Scene-wise Adaptive Network for Dynamic Cold-start Scenes Optimization in CTR Prediction

Wenhao Li
Huazhong University
of Science and Technology
Beijing, China
rzliwenhao@hust.edu.cn

Jie Zhou
School of Software,
Beihang University
Beijing, China
zhoujiee@buaa.edu.cn

Chuan Luo*
School of Software,
Beihang University
Beijing, China
chuanluo@buaa.edu.cn

Chao Tang
Meituan
Beijing, China
tangchao12@meituan.com

Kun Zhang
Meituan
Beijing, China
zhangkun32@meituan.com

Shixiong Zhao*
The University of Hong Kong
Hong Kong, China
sxzhao@cs.hku.hk

Abstract

In the realm of modern mobile E-commerce, providing users with nearby commercial service recommendations through location-based online services has become increasingly vital. While machine learning approaches have shown promise in multi-scene recommendation, existing methodologies often struggle to address cold-start problems in unprecedented scenes: the increasing diversity of commercial choices, along with the short online lifespan of scenes, give rise to the complexity of effective recommendations in online and dynamic scenes. In this work, we propose **S**cene-**w**ise **A**daptive **N**etwork (SwAN[1]), a novel approach that emphasizes high-performance cold-start online recommendations for new scenes. Our approach introduces several crucial capabilities, including scene similarity learning, user-specific scene transition cognition, scene-specific information construction for the new scene, and enhancing the diverged logical information between scenes. We demonstrate SwAN's potential to optimize dynamic multi-scene recommendation problems by effectively online handling cold-start recommendations for any newly arrived scenes. More encouragingly, SwAN has been successfully deployed in Meituan's online catering recommendation service, which serves millions of customers per day, and SwAN has achieved a 5.64% CTR index improvement relative to the baselines and a 5.19% increase in daily order volume proportion.

CCS Concepts

• **Computing methodologies → Probabilistic reasoning**.

---

*Corresponding authors.
[1]https://github.com/ChrisLiiiii/SwAN

Keywords

Recommendation, Multi-Scene, Cold-Start

1 Introduction

Delivering users with nearby commercial service suggestions through location-based online systems [14, 18, 28] has grown increasingly crucial within the era of modern mobile E-commerce. Learning to Rank (LTR) involves applying machine learning algorithms [3, 12] in optimizing the rank strategy, and is the fundamental technique to facilitate better recommendation services. Contemporary recommendation systems not only focus on users' habits derived from historical information but also endeavor to infer the preferences of the same user across diverse scenes, facilitating more accurate and high-quality multi-scene recommendation (MSR) [23].

Despite the considerable advancements in MSR research, a majority of these developments are grounded in the assumption that scenes are predefined and classified prior to offline training, with all subsequent recommendations adhering to established categories during online operations.

Therefore, the existing literature (e.g.SAML [2], STAR [19], and HMoE [11]) on MSR primarily concentrates on a static model architecture that distinguishes scenes by directing inputs of each scene to a fixed structural branch within the model.

However, empirical evidence reveals that this assumption does not always hold true. As the assortment of items and options expands in today's world, a proliferation of distinct scenes arises. Consequently, users' behaviors tend to diverge more frequently, leading to an increased variety of scenes without previous identical scenes available for reference in historical data [9].

According to Fig. 1, the online recommendation service will launch different scenes during specific periods in spring or winter, taking into account user preferences and merchant demands. Additionally, it will also design exclusive activities for specific holidays, such as New Year's Day. On the other hand, scenes often have a limited online lifespan before vanishing (e.g.Valentine's Day

Fig 1: The figure displays multiple business scenes' online / offline states over time. The *x*-axis represents time, with green and blue segments indicating the spring and winter. Boxes above the *x*-axis show the online and offline activities during certain periods within each season. Activity diagrams represent online status, while dashed lines represent offline (e.g., the Valentine's Day activity is only online on Tuesdays in the left graph). The findings suggest that scenes go online immediately when an activity starts (cold-start problem) and go offline right after it ends (limited online time and sample accumulation). The actual online business is even more time-sensitive, with over 200 scenes going online / offline on average each month. This is the dynamic multi-scene problem introduced in this paper, which poses significant challenges to existing multi-scene models.

in Fig. 1), leaving no opportunity for a recommendation system to collect data, go offline for fine-tuning, and return online [16]. The Hybrid of implicit and explicit Mixture-of-Experts (HMoE) [11] demonstrates that the performance of one scene can be enhanced (through training) by the prediction of other scenes. Unfortunately, HMoE still requires learning the historical data of a new scene and sharing information between scenes through re-parameterization.

In this paper, we demonstrate that the performance of a newly-arrived scene can be directly and significantly improved through online prediction using our **S**cene-**w**ise **A**daptive **N**etwork (SwAN) model. This suggests that cold-starting new scenes is not only feasible but also surpasses the recommendation performance of existing approaches on known scenes.

In general, SwAN employs the typical Embedding&MLP (Multi-layer Perceptron) paradigm for the recommendation [24] (Sec. 3). In the Embedding part, SwAN utilizes the Scene Relation Graph (SRG) to capture graph-structured similarities between scenes based on inherent attributes and user interaction features, thereby learning the inertial patterns among scenes. The SwAN model also incorporates the Similarity Attention Network (SAN) to capture users' habits during scene transitions by applying user attention on scene similarity knowledge. Furthermore, SwAN assigns each known scene a separate feature embedding (Scene Embedding Layers) to understand how scenes individually influence user behavior and interlace them with the SAN, allowing the impact of new scenes on users to be directly derived. In the MLP part, SwAN generally adopts the Adaptive Ensemble-experts Module (AEM), which is a Mixture-of-Experts (MoE) architecture and includes an Adaptive Expert Group (AEG) of Sparse MoE that uniquely leverages Cosine Loss to enhance diversities between scenes, as well as a Shared Expert Group (SEG) of Multi-gate MoE that captures the shared

logic of scenes. In particular, a novel component named *Dics* has been proposed for the AEG to achieve gradient propagation and select appropriate model structures adaptively.

Extensive evaluation on both the public and industrial datasets shows that the SwAN model outperforms existing MSR approaches by seamlessly adapting to new scenes and providing more accurate and high-quality recommendations. SwAN achieves up to 5.64% online CTR improvement relative to the baselines and up to 5.19% increase in daily order volume proportion, as evaluated in Sec. 4.5.

The main contributions of this paper are as follows:

- We propose SwAN, an innovative high-performance multi-scene cold-start optimization network.
- Innovatively, we propose SRG to acquire prior information from similar scenes for cold-start scenes and employ SAN to get the attention weight of these scenes from user's perspective. Finally, AEM dynamically allocates model structures to enhance the extraction capability of shared and specific information across different scenes.
- SwAN has been deployed in a real-world online business recommendation system of Meituan and achieved a 5.64% improvement in CTR compared to the baseline model.

2 Related work

Multi-scene learning tackles recommendations for users across various scenes [29]. Traditional models for multi-scene learning have been developed to enhance performance in multiple fixed scenes. Drawing inspiration from the Multi-task Mixture-of-Experts model, Li [11] introduced HMoE that implicitly identifies scene disparities and similarities in the feature space and explicitly enhances performance in the label space using a stacked model.
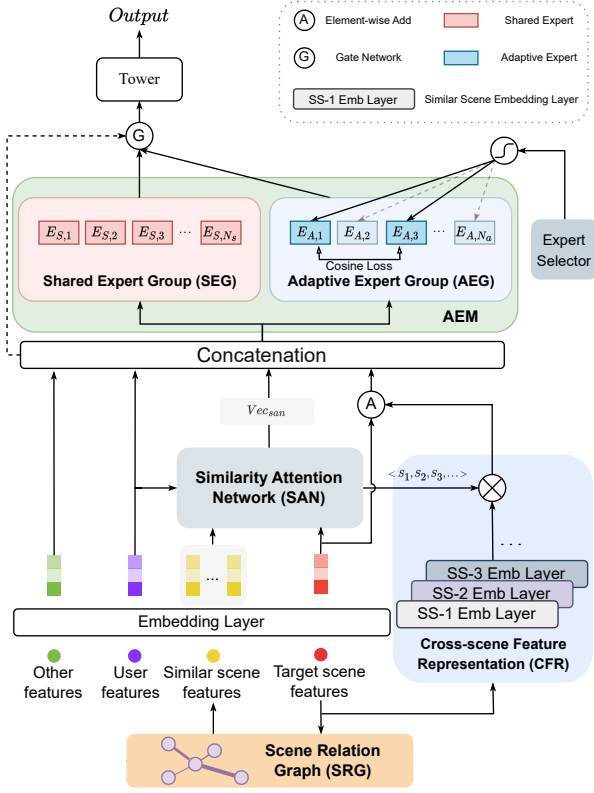
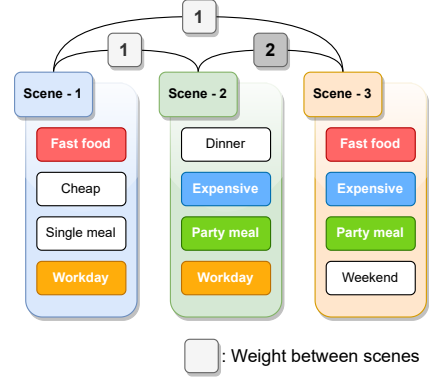Fig 2: Schematic diagram of the SwAN structure.

Fig 3: The relation between different scenes in SRG. The numbers on the lines are the number of the same key features.

scene data through the Long Short-Term Memory (LSTM) [8] module to guide the model's training direction and early-stop timing in new scenes. Nevertheless, it results in high-cost consumption when applied to multiple cold-start scenes, and it cannot comprehensively consider the distribution rules of multiple scene data.

3 Approach

This section presents our design of the proposed SwAN model (Fig. 2). In essence, SwAN follows the key principle of optimizing multi-scene (by extracting scene-specific and shared information) and cold-start (by incorporating data from similar scenes as supplements) problem and consists of multiple modules: the Scene Relation Graph (Sec.3.1), Similarity Attention Network (Sec.3.2), Cross-scene Feature Representation (Sec.3.3), Adaptive Ensemble-experts Module (Sec.3.4). The Decision Layer (Sec.3.5) of SwAN and the loss function (Sec.3.6) are appended.
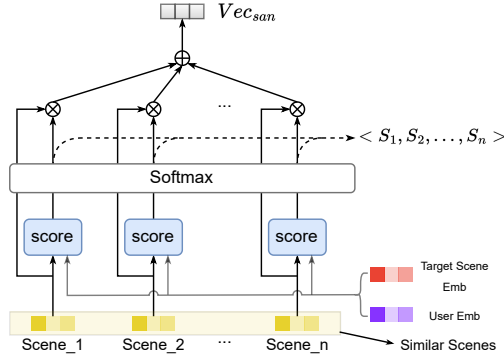
3.1 Scene Relation Graph (SRG)

A scene comprises inherent attribute features and user interaction features [1]. In dynamic multi-scene problems, there is no historical interaction data between users and new scenes, which means that only scene attribute features can be invoked to collect information.

Fortunately, users exhibit similar preferences in comparable scenes. Based on our post-fact online business analysis, users often perceive a positive correlation between the similarity of scenes and the similarity of item features within those scenes [7]. This allows a recommendation system to optimize the cold-start process by leveraging prior information from analogous scenes to resemble the target scene closely.

Based on these premises, SwAN invokes a Scene Relation Graph (SRG) module that builds a relational graph between the current scene (to be predicted) and the existing scenes based on the scene features. The construction process of SRG is as follows:

(1) Firstly, it lists the basic features (unrelated to online interactions, e.g.price and category) of the items to be sorted and uses user key interactions as labels to calculate the Pearson correlation coefficient of various features in the existing scenes, selecting the top-n (Sec. 4.3 for details) key features.

Unfortunately, the model did not distinguish different scenes expert-wise, resulting in insufficient mining of scene-specific information and affecting the model's ability to represent scenes.

To capture the diverse characteristics of various scenes and thus serve them equitably, Sheng *et al.* [19] brings the STAR model which leverages data from all scenes. PEPNet [1] implements an efficient, low deployment cost, plug-and-play multi-scene modeling paradigm by constructing parameters and embedding personalization. Zhou et al. [27] proposed HiNet, which employs multi-tasks and multi-scenes explicitly and hierarchically using a hierarchical MoE to model commonality and individuality among multi-scenes.

However, the aforementioned models are designed for multiple fixed scenes, so they need re-training with additional model structure when applied to dynamically increasing cold-start scenes.

The cold-start problem is an open and challenging research problem in the field of recommendation systems [6].

Zhu *et al.* [31] proposed the Meta Warm Up Framework (MWUF) based on meta-learning and considered that the embeddings for cold-start and warm-up stages are in different spaces. The MWUF designed Meta Scaling Network and Meta Shifting Network to map cold-start embeddings to the warm-up space and eliminate noise. However, the MWUF mainly optimizes item cold-start and is unsuitable for scene cold-start. Besides, Du *et al.* [5] developed the scene-specific Sequential Meta learner (s^2Meta) based on meta-learning. The s^2Meta model mainly learns the gradient and loss changes of the target model while fitting the distribution of old

Fig 4: Similarity Attention Network.

(2) Secondly, it aggregates the key features of items in cold-start scenes to obtain scene-level features such as averages, variances, maximums, and minimums (measuring the distribution patterns of each feature).

(3) Lastly, it categorizes the above features and counts the number of identical features between scenes as the edge weights (Fig. 3, Scene-2 and Scene-3 share 2 identical attributes).

By doing so, the SRG module obtains a similarity rank between the current and existing scenes by calculating raw feature explicit similarity, making SwAN flexible in choosing a threshold to invoke scenes with certain weighted similarities to the target scene.

3.2 Similarity Attention Network (SAN)

However, determining similar scenes based solely on attributes is inadequate. In real-world applications, various users perceive the same scene pair differently, and the model must incorporate user cognition to comprehend the latent similarity between scenes on a deeper level [30]. For instance[2], some individuals consider horror movies and zombie movies part of the same genre, while others do not.

Consequently, our model enhances the SRG module by introducing user information for attention. This is achieved by incorporating a Similarity Attention Network (SAN, shown in Fig. 4) to calculate learned latent similarity from the user's perspective.

The input of the SAN includes the features of the target scene, the similar scenes defined in the SRG, and the user. The specific attention calculation (referred to the DIN [26]) is as follows:

$$\hat{S}_i = MLP[E_u \oplus E_t \oplus (E_t - E_s^i) \oplus (E_t \otimes E_s^i)], \quad (1)$$

$$S_i = softmax(\hat{S}_i), \quad (2)$$

$$Vec_{san} = \sum^I S_i \cdot E_s^i, \quad (3)$$

where E_u, E_t, and E_s^i are the embeddings of users, target scene, and the i-th similar scene, respectively; I denotes all the scenes; $\hat{S}_i$ is

[2]Overall, from our billion-level dataset, we found that 71.69% of users prefer breakfast within 2 km, while 63.41% of users choose regular meals within a 2-5 km range. Hence, the distance feature has varying impacts on breakfast and regular meal recommendations. Regarding Sec. 3.3, from our billion-level labeled data, only 49.62% of users consider zombie movies as horror films, and 35.42% express opposing views, indicating diverse perceptions among users (zombie movies can trigger aversion in some users). Additionally, the data analysis of various features, such as price, supports our findings.

the intermediate variable (The output of the blue "score" module in Fig. 4.); S_i is the latent learned similarity between the i-th similar scene and the target scene; operator $\oplus$ means concatenation, and operator $\otimes$ means element-wise product. This structure effectively integrates user cognition to learn the genuine similarity between scenes and outputs a weighted representation of prior information from analogous scenes, which enables a more rational ranking of samples in new scenes. Furthermore, the SRG and SAN structures boosted the performance of SwAN during the cold-starting of new scenes.

3.3 Cross-scene Feature Representation (CFR)

There are differences in the bottom-level feature representation for each scene as well [2]. For example, the distance between users and dining locations has different importance in the breakfast and regular meal scenes[2].

To reflect these differences and provide information supplementation for the cold-start embedding of the target scene, SwAN added a Cross-scene Feature Representation (CFR) structure to the feature processing module (Fig. 2), which essentially assigns each extent scene a separate embedding to capture a scene's properties solely. Specifically, the input of CFR is the scene-related features f_{target_scene} and the similarity between scenes output by SAN, and the calculation formula is as follows:

$$E_{cfr} = \sum^I S_i \cdot EMB_i(f_{target\_scene}), \quad (4)$$

where $EMB_i(\cdot)$ is the embedding layer corresponding to the i-th similar scene. The input of the subsequent model is:

$$E_{in} = E_o \oplus E_u \oplus Vec_{san} \oplus (E_t + E_{cfr}), \quad (5)$$

where E_o means the embedding of other features, and + means element-wise addition.

The model transfers prior information from similar scenes regarding feature representation dimensions through CFR, optimizing the cold-start problem and enhancing the expression of differences between scenes. In addition, since CFR essentially involves multiple dictionary lookups and weighted vector summation, it does not introduce excessive computational overhead.

3.4 Adaptive Ensemble-experts Module (AEM)

Traditional static multi-scene models usually set up separate model branches for each scene (e.g.STAR [19]), using structural differences to improve the ability to mine diverged information and optimize negative transfer problem, which are the cores of multi-scene modeling. However, in dynamic multi-scene problems, numerous scenes go online and offline frequently. The traditional model design approach cannot assign model structure for cold-start scenes, while the strategy of retraining the model based on a small number of cold-start scene samples leads to computational redundancy and require frequent offline fine-tuning to update the model architecture. To solve the above problems, we designed Adaptive Ensemble-experts Module (AEM) as the backbone network of the model to enhance the ability to extract differential information and optimize negative transfer in dynamic and multi-scene environments (Fig. 2).

Firstly, we draw inspiration from the MMoE model [15] and develop multiple expert networks to enhance the model's ability

to mine information. Secondly, we divide the expert networks into groups that improve the model's ability to extract scene-specific and shared information.

The Adaptive Experts Group (AEG) is responsible for extracting scene-specific information. To avoid the high cost of model training caused by frequent scene updates, AEG adopts a dynamic combination of experts to calculate differentiated weights for different scene samples, which means learning how to allocate model structures adaptively. This function is mainly implemented by the Expert Selector (ES). As shown in Fig. 2, the input of ES is the weighted similar scene representation obtained by SAN, and the output is the gate weight (0 or 1) of each expert in AEG. In this way, ES transfers the prior information of expert selection from similar scenes. AEG enhances the model's ability to extract different scene-specific information and optimizes the cold-start phase of new scenes.

In more detail, the computations in ES consist of two steps: (1) Generate selection probabilities for each expert and the unique threshold (to unify all expert-selected baselines and enhance stability). (2) Output a weight of 1 if the probability exceeds the threshold and 0 otherwise. The specific formulas are as follows:

$$P_k = sigmoid[MLP_p(E_u \oplus Vec_{san})], \tag{6}$$

$$T = sigmoid[MLP_{thre}(Vec_{san})], \tag{7}$$

where P_k is the selection probability of the k-th expert, and T is the probability threshold. In addition, incorporating E_u into the calculations of P_k allows for more accurate computations from the user's perspective, similar to the SAN.

However, using a step function or a threshold function to compare probabilities and thresholds can lead to gradient interruption, which means that the MLP model for generating probabilities and thresholds cannot be trained. To solve this problem, we have designed a Differentiable conditional selection unit ($Dics$) based on the sigmoid function:

$$W_k = Dics(P_k, T) = \frac{1}{1 - e^{-\frac{1}{\tau} \cdot (P_k - T)}}, \tag{8}$$

where τ is a temperature coefficient greater than 0 and W_k is the weight of the k-th expert. By performing the aforementioned operations, $Dics$ dynamically selects appropriate model structures for cold-start scenes based on similar scene information, achieving adaptability at the model architecture level.

When the value of τ is close to 0, the output weight is more relative to 0 or 1. However, excessively small τ leads to unstable model training. To address this issue, we introduce the variance loss of the gate values for each expert to increase the variance between gate values and move them closer to 0 and 1.

AEG is built on ensemble learning, so improving the differences between sub-learners' outputs can help enhance model performance [17]. We add a cosine similarity loss function between the outputs of each expert in AEG, with the specific formula as follows:

$$Loss_{cos} = \sum_m^{AEG} \sum_{n \neq m}^{AEG} \left| \frac{E_a^m \cdot E_a^n}{\|E_a^m\| \cdot \|E_a^n\|} \right|, \tag{9}$$

where E_a^m and E_a^n are output vectors of two different experts in the AEG, and AEG is the whole AEG.

In addition, SwAN constructs the Shared Experts Group (SEG) to enhance the extraction performance of shared information between

scenes, following the approach of classic multi-scene models[3]. The experts of this module remain consistent across all scenes.

In summary, AEM adaptively transfers the model-building approach for similar scenes to the current one. The model enhances the generalization ability to new scenes and improves the ability to mine scene-specific and shared information, providing a solution to optimize dynamic multi-scene problems.

3.5 Decision Layer

By utilizing AEM, SwAN extracts the shared and specific information of scenes, which is contained in the output vectors of each expert in SEG and AEG, respectively. However, the contribution of each vector to the final prediction target varies. To address this issue, inspired by the solution of MMoE, we add a gating network for each expert:

$$G_i = MLP_g(E_{in}), \tag{10}$$

$$E_{final\_in} = \sum_{i=0}^{|SEG|} G_i \cdot Vec_s^i + \sum_{i=0,k=0}^{|AEG|} G_i \cdot W_k \cdot Vec_a^i, \tag{11}$$

where E_{final_in} is the input of the MLP structure in the output stage of SwAN, G_i is the gate value of each expert, $MLP_g(\cdot)$ is the MLP structure to calculate the gate value, and Vec_s and Vec_a are the expert outputs of SEG and AEG, respectively. The output of the decision layer, namely, the final output of SwAN, can be expressed as:

$$Output = sigmoid[MLP(E_{final\_in})]. \tag{12}$$

3.6 Composition of Losses

SwAN uses the Cross-Entropy loss function between output and label to guide training. The formula is as follows:

$$Loss_{ce}(y, \hat{y}) = \frac{1}{N} \sum -[y \cdot log(\hat{y}) + (1 - y) \cdot log(1 - \hat{y})], \tag{13}$$

where y and $\hat{y}$ are label and predicted value, respectively.

In addition, as described in Sec. 3.4, a variance loss is added:

$$Loss_{var} = \frac{\sum_A (W - \bar{W})^2}{N_a}, \tag{14}$$

where N_a is the number of experts in AEG.

To sum up, the loss function is as follows:

$$Loss = \alpha \cdot Loss_{ce}(y, \hat{y}) + \beta \cdot Loss_{cos} + \gamma \cdot Loss_{var}, \tag{15}$$

where α, β, and γ are hyper-parameters set according to the actual dataset (Sec. 4.3).

4 Experiments

To verify the effectiveness and generalization of the SwAN model, this study conducts experiments based on two datasets: a closed-source dataset from Meituan's online catering recommendation service with millions of daily users and an open-source dataset constructed from the Taobao public dataset [5].

4.1 Experimental Settings

Industrial Dataset. Samples from Dataset-1 are obtained from the online catering recommendation platform of Meituan, specifically from the business of Sales Campaign Session with an average daily

Table 1: The number of scenes contained in the dataset.

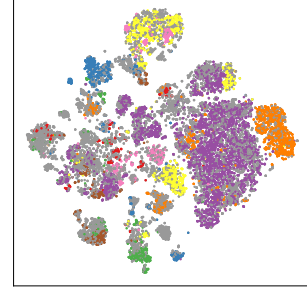| Dataset	Source	Train	Test	
			Overall	Cold-start
1	Meituan	751	309	207
2	Taobao	250	105	105

customer level of millions. This business designs promotional activity scenes to cater to the consumption preferences of users at different periods, thus involving many scenes and frequent updates. We selected three months and one month of actual user online interaction behaviors as the training and test sets, respectively. The training set contains 70 million samples, while the test set contains 30 million. The ratio of the number of positive samples between the number of negative samples is about 1 to 3. Table 1 shows that this dataset's training and testing sets contain 751 and 309 scenes, respectively. Among them, 207 scenes in the testing set have never appeared in the training set, representing cold-start scenes.

Table 2: Experimental Results (DMSM: dynamic multi-scene model).

Model	Type	Dataset-1		Dataset-2
		AUC *all*	AUC *cold-start*	AUC *cold-start*
SwAN (Ours)	DMSM	**0.7860**	**0.7799**	**0.6733**
DNN [4]	SSM	0.7646	0.7568	0.6601
DCN [21]	SSM	0.7718	0.7642	0.6617
xDeepFM [13]	SSM	0.7741	0.7649	0.6631
DCN-v2 [22]	SSM	0.7758	0.7655	0.6643
MMoE [15]	SSM	0.7755	0.7656	0.6638
PLE [20]	SSM	0.7787	0.7701	0.6682
HMoE [11]	SMSM	0.7749	0.7644	0.6673
STAR [19]	SMSM	0.7731	0.7607	0.6669
PEPNet [1]	SMSM	0.7767	0.7659	0.6679
HiNet [27]	SMSM	0.7759	0.7648	0.6677

Public Dataset. Samples from Dataset-2 are obtained from user click logs of cloud-based theme scenes on Taobao [3]. We followed the official instructions [5] and divided the dataset into training and testing sets, which include 250 and 105 different recommendation scenes, respectively. Actually, after dealing with Dataset-2 through the official instructions, it is guaranteed that none of the testing set scenes appears in the training set. In addition, to adapt the features of the Taobao dataset to the cold-start multi-scene recommendation case studied in our paper, we performed data clustering and further processing of the Taobao dataset.

The original Taobao dataset only contains item embeddings, scene theme IDs, and some item categories. The original dataset is not well suited for an effective evaluation of SwAN due to two rationales. First, the coverage of item category features is deficient (only 23.18%), which cannot produce practical scene attributes. Second, when we use t-SNE to reduce the dimensionality of item embeddings (as shown in Fig. 5), it can be observed that the distribution of items on the two-dimensional plane is not optimal (different colors represent different original categories). Specifically, there are

Fig 5: The t-SNE dimensionality reduction visualization of item embeddings, where different colors represent different categories in the original data (gray represents missing category information).

cases where different categories of items are clustered together, and items of the same category are dispersed. This graph also indicates a significant dissimilarity between different original categories. Therefore, it is unreasonable to calculate scene features using the item category information from the original data.

Table 3: Silhouette coefficients corresponding to different k values. The Silhouette coefficient measures how well each data point fits within its cluster and how well separated it is from other clusters.

k	2	3	4	6	9
Silhouette Coefficient	0.1233	**0.1437**	0.1387	0.1160	0.0993

In summary, the information in the original dataset does not meet the case setting of this paper. To address this problem, we applied the k-means algorithm to cluster the item embeddings provided in the dataset. We used the silhouette coefficient to evaluate the appropriateness of the selected hyper-parameter k. The relationship between k and the silhouette coefficient is shown in Table 3. The table shows that the optimal value for k is 3. Using clustering of item embeddings, we obtained the category information for all items. Further, we derived the inherent attributes of each scene, which meets the data requirements of our model in this paper.

Settings. The cross-entropy loss function and Adam optimizer [10] are used in the experiments. The number of experts in AEG and SEG is set equally to 10 as the default value. The value of τ, which is used in the $Dics(\cdot)$ function (Eq. 8), was set to 10^{-3} as default. Furthermore, we set α as 1, β and γ as 10^{-3} in Eq. 15. Hyper-parameter experiments can be found in Sec. 4.3.

In order to make our comparison fair, for the baseline model, its hyper-parameter settings were configured through the same method as SwAN does.

Metrics. In recommendation systems, items can be classified as relevant or irrelevant for a given user. We use the AUC (Area Under the Curve) score to evaluate how effectively the model can distinguish between these two classes of items. A higher AUC score indicates that the model can differentiate between relevant and non-relevant items for users in a more effective way, and have stronger capability of recommending items that users find interesting and

Table 4: Comparison of AUC of each model in 10 randomly selected cold-start scenes.

	SwAN	MMoE	PLE	HMoE	STAR	PEPNet	HiNet
#.1	**0.7524**	0.7476	0.7508	0.7487	0.7432	0.7469	0.7458
#.2	**0.7576**	0.7322	0.7128	0.7306	0.7312	0.7418	0.7391
#.3	**0.8114**	0.7924	0.7753	0.7788	0.7808	0.7797	0.7815
#.4	**0.7644**	0.7508	0.7308	0.7504	0.7471	0.7493	0.7459
#.5	**0.7714**	0.7704	0.7703	0.7693	0.7626	0.7659	0.7680
#.6	**0.7541**	0.7531	0.7535	0.7512	0.7492	0.7529	0.7514
#.7	**0.7409**	0.7248	0.7142	0.7404	0.7403	0.7399	0.7388
#.8	**0.8178**	0.7824	0.7866	0.7606	0.7843	0.7802	0.7851
#.9	**0.8060**	0.7769	0.7836	0.7748	0.7755	0.7768	0.7766
#.10	**0.8283**	0.7753	0.7633	0.7736	0.7874	0.7894	0.7890
All	**0.7869**	0.7676	0.7638	0.7587	0.7552	0.7591	0.7579

relevant. We used CTR (Click-Through-Rate) in online experiments, and the CTR measures the ratio of the number of clicks on recommended items to the number of items displayed. A higher CTR indicates that users find the recommended items more relevant and are more likely to click on them. Furthermore, the Gini coefficient is utilized to measure the uniformity of model improvement across different scenes. A lower Gini coefficient indicates less interference by differences between scenes, better generalization performance, and better suitability for application in cold-start scenes.

4.2 Experimental Results

The following part mainly introduces the experimental setup and analyzes the comparison among our SwAN model and other single-scene models (SSM) and static multi-scene models (SMSM) in the recommendation datasets of Meituan and Taobao.

SSM Experiments. This experiment is first based on classical SSM, including DNN [4], MMoE [15], and PLE [20]. DNN is a single-scene and single-task model. As shown in Table 2, our model has significantly improved AUC compared to DNN in both datasets, especially for the recommendation effect of new scenes in Meituan's dataset. MMoE and PLE are all single-scene and multi-task models. In this experiment, we added two objectives, click prediction and order prediction, for Dataset-1. In contrast, we conducted single-objective prediction for the Taobao dataset due to only one objective provided. In addition, to be consistent with the industrial application strategy, the SSM model uses all samples from various scenes for training and incorporates scene IDs as features. However, no multi-scene model structure optimization has been performed. The experimental results also prove that our model outperforms the baseline models in new and old scenes.

SMSM Experiments. To verify the effectiveness of SwAN compared to existing state-of-the-art SMSMs, we trained the HMoE [11], STAR [19], PEPNet [1] and HiNet [27] and then conducted comparative experiments. According to the definition mentioned earlier, both of these models belong to the static multi-scene model, which is suitable for multiple fixed scenes with stable traffic, and therefore contradicts the definition of dynamic multi-scene. To solve this problem, we adopted a standard solution in industrial applications: clustering scenes based on their attributes and treating the resulting cluster of new and old scenes as a sizeable stable scene.

Table 5: The experimental results of different cc threshold.

cc threshold	±0.1	±0.05	±0.01
Number of key features	4	13	68
AUC	0.7849	**0.7860**	0.7851

Table 6: Model under different α ($\beta = 0.001, \gamma = 0.001$).

	$\alpha = 2$	$\alpha = 1$	$\alpha = 0.5$
AUC	0.7859	**0.7860**	0.7839

Table 7: Model under different β ($\alpha = 1, \gamma = 0.001$).

	$\beta = 0.0001$	$\beta = 0.001$	$\beta = 0.01$	$\beta = 0.1$
AUC	0.7858	**0.7860**	0.7855	0.7850

Table 8: Model under different γ ($\alpha = 1, \beta = 0.001$).

	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$
AUC	0.7859	**0.7860**	0.7856	0.7852

The experimental results showed that SwAN still achieved the best performance.

Sub-scenes Experiments. In this context, a sub-scene refers to an individual scene within each source in Table 1. We randomly selected 10 sub-scenes from the test set and tested the effect comparison of different models, as shown in Table 4. It can be seen that SwAN achieves the highest AUC in each sub-scene.

In summary, SwAN has shown advantages in solving dynamic multi-scene problems compared to other widely-used single-scene and multi-scene models. This further proves that SwAN is theoretically practical and widely applicable.

4.3 Hyperparameters Experiments

To illustrate the impact of hyperparameters on the experimental results, we conducted relevant experiments based on dataset-1.

Hyperparameters of SRG. We tested the experimental results of constructing SRG by filtering features according to different correlation coefficients (cc) thresholds (Table.5). It can be found from the experimental results that a reasonable threshold can filter out noise features and select as many effective features as possible to improve the model performance. The empirical threshold is 0.05, which can also be experimented with and adjusted according to specific business data.

Hyperparameters of Loss Functions. Regarding the hyperparameter selection of loss functions, the experimental outcomes for α, β, and γ are presented in Table 6, Table 7, and Table 8 (Section.3.6 for details). Given that $Loss_{ce}$ plays a pivotal role in model optimization, it is advisable to set α to a relatively substantial value. Conversely, as both $Loss_{cos}$ and $Loss_{var}$ serve as auxiliary components in the training process, it is recommended to keep β and γ small-valued.

Hyperparameters of $Dics$. Based on the experimental findings for temperature coefficients τ of $Dics$ as shown in Table 10, it is evident that excessively high temperature coefficients result in a

Table 9: Ablation experiments without (w/o) structures based on Meituan's Dataset. The statistical significance of SwAN's performance improvement over its alternative versions has been validated by the Friedman test.

	SwAN	w/o SRG	w/o AEM	w/o CFR	w/o $Loss_{var}$	w/o $Loss_{cos}$
AUC of all scenes	**0.7860**	0.7805	0.7819	0.7840	0.7838	0.7841
AUC of cold-start scenes	**0.7799**	0.7732	0.7749	0.7771	0.7767	0.7774

Table 10: Model under different temperature coefficient τ.

	$\tau = 1$	$\tau = 0.1$	$\tau = 0.01$	$\tau = 0.001$	$\tau = 0.0001$
AUC	0.7821	0.7838	0.7854	0.7860	0.7860

Table 11: Model performance under different expert number of AEG ($N_s = 10$). The inference time refers to the time consumption of each sample, measured in milliseconds.

N_a	3	5	8	10	13	15
AUC	0.7802	0.7831	0.7845	0.7860	0.7863	0.7863
Inference time	0.1641	0.1670	0.1710	0.1741	0.1785	0.1814

Table 12: Model performance under different expert number of SEG ($N_a = 10$). The inference time refers to the time consumption of each sample, measured in milliseconds.

N_s	3	5	8	10	13	15
AUC	0.7815	0.7839	0.7849	0.7860	0.7862	0.7863
Inference time	0.1651	0.1677	0.1714	0.1741	0.1781	0.1809

decline in performance, leading to reduced diversity among various scenes in AEG. Conversely, referencing Eq. 8, overly small temperature coefficients introduce discontinuities in the curve of the *Dics* function, thereby destabilizing the training process. Hence, a temperature coefficient value of around 0.001 can be selected and appropriately increase the value of $Loss_{var}$.

Hyperparameters of AEM. The experiments regarding the number of experts in AEG (N_a) and SEG (N_s) can be found in Table 11 and Table 12. It's worth noting that the impact of adding experts is most pronounced when N_a and N_s are relatively small. However, as these values grow larger, the model's computational efficiency decreases, and the gains in performance become less significant. Therefore, it is advisable to strike a reasonable balance between effectiveness and efficiency.

4.4 Ablation Study and Analysis

The results of the ablation experiments are shown in Table 9. Firstly, we tested the impact of SRG on the model performance. SRG is mainly responsible for introducing prior knowledge of similar scenes to the model, which is the theoretical basis of SwAN. After removal, other affected structures must be randomly initialized and uniformly distributed. This structure significantly impacts the recommendation performance (the AUC decreased from 0.7860 to 0.7805 after removal). Secondly, we conducted ablation experiments on the AEM structure. This component is responsible for learning how to design the model structure and extracting exclusive and

shared information for each scene. The control group for this experiment used uniformly distributed experts instead of the original dynamic allocation in AEG. The results showed a certain degree of degradation in AUC (from 0.7860 to 0.7819). Thirdly, the CFR was masked, and the AUC dropped to 0.7840. Finally, we performed ablation experiments on $Loss_{var}$ and $Loss_{cos}$. After adding $Dics(\cdot)$, AEG can dynamically allocate experts, and $Loss_{var}$ further enhances the discrimination of the allocation. The experimental results showed that removing the two types of losses decreased 22 BP (Basic Point) and 19 BP in the model AUC, respectively.

In the research field of recommendation, improving a model to achieve higher AUC than state-of-the-art recommendation approaches is generally recognized to be highly challenging in practice [19, 25, 26]. The comparative experiments in this paper were repeated 10 independent times for validation, and the results were statistically significant at the 0.05 level (Friedman test), indicating confidence in the effectiveness of SwAN.

4.5 Application in Practice

SwAN has been deployed in the online recommendation system to validate its practical effectiveness and component efficacy.

Online Application Performance. Besides the importance in theory, recommendation plays a pivotal role in commercial situations. To further demonstrate the superior performance of SwAN in practical applications, we deployed it in the online recommendation system of Meituan's catering business with an average daily user level of millions, which has typical dynamic multi-scene characteristics, and conducted an A/B test with 20% of the traffic over a period of two months. More than 200 new online scenes were added during the experiment, and the total number exceeded 400. The experimental results showed that SwAN achieved a 5.64% increase in the CTR index compared to the best baseline model (PLE) and a 5.19% increase in daily order volume proportion after full traffic promotion.
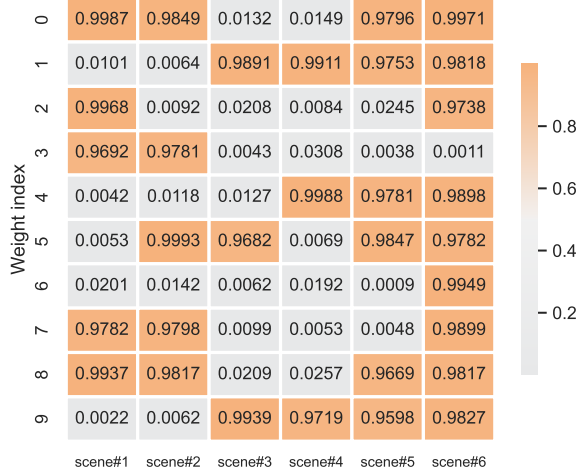
In addition, we randomly selected 6 new scenes online and calculated the CTR improvements of both SwAN and baseline models relative to the default ranking method (Table 13):

First, the baseline models and SwAN have significant CTR enhancements relative to the default ranking. However, SwAN has a smaller Gini coefficient for the improvement ratio between scenes, demonstrating superior stability. We also calculated the Gini coefficient of the improvement ratio of SwAN relative to the baseline model in each scene, which is 0.2651 (0.2351 in all scenes). This value proves that the improvement of SwAN relative to the baseline model is evenly distributed among different scenes instead of only focusing on large scenes and ignoring small ones.

Second, combining each scene's daily exposure samples, we found that SwAN has a more significant improvement ratio than

Table 13: Online Experimental Results.

	Sub Scene #1	Sub Scene #2	Sub Scene #3	Sub Scene #4	Sub Scene #5	Sub Scene #6	Gini
#Exposure (1 day)	16K	33K	20K	65K	7K	48K	-
#CTR	21.97%	9.46%	0.36%	1.39%	27.18%	0.72%	-
Baseline / Default	+50.60%	+47.20%	+38.19%	+53.51%	+33.35%	+46.53%	0.0858
Ours / Default	+53.91%	+49.69%	+41.56%	+55.07%	+37.70%	+48.57%	0.0727
Ours / Baseline	+6.55%	+5.29%	+8.83%	+2.92%	+13.07%	+4.40%	0.2651

Fig 6: W_k (10 selected) in the AEG calculated from samples of 6 randomly chosen scenes. Each column represents 10 W_k calculated from a sample.

the baseline model in small scenes, demonstrating its better generalization ability and competence to optimize cold-start problems for scenes with sparse user behaviors.

Visualization of the Expert Selector. To demonstrate the effectiveness of information transfer in Expert Selector, we randomly selected 6 scenes and one sample from each scene. The W_k values in AEG calculated from these samples are shown in Fig. 6.

Firstly, it can be seen that the expert selection in different scenes is significantly different, indicating that Expert Selector can distinguish between different scenes. Secondly, from the figure, it can be observed that scene#1 and scene#2 exhibit a high degree of similarity in the selection of experts. Upon examining the actual scene data, we find that both scenes primarily focus on selling afternoon tea. This further confirms that the selection of experts is related to the similarity of the actual scenes. Finally, it can be seen that the model increases the number of selected experts in scene#6 due to the diverse item types, which strengthens its generalization performance.

4.6 Model Complexity

All experiments were conducted on NVIDIA Tesla A100 GPU, 80G RAM, and Intel(R) Xeon(R) Gold 5218 CPU servers. Table 14 shows that SwAN maintains a reasonable number of parameters and prediction time. Generally, we retrieve online data in real-time for training and update the model approximately every half hour.

Table 14: Time and space complexity of models on the Meituan dataset. The inference time refers to the time consumption of each sample, measured in milliseconds.

Model	Params ($\times 10^7$)	Training time	Inference time
DNN	1.90	142 mins	0.1389
MMoE	1.99	164 mins	0.1442
PLE	2.06	176 mins	0.1675
HMoE	2.06	168 mins	0.1739
STAR	2.18	178 mins	0.1808
PEPNet	2.39	188 mins	0.1889
HiNet	2.27	182 mins	0.1865
SwAN	2.21	173 mins	0.1741

5 Conclusion

In this paper, we propose the SwAN model, a novel approach to addressing the cold-start problem in Multi-scene Recommendation (MSR) systems. The proposed model overcomes the limitations of traditional MSR approaches by directly and significantly enhancing the performance of newly-arrived scenes through online prediction. The unique architecture of the SwAN model, which combines the Scene Relation Graph (SRG), Similarity Attention Network (SAN), and Adaptive Ensemble-experts Module (AEM), enables it to capture graph-structured similarities between scenes, understand user behavior transitions, and identify shared logic among different scenes. Our extensive evaluation of SwAN on both public and Meituan industrial datasets demonstrate the superiority of the SwAN model over existing MSR approaches, providing more accurate and high-quality recommendations in a dynamic and adaptable manner. Furthermore, SwAN has been deployed in the online catering recommendation service of Meituan, which serves millions of daily customers, and has achieved a significant improvement in CTR (Click-Through Rate) index. This work represents a significant step forward in developing efficient and adaptable recommendation systems, particularly in the context of a rapidly evolving E-commerce landscape.

Acknowledgments

References

[1] Jianxin Chang, Chenbin Zhang, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. 2023. Pepnet: Parameter and embedding personalized network for infusing with personalized prior information. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3795–3804.

[2] Yuting Chen, Yanshi Wang, Yabo Ni, An-Xiang Zeng, and Lanfen Lin. 2020. Scenario-aware and Mutual-based approach for Multi-scenario Recommendation in E-Commerce. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 127–135.

[3] 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.

[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[5] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Sequential scenario-specific meta learner for online recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2895–2904.

[6] Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 133–138.

[7] Yulong Gu, Wentian Bao, Dan Ou, Xiang Li, Baoliang Cui, Biyu Ma, Haikuan Huang, Qingwen Liu, and Xiaoyi Zeng. 2021. Self-Supervised Learning on Users' Spontaneous Behaviors for Multi-Scenario Ranking in E-commerce. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3828–3837.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[9] Yang Hu, Adriane Chapman, Guihua Wen, and Dame Wendy Hall. 2022. What can knowledge bring to machine learning?—a survey of low-shot learning for structured data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 3 (2022), 1–45.

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

[11] 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.

[12] Wenhao Li, Haiou Zhang, Guilan Wang, Gang Xiong, Meihua Zhao, Guokuan Li, and Runsheng Li. 2023. Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. *Robotics and Computer-Integrated Manufacturing* 80 (2023), 102470.

[13] 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.

[14] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[15] 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.

[16] Kiran Rama, Pradeep Kumar, and Bharat Bhasker. 2019. Deep learning to address candidate generation and cold start challenges in recommender systems: A research survey. *arXiv preprint arXiv:1907.08674* (2019).

[17] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.

[18] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. *The adaptive web: methods and strategies of web personalization* (2007), 291–324.

[19] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4104–4113.

[20] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 269–278.

[21] 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.

[22] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*. 1785–1797.

[23] Zheni Zeng, Chaojun Xiao, Yuan Yao, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2021. Knowledge transfer via pre-training for recommendation: A review and prospect. *Frontiers in big Data* 4 (2021), 602071.

[24] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.

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

[26] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[27] Jie Zhou, Xianshuai Cao, Wenhao Li, Lin Bo, Kun Zhang, Chuan Luo, and Qian Yu. 2023. Hinet: Novel multi-scenario & multi-task learning with hierarchical information extraction. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2969–2975.

[28] Jie Zhou, Qian Yu, Chuan Luo, and Jing Zhang. 2023. Feature decomposition for reducing negative transfer: a novel multi-task learning method for recommender system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37.

[29] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696* (2021).

[30] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1507–1515.

[31] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1167–1176.