SAME: Scenario Adaptive Mixture-of-Experts for Promotion-Aware Click-Through Rate Prediction

Xiaofeng Pan*[1], Yibin Shen*[1], Jing Zhang[2], Keren Yu[1], Hong Wen[1], Shui Liu[1]
Chengjun Mao[1] and Bo Cao[1]
[1]Alibaba Group, [2]The University of Sydney
pxfvintage@163.com, {shilou.syb, keren.ykr, qinggan.wh, shui.lius, chengjun.mcj, zhizhao.cb}@alibaba-inc.com
jing.zhang1@sydney.edu.au

***Abstract*—Promotions are becoming more important and prevalent in e-commerce platforms to attract customers and boost sales. However, Click-Through Rate (CTR) prediction methods in recommender systems are not able to handle such circumstances well since: 1) they can't generalize well to serving because the online data distribution is uncertain due to the potentially upcoming promotions; 2) without paying enough attention to scenario signals, they are incapable of learning different feature representation patterns which coexist in each scenario. In this work, we propose Scenario Adaptive Mixture-of-Experts (SAME), a simple yet effective model that serves both promotion and normal scenarios. Technically, it follows the idea of Mixture-of-Experts by adopting multiple experts to learn feature representations, which are modulated by a Feature Gated Network (FGN) via an attention mechanism. To obtain high-quality representations, we design a Stacked Parallel Attention Unit (SPAU) to help each expert better handle user behavior sequence. To tackle the distribution uncertainty, a set of scenario signals are elaborately devised from a perspective of time series prediction and fed into the FGN, whose output is concatenated with feature representation from each expert to learn the attention. Accordingly, a mixture of the feature representations is obtained scenario-adaptively and used for the final CTR prediction. In this way, each expert can learn a discriminative representation pattern. To the best of our knowledge, this is the first study for promotion-aware CTR prediction. Experimental results on real-world datasets validate the superiority of SAME. Online A/B test also shows SAME achieves significant gains of 3.58% on CTR and 5.94% on IPV during promotion periods as well as 3.93% and 6.57% in normal days, respectively.**

***Index Terms*—Recommender System, Click-Through Rate Prediction, E-commerce Promotions, Scenario Adaptive, Mixture-of-Experts**

I. Introduction

A well-performing recommender system needs to discover valuable items from massive available options for customers not only accurately but also timely [12], [14], [25], [28], [35], [36], [42]. To achieve this goal, most of advanced Click-Through Rate (CTR) prediction models use high-order interactions of features to improve their representation capacity [3], [10], [21], [34], [40], and leverage sequential user behaviors to model users in a dynamic manner [8], [38], [39], [44], [45].

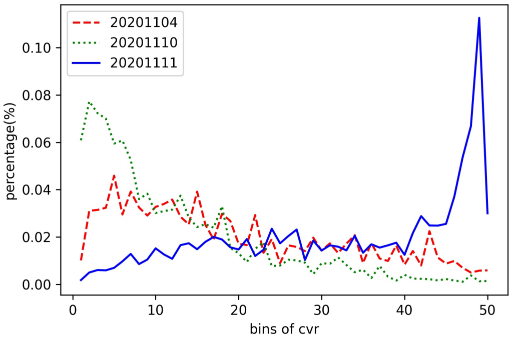


Fig. 1. The distribution of users' conversion rate (CVR) varies significantly in different scenarios.

Most of the previous works assume that users' preferences are coherent and change smoothly along time. However, there exists scenarios where users' behaviors can be driven differently by different occasions [33]. Such scenarios refer to time periods with different data distributions and are related to particular time or events. Especially with the intensification of e-commerce competition, various online promotions become quite frequent, which leads to significant distribution shift of online data as illustrated in Figure 1. Concretely, promotions can be grouped into three categories: 1) annual shopping festivals, such as Black Friday and Double 11; 2) promotions for Women's Day, Spring Festival, Christmas Day, and so forth, which are derived by certain culture; and 3) promotions initiated according to specific strategies of e-commerce platform or emerging consumption trends, such as Jingdong's Digital Festival and Pinduoduo's Zhenxiang Festival, which may be irregular in time. Impacts of promotions on e-commerce are partly investigated in [41] by a statistical analysis about Double 11. However, no suitable approach is proposed and promotions of the other categories draw little attention.

As far as we know, existing approaches to deal with recommendations for e-commerce promotions are mostly based on temporarily customizing or adding extra models. It highly relies on expert experience to decide when to switch models for online serving because it's uncertain that when online data distribution will change. Besides, different from Multi-Scenario CTR prediction [20], [26], it's not easy to build scenario-specific training data since scenarios in this work are time-variant and intertwined with each other in time, which

*Equal contribution.

makes medium and long term statistical features unavailable. As a result, it takes considerable efforts to manually handle every promotion in this way, while the improvements are uncertain because the influence of promotions varies according to many factors, such as the promotion intensity and the capacity of e-commerce platforms. For example, only super promotions like 618 and Double 11 could impact Taobao remarkably while almost every single promotion may have a significant impact on small e-commerce businesses, such as Tmall Global and Kaola.

Up to now, it remains a challenge to handle recommendations for both normal days and promotions using a unified model. Although there exist a few works [16], [31]–[33] paying attention to this challenge, they are incapable of modeling the aforementioned complex scenarios sufficiently. In [33], a model is developed to adapt to different scenarios by learning representations indexed by timestamp. However, timestamps are not enough to describe all promotions of the second and third category mentioned above because calendar in different cultures varies and promotions of the third category may not be time-dependent. In [16], item behaviors [7], $i.e.$, a set of users who interact with this item, are introduced via a carefully designed time-sensitive neural structure, which models items in a dynamic manner to strengthen the ability to predict users' emerging interests. Methods proposed in [31], [32] follow the similar idea of modeling items in a dynamic manner by adopting a hypergraph. However, the mutual interference between non-identically distributed data from different scenarios is not tackled, which may impose extra difficulties on the learning process. Last but not the least, existing sequential modeling methods may not perform well when serving small e-commerce businesses due to the highly sparse and incoherent user behaviors.

Based on the above analysis, it is non-trivial to develop a promotion-aware CTR prediction method for e-commerce recommendations. Inspired by works on time series prediction [5], [37] and multi-task learning [11], [17], [27], we propose a novel **S**cenario **A**daptive **M**ixture-of-**E**xperts (SAME) model. It explicitly models each sample from different scenarios as a mixture of experts, each of which could learn a discriminative feature representation pattern under the modulation of Feature Gated Network (FGN) and thus facilitates mitigating the mutual interference between non-identically distributed data. Our contribution is threefold:

- To the best of our knowledge, this is the first study of CTR prediction in the context of e-commerce promotions. We propose a simple yet effective SAME model to serve both promotion and normal scenarios.
- We address the above issues in a novel way by devising multiple experts modulated by a FGN. The FGN can efficiently adapt to different scenarios by elaborately processing scenario signals so that the distribution uncertainty is handled. Besides, a Stacked Parallel Attention Unit (SPAU) is designed and applied in each expert for effective sequential modeling.
- Experiments on real-world datasets and online tests demonstrate the superiority of our SAME model over representative methods. We also perform visualization analysis for better interpretability.

II. Related work

In this part, we briefly review the most related works to our SAME model, $i.e.$, CTR prediction and multi-task learning. We also discuss the existing methods for Multi-domain Learning to further clarify the specificity of promotion-aware CTR prediction.

A. CTR Prediction

CTR prediction has become a crucial part of many online applications, such as search engines, recommender systems, and online advertising. In recent years, Deep Neural Networks (DNN) have shown powerful capacity of learning feature representations and modeling high-order feature interactions. For CTR prediction, there are some representative DNN models have been proposed, including Wide&Deep [3], PNN [21], PIN [22], DeepFM [10], and DCN [34]. More recently, since the sequence of user behaviors contains rich information of users' interests and preferences, there is increasing attention on sequential modeling in online systems. Given a target item, DIN [45] introduces the attention mechanism to activate the historical behaviors and capture the diversity characteristic of user interests. ATRank [43] proposes an attention-based framework modeling the influence between heterogeneous behaviors of a user. To capture dependencies between sequential behaviors, DIEN [44] adopts a two-layer RNN structure to model the evolving process of specific interest for different target items. For better modeling of the temporal effects, HPMN [24] is proposed to capture the periodic patterns of users with a hierarchical recurrent memory network, while TIEN [16] models items in a dynamic manner by using item behaviors to strengthen the ability to predict users' emerging interests.

To sum up, learning higher-order representations of features and introducing sequential behaviors improve the expressive ability of models and the prediction accuracy significantly. However, none of these models pay enough attention to the issues caused by e-commerce promotions.

B. Multi-task Learning

Multi-task learning has been used successfully across various applications of machine learning, from natural language processing [2], [4] to computer vision [9], [23], and recommender systems [19]. By sharing representations between related tasks, multi-task learning allows to exploit common underlying factors and transfer knowledge across different tasks, where more data can be leveraged to learn better-shared representations. Multi-task learning is typically done in either a hard or a soft way by sharing parameters of hidden layers. The hard parameter sharing method [1] shares the hidden layers between all tasks and keeps several task-specific output layers. Such framework has been widely used in online advertising and recommender systems [18], [35], [36], which are based on the explicit decomposition of user sequential behavior graph from impression to purchase. For soft parameter sharing, each task has its own model with its own parameters, where the distance between the parameters is regularized [11], [17], [27]. With advanced techniques including expert model, gate mechanism, and connection routing, such models achieve parameter sharing between different experts while learning specific experts for different tasks.

Partially inspired by these prior works, we borrow the idea of Mixture-of-Experts. Nevertheless, we deal with a single task ($i.e.$, CTR prediction) in the context of time-variant scenarios, including promotion events and normal days.

C. Multi-domain Learning

Multi-domain learning [6], [13], [15] enables knowledge transfer between domains to improve learning. The difference between multi-domain learning and multi-task learning is that multi-domain learning makes predictions for multiple domains addressing the same problem. Multi-Scenario CTR prediction [20], [26] can be seen as a case of multi-domain learning, in which each scenario corresponds to a business domain and the task is the CTR prediction. Sheng $et\ al.$ [26] propose the Star Topology Adaptive Recommender (STAR) model to leverage data from all domains simultaneously and exploit the domain relationship so that the model can serve all domains. With shared centered parameters and multiple sets of domain-specific parameters, STAR can achieve effective information transformation across multiple domains to learn domain commonalities and distinctions. Niu $et\ al.$ [20] propose a tree-guided mixture of expert networks named TREEMS to accommodate all sharing scenarios under a single unified recommendation model, where each scenario obtains its specific context information from a scenario tree to determine the combination of experts. In general, Multi-Scenario CTR prediction tackles *spatially different* scenarios in different venues, where each scenario has its explicit indicators ($e.g.$, scenario id). Each scenario can be treated as stable and time-independent so data of each scenario are accessible for training.

In contrast, promotion-aware CTR prediction handles *time-variant* scenarios which are intertwined with each other in time, so it's uncertain that when online data distribution will change. Promotion-aware CTR models are expected to generalize well to the upcoming different scenarios automatically, whose data have never been seen during training, which makes existing Multi-Scenario CTR models not applicable. Besides, explicit scenario indicators are not available so it's not easy to build scenario-specific training data, which makes the design of scenario-specific experts [26] infeasible.

III. The Proposed Method

A. Problem Definition

In CTR prediction, the model takes input as $(\boldsymbol{x}, y) \sim (X, Y)$, where $\boldsymbol{x}$ is the feature and $y \in \{0, 1\}$ is the click label. Specifically, the input features of CTR models mainly consist of six parts. The first part is the user behavior sequence, which records user history of clicked/purchased items. The second part consists of the user features, including user profile ($e.g.$, age and gender) and statistic features from user history. The third part consists of the item features, $e.g.$, item id, category, brand, and related statistic features. The fourth part is the interaction features of the target item and user, $e.g.$, clicks of the user in the category during the last 24 hours, and purchases of the user in the shop during the last 24 hours. The fifth part consists of context features, such as position, device and time information. The sixth part of features contains scenario signals that are sensitive to promotions, including global statistics such as the amount of active users and the GMV (Gross Merchandise Volume) of several recent time windows, and the change rate of corresponding features.

The goal of CTR prediction is to learn a model f_θ with parameter θ that minimizes the the generalization error:

$$\theta^* = \underset{\theta}{min} E_{(\boldsymbol{x},y)\sim(X,Y)}[L(\boldsymbol{x}, y; f_\theta(\boldsymbol{x}))], \tag{1}$$

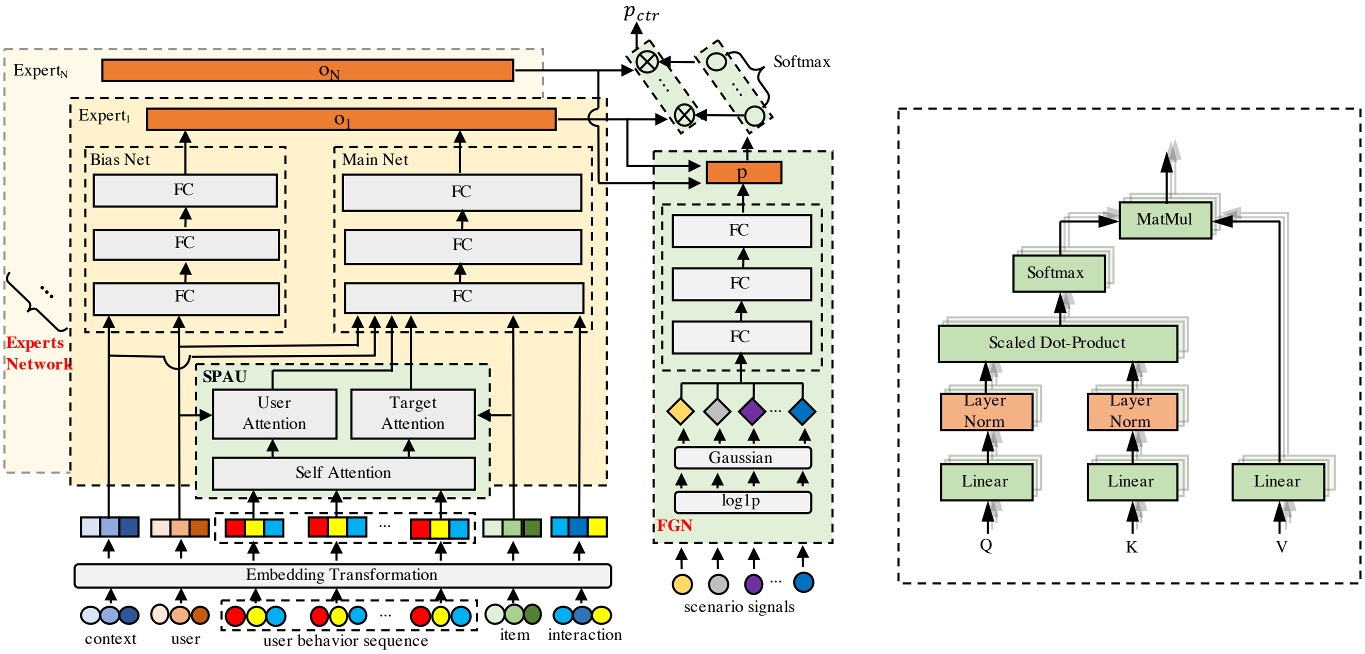
where L is the loss function. Note that the training datasets $\mathcal{D}$ consist of samples from both normal days and promotions, so in this work, the assumption of identical data distribution does not hold. Besides, taking the irregular promotions into consideration, the upcoming online data distribution is also uncertain.

Finally, we formulate the problem of promotion-aware CTR prediction as training a model on the datasets $\mathcal{D}$ which can predict the click probability given a user-item pair and generalize well to the upcoming different distribution automatically.

B. Motivation

The focus of this work is to develop a unified CTR prediction method serving both promotions and normal days. As mentioned in the introduction, promotion, a widespread phenomenon in e-commerce that can cause certain degrees of data distribution shift, has not been explored thoroughly in existing recommendation works. It can be observed that different interaction tendencies of users and items coexist in various scenarios, $e.g.$, normal days and promotions. For example, some users may tend to interact with items of their intrinsic preference while some may tend to interact with items globally or locally hot. Despite the coexistence, it's obvious that the impact of different interaction tendencies varies in different scenarios, which may be the main reason for the data distribution shift. We assume that different interaction tendencies lead to discriminative user-item feature representation patterns. Intuitively, to mitigate the mutual interference between non-identically distributed data and adapt to the potentially different online distribution, we need to learn different representation patterns and mix them according to the detailed context to make the final prediction.

Besides, we also need to consider that whether the existing sequential modeling methods can work well when it comes to recommendations for e-commerce businesses of which the scale is much smaller than platforms such as Taobao and Amazon. The more severe sparsity and incoherence in user behaviors may introduce new challenges on representation learning when handling the user behavior sequence.



(a) (b)

Fig. 2. (a) The SAME model, where the final prediction layer is omitted for simplicity. (b) Attention Structure of SPAU.

C. Overall Structure

In the following sections, we present the details of our proposed SAME model. As shown in Figure 2(a), scenario signals, a set of features elaborately devised from a perspective of time series prediction, are fed into the FGN, while the other features are fed into each expert via a shared embedding layer. For each expert, the SPAU is used for generating a representation of the target item and user. The FGN aims at generating a high-quality scenario representation to capture shifted data distribution so that different scenarios can be distinguished automatically. With the output of the experts network and the FGN, attention weights are calculated to combine different experts for the final CTR prediction. With scenario signals carefully processed, we try to make the FGN more efficient in learning scenario representation than the gate mechanism of multi-task learning, and provide better guidance of representation learning for experts.

D. Shared Embedding Layer

The shared embedding layer is devised to handle the input of experts network, which includes all features mentioned in Section III-A except for scenario signals. They can be further grouped into two kinds of features: categorical feature and numerical feature. We discrete the numerical features based on their boundary values, transforming them into the categorical type. Then each categorical feature is encoded as a one-hot vector. After raw feature processing, $\boldsymbol{x}^u$, $\boldsymbol{x}^{seq}$, $\boldsymbol{x}^i$, $\boldsymbol{x}^{ui}$ and $\boldsymbol{x}^c$ denote user features, user behavior sequence, item features, user-item interaction features and context features respectively. Due to the sparseness nature of one-hot encoding, we apply linear fully connected layers to obtain low dimensional embedding $\boldsymbol{e}^u$, $\boldsymbol{e}^i$, $\boldsymbol{e}^{ui}$ and $\boldsymbol{e}^c$, according to:

$$\begin{aligned}
\boldsymbol{e}^u &= \left[\boldsymbol{W}_1^u \boldsymbol{x}_1^u, ..., \boldsymbol{W}_k^u \boldsymbol{x}_k^u, ...\right], \boldsymbol{W}_k^u \in \boldsymbol{R}^{d_k^u \times v_k^u}, \\
\boldsymbol{e}^i &= \left[\boldsymbol{W}_1^i \boldsymbol{x}_1^i, ..., \boldsymbol{W}_k^i \boldsymbol{x}_k^i, ...\right], \boldsymbol{W}_k^i \in \boldsymbol{R}^{d_k^i \times v_k^i}, \\
\boldsymbol{e}^{ui} &= \left[\boldsymbol{W}_1^{ui} \boldsymbol{x}_1^{ui}, ..., \boldsymbol{W}_k^{ui} \boldsymbol{x}_k^{ui}, ...\right], \boldsymbol{W}_k^{ui} \in \boldsymbol{R}^{d_k^{ui} \times v_k^{ui}}, \\
\boldsymbol{e}^c &= \left[\boldsymbol{W}_1^c \boldsymbol{x}_1^c, ..., \boldsymbol{W}_k^c \boldsymbol{x}_k^c, ...\right], \boldsymbol{W}_k^c \in \boldsymbol{R}^{d_k^c \times v_k^c},
\end{aligned} \tag{2}$$

where $\boldsymbol{W}_k^* \in \boldsymbol{R}^{d_k^* \times v_k^*}$ denotes the embedding matrix of the k^{th} feature, *i.e.*, $\boldsymbol{x}_k^*$, and d_k^* is the embedding dimension of $\boldsymbol{x}_k^*$ while v_k^* is the vocabulary size, and $* \in \{u, i, ui, c\}$. Then the embedding of user behavior sequence is formed with item embedding in the sequence, *i.e.*, $\boldsymbol{e}^{seq} = \{\boldsymbol{e}_1^i, ..., \boldsymbol{e}_t^i\}$ where $\boldsymbol{e}_t^i$ denotes the item embedding of t^{th} user behavior and t is the sequence length.

E. Experts Network

The basic principle of designing the experts network is to make every single expert be able to generate a high-quality representation from the output of the shared embedding layer, *i.e.*, $\boldsymbol{e}^u$, $\boldsymbol{e}^{seq}$, $\boldsymbol{e}^i$, $\boldsymbol{e}^{ui}$ and $\boldsymbol{e}^c$.

The embedding of user behavior sequence is of great importance since it contains rich information about user interest. When online shopping, users may browse some unrelated items, which would somehow influence the representation learning of the sequence. Besides, how active the users are in e-commerce depends a lot on the scale of business and the abundance of items, which means user behaviors tend to be more sparse and incoherent if the scale of business is smaller. Under such circumstances, the existing sequential modeling methods, such as target attention mechanism [45] and RNN-based network [44], may not be able to learn well and thus lead to unreliable personalized recommendations.

Due to the above reason, we propose the SPAU structure to handle the user behavior sequence, in which three kinds of attention weights are calculated. For all of them, we apply **L**ayer **N**ormalization on ***Query*** and ***Key*** (LN-QK) vectors as shown in Figure 2(b), which can relieve the vanishing gradient problem so better attention weights can be learned. Layer normalization is not applied on *Value* for the reason that we don't expect to change its original distribution. In this way, the difficulty of learning from the sparse and incoherent sequence is alleviated and the training performance is boosted. First, we use a multi-head [30] self-attention network to model user preference from multiple views of interest and decrease the influence of unrelated and incoherent behaviors, according to:

$$\begin{aligned}\boldsymbol{output} &= Concat(\boldsymbol{head}_1, ..., \boldsymbol{head}_h)\boldsymbol{W}^O,\\ \boldsymbol{head}_i &= Attention(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i)\\ &= softmax(\frac{LN(\boldsymbol{Q}_i)\cdot LN(\boldsymbol{K}_i^T)}{\sqrt{d_k}})\boldsymbol{V}_i,\end{aligned} \tag{3}$$

where h represent the number of heads and $\boldsymbol{W}^O$ denotes the weight matrix of output linear transformation. $LN(.)$ denotes layer normalization operation and d_k is the dimension of $\boldsymbol{K}_i$. $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$ and $\boldsymbol{V}_i$ are calculated by linear projection:

$$\boldsymbol{Q}_i = \boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}_i = \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}_i = \boldsymbol{V}\boldsymbol{W}_i^V, \tag{4}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ refer to *Query*, *Key* and *Value* respectively, and $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$ and $\boldsymbol{W}_i^V$ denote linear projection weight matrix for $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$ and $\boldsymbol{V}_i$ of the i^{th} head respectively. For self-attention, $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ all refer to $\boldsymbol{e}^{seq}$, and $\hat{\boldsymbol{e}}^{seq} = \{\hat{e}_1^i, ..., \hat{e}_t^i\}$ is the $\boldsymbol{output}$ in which $\hat{e}_t^i$ denotes the output of multi-head self-attention at t^{th} position. Then, on top of the multi-head self-attention, user attention and target attention are performed in parallel. To mine more fine-grained personalized information and suppress noisy behavior, $\boldsymbol{e}^u$ is used as the query vector attending to $\hat{\boldsymbol{e}}^{seq}$. In Equation 4, $\boldsymbol{e}^u$ is used as $\boldsymbol{Q}$ and $\hat{\boldsymbol{e}}^{seq}$ is used as $\boldsymbol{K}$ and $\boldsymbol{V}$, and then the user attention $\boldsymbol{s}^u$ can be calculated according to Equation 3. Similarly, we perform target attention to activate historical interests related to the target item, with $\boldsymbol{e}^i$ attending to $\hat{\boldsymbol{e}}^{seq}$. By replacing $\boldsymbol{e}^u$ with $\boldsymbol{e}^i$, target attention $\boldsymbol{s}^i$ is calculated in the same way as user attention.

It can be observed that different users in different context usually behave differently even to similar items, which implies that users' behaviors are biased. Therefore, we concatenate $\boldsymbol{e}^u$ and $\boldsymbol{e}^c$, and feed them into a Multi-Layer Perception (*MLP*), which is called *Bias Net*. Meanwhile, all information, including $\boldsymbol{s}^u$, $\boldsymbol{s}^i$, $\boldsymbol{e}^i$, $\boldsymbol{e}^u$, $\boldsymbol{e}_{ui}$ and $\boldsymbol{e}^c$, are concatenated and fed into another *MLP*, $i.e.$, *Main Net*. Finally, we concatenate the outputs of *Main Net* and *Bias Net* to obtain the output of an expert, $i.e.$, the user-item feature representation.

F. Feature Gated Network (FGN)

Scenario signals, described in Section III-A, are devised following the idea of trend features in time series prediction. As the input of the FGN, it should be noted that the number of unique feature values of most scenario signals relies on the amount of related time windows in training data. For example, given training data sampled from 30 days, there are only 30 unique feature values for the feature of the amount of active users in the last day. In this situation, direct usage or discretization and embedding of these features may lead to poor online performance because of the over-fitting problem and unstable distribution of feature value. To address this issue, we adopt the Gaussian log1p transformation [46] to process scenario signals, which helps keep sensitive to feature values while improving the stability of feature transformation. Specifically, for scenario signals $\boldsymbol{x}^s$, this process can be formulated as:

$$\begin{aligned}\hat{\boldsymbol{x}}^s[i] &= log(1+\boldsymbol{x}^s[i]),\\ \hat{\boldsymbol{\mu}}^s &= \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\boldsymbol{x}^{\hat{s}}[i],\\ \hat{\boldsymbol{\sigma}}^s &= \sqrt{\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}(\hat{\boldsymbol{x}}^s[i]-\hat{\boldsymbol{\mu}}^s)^2},\\ \boldsymbol{z}^s[i] &= \frac{\hat{\boldsymbol{x}}^s[i]-\hat{\boldsymbol{\mu}}^s}{\hat{\boldsymbol{\sigma}}^s},\end{aligned} \tag{5}$$

where $\mathcal{D}$ denotes training set and $|\mathcal{D}|$ denote the number of samples in $\mathcal{D}$. $\boldsymbol{x}^s[i]$ denotes scenario signals of the i^{th} sample in $\mathcal{D}$ and $\boldsymbol{z}^s[i]$ is the corresponding output of Gaussian log1p transformation.

As shown in the right part of Figure 2(a), $\boldsymbol{z}^s$ is fed into a *MLP* to get the higher-order scenario representation $\boldsymbol{p}$. Then, the gated (attention) vector $\boldsymbol{\alpha}$ is derived according to:

$$\alpha_i = \frac{exp(f(\boldsymbol{p}, \boldsymbol{o}_i))}{\sum_{j=1}^{N} exp(f(\boldsymbol{p}, \boldsymbol{o}_j))}, \tag{6}$$

where α_i is the gated weight for i^{th} expert and f is a function that projects input into a scalar. N denotes the number of experts and $\boldsymbol{o}_i$ denotes the output of i^{th} expert.

G. Prediction and Loss

The final CTR prediction layer is formulated as follows:

$$\hat{y} = f_\theta(\boldsymbol{x}) = \mathcal{F}\left(\sum_{i=1}^{N}\alpha_i\cdot\boldsymbol{o}_i\right), \tag{7}$$

where $\mathcal{F}$ is a function implemented as a *MLP*, of which the last layer uses *Sigmoid* as activation function while the other layers use *ReLU*.

We adopt the widely-used cross-entropy loss during training our SAME model, which is defined as:

$$L = -\frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{x},y)\in\mathcal{D}}(y\,log\,\hat{y} + (1-y)\,log(1-\hat{y})). \tag{8}$$

TABLE I
STATISTICS OF THE ESTABLISHED DATASETS.

#Users	#Items	#Categories
5.85M	0.82M	4717
#Impressions	**#Clicks**	**#Purchases**
1.50B	75.39M	0.91M

IV. EXPERIMENTS

In this section, we conduct a series of experiments to answer the following research questions:

RQ1 How does SAME perform compared to the state-of-the-art models for the CTR prediction task in the context of both promotion and normal scenarios?

RQ2 Can FGN adapt to different scenarios and guide each expert to learn discriminative user-item feature representation?

RQ3 How does the number of experts in SAME affect the performance?

RQ4 Does the SPAU component necessarily contribute to the learning of user behavior and the improvement of the performance?

A. Experimental Setup

1) Datasets: We establish the datasets by collecting the users' interaction logs[1] from our online e-commerce platform, where promotions are highly frequent and have a considerable impact on the recommender system. Logs are sampled from 2020/10/01 to 2020/12/31, including Double 11, Black Friday, Double 12, and three monthly promotions (21st of every month). The detailed statistics are summarized in Table I. We split the entire datasets into non-overlapped training set and testing set according to the timestamp of the prediction behavior, effectively avoiding feature leakage. In this way, the training set is about 80% of the whole datasets and the left 20% of data is used as the testing set. To comprehensively evaluate the performance of the proposed model under scenarios of promotions and normal days, the testing set is further divided into two parts accordingly. All the data we use have been anonymously processed by the log system and users' information is protected.

2) Evaluation Metrics: To compare with the SOTA methods, area under ROC curve (AUC) is used as the offline evaluation metric. For online A/B testing, we choose click-through rate (CTR) and average number of user clicks (IPV), which are widely adopted in industrial recommender systems for evaluating online performance. Improving CTR and IPV simultaneously implies not only more accurate recommendation but also more active users.

3) Comparison Methods: We compare our SAME model with three classes of the previous methods: 1) methods that capture high-order feature interactions; 2) sequential user behaviors based methods; and 3) multi-task learning based methods. We also include two variants of our model for ablation study. They are briefly described as follows:

- **DCN** [34] explicitly applies feature crossing at each layer while only adding negligible extra complexity to a DNN model.
- **DIEN** [44] is a two-layer RNN structure with an attention mechanism. It models interests evolving process from user behaviors and calculates attention values to control the second RNN layer to activate the most relative interests to the candidate item.
- **MMoE** [17] is a representative model of multi-task learning methods. In this work, MMoE is trained with CTR prediction task and CVR prediction task, using the same expert structure as SAME. The CVR task is chosen because conversion is more sensitive to promotions.
- **TIEN** [16] develops a time-interval attention layer to calculate the importance weight for each user in item behaviors and captures the popularity of the items by a time-aware evolution layer. It strengthens the ability to predict users' emerging interests by modeling items in a dynamic manner.
- **SAME-OSE**. **SAME** with **O**nly a **S**ingle **E**xpert is a simplified version of SAME, $i.e.$, using a single expert without FGN. It can also be regarded as a variant of DIEN, of which the two-layer RNN structure with an attention is replaced with the SPAU.
- **SAME-OSE-NoLN** is almost the same as SAME-OSE, without LN-QK.

Note that for all the above methods, scenario signals are processed as the other input features, $i.e.$, discretized and embedded into low dimension vectors.

4) Implementation Details: We implement these deep learning models in distributed Tensorflow 1.4. During training, we use 3 parameter servers and 6 Nvidia Tesla V100 16GB GPU workers. Item ID, category ID and brand ID have an embedding size of 32 while 8 for the other categorical features. We use 8-head attention structures with a hidden size of 128. Both *Main Net* and *Bias Net* are *MLP*s with 3 layers. As for the FGN, a 3-layer *MLP* is used after Gaussian log1p transformation. Adagrad optimizer with a learning rate of 0.01 and a mini-batch size of 256 is used for training. The number of experts N varies from 2 to 5 and is set to 2 by default. We report the results of each method under its empirically optimal hyper-parameters settings.

B. Experimental Results: RQ1

1) Offline Results: Table II presents the results of all methods in both promotion and normal scenarios. The major observations are summarized as follows:

- SAME-OSE is the runner-up in normal days, ranks third in promotions, and outperforms DIEN in both scenarios, validating the effectiveness of the proposed SPAU.
- With auxiliary CVR prediction task, MMoE becomes the runner-up method in promotion scenarios, outperforming SAME-OSE slightly. However, it's not comparable to SAME-OSE in normal days, implying that MMoE fails to adapt to different scenarios without paying enough attention to scenario signals.

[1] To the extent of our knowledge, there are no public datasets suited for this promotion-aware CTR prediction task.

TABLE II
OFFLINE RESULTS IN BOTH PROMOTION AND NORMAL SCENARIOS, WITH EVALUATION OF EACH MODEL RUNNING FOR 3 TIMES. BOLD: BEST. UNDERLINE: RUNNER-UP.

Model	AUC (mean±std.)	
	Promotion	Normal
DCN	0.7132±0.00035	0.7069±0.00028
DIEN	0.7174±0.00171	0.7145±0.00162
MMoE	$\underline{0.7231\pm0.00125}$	0.7157±0.00101
TIEN	0.7143±0.00225	0.7140±0.00301
SAME-OSE	0.7216±0.00112	$\underline{0.7180\pm0.00109}$
SAME-OSE-NoLN	0.7169±0.00163	0.7153±0.00157
SAME	**0.7457±0.00098**	**0.7376±0.00057**

TABLE III
RESULTS OF ONLINE A/B TESTING.

Model	Promotion		Normal	
	CTR	IPV	CTR	IPV
DIEN	3.59%	1.96	3.54%	1.91
MMoE	$\underline{3.63\%}$	$\underline{2.05}$	3.54%	1.90
SAME-OSE	3.63%	2.02	$\underline{3.56\%}$	$\underline{1.98}$
SAME	**3.76%**	**2.14**	**3.70%**	**2.11**

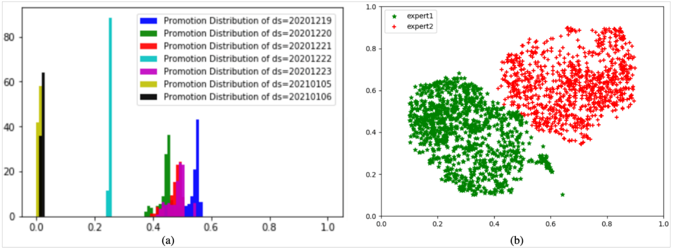


Fig. 3. (a) Distributions of the gated weights in different scenarios. (b) Visualization of feature representations from two experts using t-SNE [29].

TABLE IV
INFLUENCE OF THE NUMBER OF EXPERTS ON PERFORMANCE.

Number of experts	AUC (mean±std.)
2 experts	0.7396±0.00083
3 experts	0.7388±0.00122
4 experts	0.7383±0.00145
5 experts	0.7269±0.00171

- TIEN delivers unsatisfactory performance because the degree of user activity in our business is low so the huge user embedding parameters introduced by item behaviors can't be sufficiently trained.
- For both promotion and normal scenarios, SAME yields the best performance, outperforming the runner-up methods by a large margin.

2) Online A/B testing: From 2021/01/18 to 2021/01/31, online A/B testing was conducted in our online recommender system. DCN has been offline for a long time since it was outperformed by SAME-OSE, while SAME-OSE-NoLN and TIEN can't perform well in offline evaluation, so they were not selected for online tests. SAME-OSE was the main online deployed model and served as the baseline model, while MMoE was another choice because we observed that the CVR prediction is more sensitive to promotions. As shown in Table III, compared to the baseline model, SAME improves 3.58% on CTR and 5.94% on IPV during promotions while 3.93% and 6.57% in normal days respectively, outperforming all the other models. The narrowed gap of CTR and IPV between different scenarios demonstrates that our model has a better adaption ability, which helps to recommend more attractive items and improves the amount of purchases.

C. Visualization of the FGN and Experts Network: RQ2

In this section, we conduct an analysis on FGN and Experts Network to study how they contribute to the final performance. We feed samples from different scenarios into SAME and generate gated weights for all samples. To be specific, scenarios of normal days (20210105-20210106), days right before promotion (20201219-20201220), middle of promotion (20201222) when users' primary demands have been released, and the other days of promotion (20201221, 20201223) are included. Then, distributions of the gated weights of different days are plotted in different colors as shown in Figure 3(a), of which the x-axis refers to the value of gated weights and the y-axis refers to the amount of samples. It can be observed that gated weights of days of the same scenario are close while for different scenarios, *e.g.*, normal scenario, middle of promotion and the other scenarios, the gated weights distinguish from each other obviously, which implies that the FGN can adapt to different scenarios.

To validate the effectiveness of the Experts Network, we randomly sample hundreds of samples and feed them into SAME. The output of the two experts is visualized using t-SNE [29]. As shown in Figure 3(b), representations of the two experts clearly distinguish from each other, which implies that different experts in the Experts Network are able to learn discriminative user-item feature representation patterns under the modulation of FGN.

D. Influence of the Number of Experts: RQ3

We investigate the influence of the number of experts in our SAME model. We retrain SAME with the number of experts varying from 2 to 5 and evaluate these models with the whole test set, observing a decreasing trend of AUC which is detailed in Table IV.

The visualization displayed in Figure 3(b) and Figure 4 indicates that representations between experts become more indistinguishable as the number of experts increases from 2 to 5, *i.e.*, two experts can capture clearly distinguishable representation patterns while as experts increase the captured representation patterns mix with each other. Intuitively, we conclude that with too many experts the learned representation pattern of each expert may entangle with each other, leading to the final performance degradation. In the promotion-aware CTR prediction task, two experts can get the best results. When SAME is applied to the other similar tasks, we consider that best performance may be achieved when each expert corresponds to a certain representation pattern empirically.

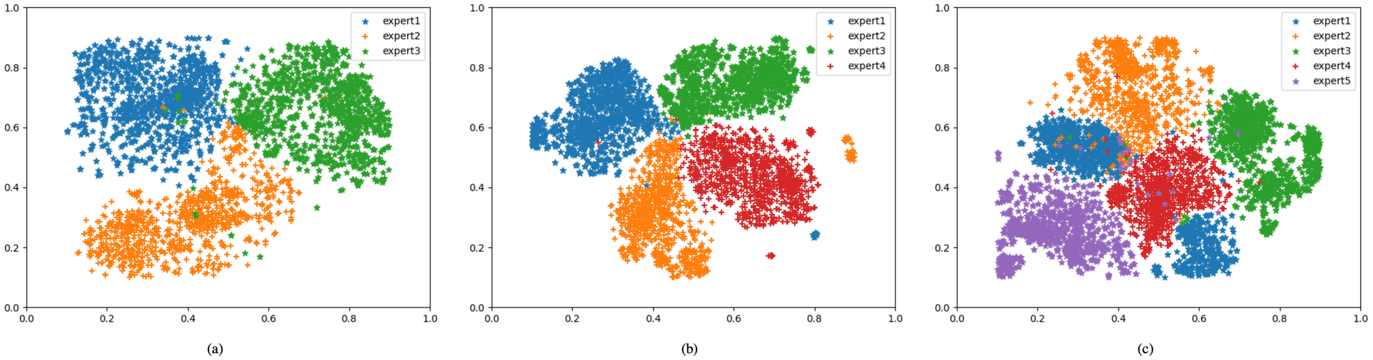


Fig. 4. Visualization of feature representations from 3, 4, 5 experts respectively.

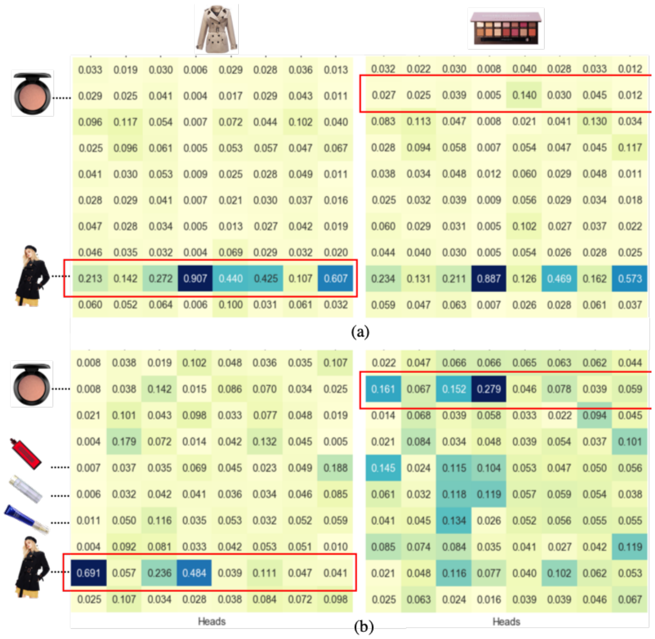
E. Study of Attention: RQ4

The offline and online results in Table II and Table III indicate that SAME-OSE outperforms DIEN consistently. This ablation study confirms the superiority of the SPAU in sequential modeling. Another ablation study is conducted by comparing SAME-OSE and SAME-OSE-NoLN. The results shown in Table II confirm the effectiveness of LN-QK.

To be more intuitive, we visualize the attention weights of the multi-head target attention corresponding to a user randomly sampled from the test datasets. As shown in Figure 5, given the foundation make-up as the target item, the SAME-OSE-NoLN fails to attend to the related cosmetic items in the user's historical behaviors, while the SAME-OSE is able to highlight the related historical interactions correctly for both clothing and foundation make-up. The above case further illustrates that LN-QK can contribute to extracting information related to the target item and eliminating the impact of irrelevant items especially when user behaviors are highly sparse and incoherent, while the existing sequential modeling methods cannot perform well.

V. Conclusion

In this paper, we investigate the difficulties of CTR prediction on e-commerce platforms with frequent promotions and propose a simple yet effective SAME model for both promotion and normal scenarios, *i.e.*, Scenario Adaptive Mixture-of-Experts. Our SAME model consists of a Feature Gated Network (FGN) and Experts Network, where the former can learn distinct gated weights for different scenarios and guides the latter to learn distinguishing feature representation patterns by different experts. In this way, our SAME model can adapt to both promotion and normal scenarios and outperforms representative CTR methods on both real-world offline datasets as well as online A/B testing. The empirical study of gated weights distribution and feature representation visualization further confirms the effectiveness of the proposed FGN and Experts Network. With the help of SPAU, SAME can overcome the shortcoming of existing sequential modeling methods in situations of highly sparse and incoherent user behaviors. The exploration on the number of experts shows useful insights for future work, *i.e.*, adapting SAME to scenarios in the other context beyond promotions.



0.033	0.019	0.030	0.006	0.029	0.028	0.036	0.013	0.032	0.022	0.030	0.008	0.040	0.028	0.033	0.012
0.029	0.025	0.041	0.004	0.017	0.029	0.043	0.011	0.027	0.025	0.039	0.005	0.140	0.030	0.045	0.012
0.096	0.117	0.054	0.007	0.072	0.044	0.102	0.040	0.083	0.113	0.047	0.008	0.021	0.041	0.130	0.034
0.025	0.096	0.061	0.005	0.053	0.057	0.047	0.067	0.028	0.094	0.058	0.007	0.054	0.047	0.045	0.117
0.041	0.030	0.053	0.009	0.025	0.028	0.049	0.011	0.038	0.034	0.048	0.012	0.060	0.029	0.048	0.011
0.028	0.029	0.041	0.007	0.021	0.030	0.037	0.016	0.025	0.032	0.039	0.009	0.056	0.029	0.034	0.018
0.047	0.028	0.034	0.005	0.013	0.027	0.042	0.019	0.060	0.029	0.031	0.005	0.102	0.027	0.037	0.022
0.046	0.035	0.032	0.004	0.069	0.029	0.032	0.020	0.044	0.040	0.030	0.005	0.054	0.026	0.028	0.025
0.213	0.142	0.272	0.907	0.440	0.425	0.107	0.607	0.234	0.131	0.211	0.887	0.126	0.469	0.162	0.573
0.060	0.052	0.064	0.006	0.100	0.031	0.061	0.032	0.059	0.047	0.063	0.007	0.026	0.028	0.061	0.037

(a)

0.008	0.038	0.019	0.102	0.048	0.036	0.035	0.107	0.022	0.047	0.066	0.066	0.065	0.063	0.062	0.044
0.008	0.038	0.142	0.015	0.086	0.070	0.034	0.025	0.161	0.067	0.152	0.279	0.046	0.078	0.039	0.059
0.021	0.101	0.043	0.098	0.033	0.077	0.048	0.019	0.014	0.068	0.039	0.058	0.033	0.022	0.094	0.045
0.004	0.179	0.072	0.014	0.042	0.132	0.045	0.005	0.021	0.084	0.034	0.048	0.039	0.054	0.037	0.101
0.007	0.037	0.035	0.069	0.045	0.023	0.049	0.188	0.145	0.024	0.115	0.104	0.053	0.047	0.050	0.056
0.006	0.032	0.042	0.041	0.036	0.034	0.046	0.085	0.061	0.032	0.118	0.119	0.057	0.059	0.054	0.038
0.011	0.050	0.116	0.035	0.053	0.032	0.052	0.059	0.041	0.045	0.134	0.026	0.052	0.056	0.055	0.055
0.004	0.092	0.081	0.033	0.042	0.053	0.051	0.010	0.085	0.074	0.084	0.035	0.041	0.027	0.042	0.119
0.691	0.057	0.236	0.484	0.039	0.111	0.047	0.041	0.021	0.048	0.116	0.077	0.040	0.102	0.062	0.053
0.025	0.107	0.034	0.028	0.038	0.084	0.072	0.098	0.025	0.063	0.024	0.016	0.039	0.039	0.046	0.067

Fig. 5. Cases of 8-head attention weight matrix in different models: (a) SAME-OSE-NoLN; (b) SAME-OSE.

References

[1] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[2] Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-criteria learning for chinese word segmentation. *arXiv preprint arXiv:1704.07556*, 2017.

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[4] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[5] Sven F Crone and Nikolaos Kourentzes. Feature selection for time series prediction–a combined filter and wrapper approach for neural networks. *Neurocomputing*, 73(10-12):1923–1936, 2010.

[6] Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1):123–149, 2010.

[7] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

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

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

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

[12] Wendi Ji, Keqiang Wang, Xiaoling Wang, Tingwei Chen, and Alexandra Cristea. Sequential recommender via time-aware attentive memory network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 565–574, 2020.

[13] Mahesh Joshi, Mark Dredze, William Cohen, and Carolyn Rose. Multi-domain learning: when do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1302–1312, 2012.

[14] Kyeongpil Kang, Junwoo Park, Wooyoung Kim, Hojung Choe, and Jaegul Choo. Recommender system using sequential and global preference via attention mechanism and topic modeling. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1543–1552, 2019.

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

[16] Xiang Li, Chao Wang, Bin Tong, Jiwei Tan, Xiaoyi Zeng, and Tao Zhuang. Deep time-aware item evolution network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 785–794, 2020.

[17] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed 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*, pages 1930–1939, 2018.

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 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*, pages 1137–1140. ACM, 2018.

[19] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 596–605, 2018.

[20] Xichuan Niu, Bofang Li, Chenliang Li, Jun Tan, Rong Xiao, and Hongbo Deng. Heterogeneous graph augmented multi-scenario sharing recommendation with tree-guided expert networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1038–1046, 2021.

[21] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1149–1154. IEEE, 2016.

[22] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems (TOIS)*, 37(1):1–35, 2018.

[23] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):121–135, 2017.

[24] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 565–574, 2019.

[25] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[26] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo 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*, pages 4104–4113, 2021.

[27] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Fourteenth ACM Conference on Recommender Systems*, pages 269–278, 2020.

[28] Thanh Tran, Di You, and Kyumin Lee. Quaternion-based self-attentive long short-term user preference encoding for recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1455–1464, 2020.

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

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[31] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1110, 2020.

[32] Jianling Wang, Kaize Ding, Ziwei Zhu, and James Caverlee. Session-based recommendation with hypergraph attention networks. 2021.

[33] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. Time to shop for valentine's day: Shopping occasions and sequential recommendation in e-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 645–653, 2020.

[34] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. 2017.

[35] Hong Wen, Jing Zhang, Fuyu Lv, Wentian Bao, Tianyi Wang, and Zulong Chen. Hierarchically modeling micro and macro behaviors via multi-task learning for conversion rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[36] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 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*, pages 2377–2386, 2020.

[37] Dingming Wu, Xiaolong Wang, Jingyong Su, Buzhou Tang, and Shaocong Wu. A labeling method for financial time series prediction based on trends. *Entropy*, 22(10):1162, 2020.

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

[39] Weinan Xu, Hengxu He, Minshi Tan, Yunming Li, Jun Lang, and Dongbai Guo. 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*, pages 1905–1908, 2020.

[40] Feng Yu, Zhaocheng Liu, Qiang Liu, Haoli Zhang, Shu Wu, and Liang Wang. Deep interaction machine: A simple but effective model for high-order feature interactions. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2285–2288, 2020.

[41] Ming Zeng, Hancheng Cao, Min Chen, and Yong Li. User behaviour

modeling, recommendations, and purchase prediction during shopping festivals. *Electronic Markets*, 29(2):263–274, 2019.

[42] Jing Zhang and Dacheng Tao. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2021.

[43] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. Atrank: An attention-based user behavior modeling framework for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[44] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5941–5948, 2019.

[45] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068, 2018.

[46] Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. Feature transformation for neural ranking models. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1649–1652, 2020.