEPLER [18] employs item features as discrete prompt and ID embeddings as continuous prompt to generate recommendation explanation. P5 [11] designs a collection of personalized hard prompts and integrates five recommendation task families into a unified conditional language generation framework. M6-Rec [8] also acts as a foundation model using prompt tuning with soft options and adapters. PPR [35] first pre-trains a transformer-based recommendation model, then conducts personalized prompt tuning to address cold-start recommendation.

However, all the aforementioned methods apply prompt tuning with transformer-based pre-trained models for sequential recommendations. In this paper, we are the first to investigate a new prompt tuning paradigm combining soft prompt with typical DRS architecture introduced in Section 2.2.

6 CONCLUSION

In this paper, we propose a Prompt Learning And Tuning Enhancement (PLATE) paradigm with newly introduced domain prompt and user prompt to conduct multi-scenario CTR prediction in a more personalization and domain-aware manner, which is compatible with various single domain DRS models. Extensive experiments on Douban, Amazon 5-core, and Ali-CPP datasets show that our method surpasses state of art baselines with low computational and

[3] https://jmcauley.ucsd.edu/data/amazon/

Table 4: The statistics of Douban, Amazon 5-core, and Ali-CCP

Dataset	Douban			Amazon 5-core			Ali-CCP-Train			Ali-CCP-Test		
Domains	Music	Book	Movie	Clothing	Beauty	Health	#1	#2	#3	#1	#2	#3
Users	1672	2110	2718	39387	22363	38609	80704	1986	136406	47400	1156	73924
Items	5567	6777	9565	23033	12101	18534	297046	109259	297733	291213	103080	295281
Interactions	69709	96041	1133420	278677	198502	346355	15885371	318873	26095661	16351580	321024	26344010
Sparsity	25.43%	27.94%	39.09%	20.48%	22.28%	19.22%	96.00%	95.63%	96.18%	95.99%	95.61%	96.19%

memory cost. Our work also raise some intriguing problems to be solved in the future. For example, compared with transformer-based large-sized pre-trained models, in the proposed prompt-enhanced paradigm, the position of prompt with fixed length needs to be reserved before pre-training. Consequently, the design of prompt can be optimized according to the task and domain in advance.

A EXPERIMENTAL SETTINGS

A.1 Datasets

The statistics of three datasets are illustrated in Table 4.

- **Douban**[4]**.** This dataset is crawled from Douban and only users are overlapped in three domains. We randomly divide data into training set, validation set, and test set with the ratio of 8:1:1 on each domain with random seed 0. Only user ID, item ID, domain ID, and ratings are used. Specifically, rating is ranged from 1 to 5 and our goal is to predict whether a user gives a rating higher than 3 to an item.
- **Amazon 5-core**[5]**.** It is a dense subset from Amazon in which each user and item has at least 5 related records. We choose three related domains same as the task 2 in experiments conducted by HeroGRAPH [7]. There are both overlapped users and items in three domains. By convention, the time range of validation set is between 1st March 2014 and 30th April 2014, while the records before 1st March 2014 and after 30th April 2014 are counted as training set and test set, respectively. Only user ID, item ID, domain ID, and ratings are used. Similarly, records with rating greater than 3 are considered as positive samples.
- **Ali-CCP**[6]**.** It is collected from the real traffic logs from Taobao's recommendation system. The original dataset already consists of train and test set split by time. We consider the label of click as target and utilize 15 categorical features with 9 user features, 5 item features, and 1 context feature as domain ID.

A.2 Baselines Models

Apart from **Single**, **Mix**, and **Pre-train&Fine-tune**, we compare our proposed PLATE paradigm with the following MTL baselines:

- **Shared Bottom** is a multi-task model with shared parameters of bottom layers. In implementation, we follow the convention that the embedding layer is shared on which one specific fully-connected network is built for each domain.
- **OMoE** [14] is a multi-task model with a group of expert networks and a gating network ensembling the results from all experts.
- **MMoE** [24] is a multi-task model based on Shared Bottom. It has a group of bottom networks called expert instead of only one bottom network and learns a separate gating network to select a subset of experts for each task.
- **AITM** [36] is a multi-task model and it proposes a Adaptive Information Transfer (AIT) module to learn what and how much information to transfer with sequential dependence.
- **PLE** [31] is a multi-task model which separates the task-shared experts and the task-specific experts and explicitly employing a progressive routing mechanism.
- **STAR** [30] is a multi-domain model with star topology fully-connected network, which consists of shared centered and domain specific factorized networks.

For multi-scenario recommendation, it has become a trend to build a unified multi-task framework and treat different domains as different tasks to learn their commonalities and correlations [42].

Table 5: The hyper-parameter settings in our experiments

Dataset	Douban	Amazon 5-core	Ali-CCP
Embedding dimension (k)	16	16	20
Batch size	512	2048	6000
Hidden layers of DRS	[16,16]	[64,64]	[256,128,64]
Hidden layers for $\boldsymbol{P}_{\text{user}}$	[32,32]	[32,32]	[128,64]
Number of meta-prompts (R)	10	10	20
Learning rate of train	2e-3	1e-2	5e-4
Learning rate of test	5e-3/5e-4	5e-3	5e-4
$L2$ regularization	1e-5	1e-5	1e-5
Dropout	0.2	0.2	0.2

A.3 Implementation Details

All models are trained with Adam [16] optimizer to minimize the binary cross entropy loss and the hyper-parameters are listed in Table 5. In the table, Hidden layers represent the structure of hidden layers of deep network in recommendation model, while hidden layers in prompt represent the structure of network generating user prompt. Meanwhile, ReLU is used as activation function except in output layer. In addition, the test learning rate in Douban experiment is 5e-3 for the first two domains and 5e-4 for the last domain of movie. For the prompt generation, it is notable that only the embedding of user ID is used for Douban and Amazon 5-core while the embeddings of 9 user features are applied for Ali-CCP. Meanwhile, the domain prompt is randomly initialized by $xavier\_uniform$ while user prompt is initialized by zero. Finally, all the reported results are averaged over 3 runs.

ACKNOWLEDGEMENT

This research was supported by Huawei (Huawei Innovation Research Program). We thank MindSpore [1] for the partial support of this work, which is a new deep learning computing framework. It was also partially supported by APRC - CityU New Research Initiatives (No.9610565, Start-up Grant for New Faculty of City University of Hong Kong), SIRG - CityU Strategic Interdisciplinary Research Grant (No.7020046, No.7020074), CityU - HKIDS Early Career Research Grant (No.9360163).

[4]https://github.com/FengZhu-Joey/GA-DTCDR/tree/main/Data
[5]http://jmcauley.ucsd.edu/data/amazon/
[6]https://tianchi.aliyun.com/dataset/dataDetail?dataId=408

REFERENCES

[1] 2020. MindSpore. https://www.mindspore.cn/

[2] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-domain mediation in collaborative filtering. In *International Conference on User Modeling*. Springer, 355–359.

[3] Joseph Berkson. 1944. Application of the logistic function to bio-assay. *Journal of the American statistical association* 39, 227 (1944), 357–365.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[5] R Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias1. In *Proceedings of the Tenth International Conference on Machine Learning*. Citeseer, 41–48.

[6] 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.

[7] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation.. In *ORSUM@ RecSys*.

[8] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. *arXiv:2205.08084* (2022).

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

[10] Wenqi Fan, Tyler Derr, Xiangyu Zhao, Yao Ma, Hui Liu, Jianping Wang, Jiliang Tang, and Qing Li. 2021. Attacking Black-box Recommendations via Copying Cross-domain User Profiles. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1583–1594.

[11] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.

[12] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An embedding learning framework for numerical features in ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2910–2918.

[13] 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.

[14] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.

[15] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual Prompt Tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. 709–727.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980* (2014).

[17] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 3045—-3059.

[18] Lei Li, Yongfeng Zhang, and Li Chen. 2022. Personalized prompt learning for explainable recommendation. *arXiv:2202.07371* (2022).

[19] Pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang. 2020. 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*. 2605–2612.

[20] Xinhang Li, Zhaopeng Qiu, Xiangyu Zhao, Zihao Wang, Yong Zhang, Chunxiao Xing, and Xian Wu. 2022. Gromov-Wasserstein Guided Representation Learning for Cross-Domain Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1199–1208.

[21] 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.

[22] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[23] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv:2110.07602* (2021).

[24] 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.

[25] 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.

[26] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[27] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[28] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 255–262.

[29] Qijie Shen, Wanjie Tao, Jing Zhang, Hong Wen, Zulong Chen, and Quan Lu. 2021. SAR-Net: A Scenario-Aware Ranking Network for Personalized Fair Recommendation in Hundreds of Travel Scenarios. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4094–4103.

[30] Xiangrong 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.

[31] 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 *Fourteenth ACM Conference on Recommender Systems*. 269–278.

[32] 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.

[33] Yichao Wang, Huifeng Guo, Bo Chen, Weiwen Liu, Zhirong Liu, Qi Zhang, Zhicheng He, Hongkun Zheng, Weiwei Yao, Muyu Zhang, et al. 2022. CausalInt: Causal Inspired Intervention for Multi-Scenario Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4090–4099.

[34] Yejing Wang, Xiangyu Zhao, Tong Xu, and Xian Wu. 2022. Autofield: Automating feature selection in deep recommender systems. In *Proceedings of the ACM Web Conference 2022*. 1977–1986.

[35] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized Prompts for Sequential Recommendation. *arXiv:2205.09666* (2022).

[36] Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3745–3755.

[37] Chi Zhang, Rui Chen, Xiangyu Zhao, Qilong Han, and Li Li. 2023. Denoising and Prompt-Tuning for Multi-Behavior Recommendation. In *Proceedings of the Web Conference 2023*.

[38] Fan Zhang, Qiuying Peng, Yulin Wu, Zheng Pan, Rong Zeng, Da Lin, and Yue Qi. 2022. Multi-Graph based Multi-Scenario Recommendation in Large-scale Online Video Services. In *Proceedings of WWW 2022 Graph Learning workshop*.

[39] Qianqian Zhang, Xinru Liao, Quan Liu, Jian Xu, and Bo Zheng. 2022. Leaving No One Behind: A Multi-Scenario Multi-Task Meta Learning Approach for Advertiser Modeling. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1368–1376.

[40] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European conference on information retrieval*. Springer, 45–57.

[41] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language Models as Recommender Systems: Evaluations and Limitations. In *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*.

[42] Yuanliang Zhang, Xiaofeng Wang, Jinxin Hu, Ke Gao, Chenyi Lei, and Fei Fang. 2022. Scenario-Adaptive and Self-Supervised Model for Multi-Scenario Personalized Recommendation. In *Proceedings of CIKM 2022*.

[43] Xiangyu Zhao. 2022. Adaptive and automated deep recommender systems. *ACM SIGWEB Newsletter* Spring (2022), 1–4.

[44] 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.

[45] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*.