SAR-Net: A Scenario-Aware Ranking Network for Personalized Fair Recommendation in Hundreds of Travel Scenarios

Qijie Shen
Alibaba Group
Hangzhou, China
qijie.sqj@alibaba-inc.com

Wanjie Tao
Alibaba Group
Hangzhou, China
wanjie.twj@alibaba-inc.com

Jing Zhang
The University of Sydney
Darlington NSW 2008, Australia
jing.zhang1@sydney.edu.au

Hong Wen
Alibaba Group
Hangzhou, China
qinggan.wh@alibaba-inc.com

Zulong Chen
Alibaba Group
Hangzhou, China
zulong.czl@alibaba-inc.com

Quan Lu
Alibaba Group
Hangzhou, China
luquan.lq@alibaba-inc.com

ABSTRACT

The travel marketing platform of Alibaba serves an indispensable role for hundreds of different travel scenarios from Fliggy, Taobao, Alipay apps, etc. To provide personalized recommendation service for users visiting different scenarios, there are two critical issues to be carefully addressed. First, since the traffic characteristics of different scenarios, e.g., individual data scale or representative topic, are significantly different, it is very challenging to train a unified model to serve all. Second, during the promotion period, the exposure of some specific items will be re-weighted due to manual intervention, resulting in biased logs, which will degrade the ranking model trained using these biased data. In this paper, we propose a novel Scenario-Aware Ranking Network (SAR-Net) to address these issues. SAR-Net harvests the abundant data from different scenarios by learning users' cross-scenario interests via two specific attention modules, which leverage the scenario features and item features to modulate the user behavior features, respectively. Then, taking the encoded features of previous module as input, a scenario-specific linear transformation layer is adopted to further extract scenario-specific features, followed by two groups of debias expert networks, i.e., scenario-specific experts and scenario-shared experts. They output intermediate results independently, which are further fused into the final result by a multi-scenario gating module. In addition, to mitigate the data fairness issue caused by manual intervention, we propose the concept of Fairness Coefficient (FC) to measures the importance of individual sample and use it to reweigh the prediction in the debias expert networks. Experiments on an offline dataset covering over 80 million users and 1.55 million travel items and an online A/B test demonstrate the effectiveness of our SAR-Net and its superiority over state-of-the-art methods. SAR-Net has also been deployed in the online travel marketing platform of Alibaba and is serving hundreds of travel scenarios.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommender system, Click-through rate prediction, Scenario-aware, Fairness coefficient

ACM Reference Format:
Qijie Shen, Wanjie Tao, Jing Zhang, Hong Wen, Zulong Chen, and Quan Lu. 2021. SAR-Net: A Scenario-Aware Ranking Network for Personalized Fair Recommendation in Hundreds of Travel Scenarios. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3481948

1 INTRODUCTION

In recent years, Recommender Systems (RS) have played an increasingly important role in e-commerce platforms [26, 35, 36, 40]. It not only boosts businesses via traffic utilization, but also greatly reduces the time cost for users to find items of interest. There are two main phases in a typical e-commerce RS, i.e., *Matching* and *Ranking*, where *Matching* is able to retrieve several thousands of candidate items from hundreds of millions of items, while *Ranking* is responsible for sorting these candidates according to specific metrics, such as click-through rate (CTR) [7, 32, 37] or conversion rate prediction (CVR) [18, 34–36]. Sometimes, after *Ranking*, there is an additional phase called *Reranking*, which is sometimes utilized to adjust the ranking results based on some manually defined rules, especially during the promotion period, such as the Double-Eleven Shopping Festival in China.

As a specific instance of RS, the Alibaba travel marketing platform aims to provide personalized recommendation service to users from hundreds of travel scenarios in Fliggy, Alipay, and Taobao apps. For providing preeminent service for users visiting different travel scenarios, there are two critical issues encountered in practice. On the one hand, due to the individual topic or data scale for each scenario, the data distribution among these travel scenarios is significantly different, resulting in the difficulty of training a unified model to serve all. We call it as the *Multi-Scenario Modeling Issue*. On the other hand, to ensure the definiteness of traffic for certain important merchants and items during the promotion period, there is always manual intervention to adjust the ranking results in the *Reranking* phase, which will make the real exposure traffic be

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481948

biased towards those intervened merchants or items. Consequently, when training with the biased logs, the model in *Ranking* phase will inevitably learn more information towards these overexposed intervened items, resulting in unexpected self-reinforcement in the ranked results, i.e., systematic discrimination of disadvantaged items. We call it as the *Data Fairness Issue*.

For the *Multi-Scenario Modeling Issue*, there are typically three kinds of solutions: 1) training individual ranking model for each scenario [42, 43]; 2) training a unified ranking model with all scenario data; and 3) using a multi-task learning framework to output multiple prediction results simultaneously [4, 17, 20], and each result for each scenario. However, the first method has obvious shortcomings. First, as the number of scenarios increases, maintaining individual model for each scenario requires a huge amount of cost. Second, since the data scale from certain scenarios is small, it is difficult to train an excellent ranking model only leveraging data from their own data. Alternatively, the second type of methods try to utilize the data of all scenarios. However, since the traffic characteristics of different scenarios are significantly different, it is very challenging to train a unified model that performs well in all scenarios. The third type of methods focus on the multi-task learning approaches while trying to harvest all data by discovering the explicit relationships among different scenarios. And existing multi-task learning approaches adopt an early-sharing strategy by learning a shared feature embeddings among different tasks, followed by feeding them into individual task-specific sub-network, respectively. However, the traditional multi-task modeling methods ignore the modeling of user interest transfer across scenarios, which consequently are not able to predict user interest accurately. In addition, the importance of input information from different scenarios is inevitably different while previous methods do not capture it explicitly.

For the *Data Fairness Issue*, recent related studies on the bias issue mainly focus on exposure bias [22, 31, 33], selection bias [21, 27], etc. However, to the best of our knowledge, there is no previous research on intervention bias in recommender systems. A straightforward way to handle this problem is to down-sampling the overexposed items caused by intervention. However, this method requires too much engineering tricks and manpower to manually adjust the data set, which are not applicable due to the frequently changing intervention rules, especially during promotion period. Additionally, over down-sampling the data set will inevitably lead to serious wastage of usable data.

To address the above two issues, we propose a Scenario-Aware Ranking Network (SAR-Net) in this paper. The network structure is based on a multi-expert network. Two attention modules are employed to extract the user's cross-scenario interest considering the scenario features and item features, respectively. A scenario-wise linear transformation is devised to strengthen the important information for each individual scenario. The linear transformation uses an element-wise operation, which almost does not increase the overall parameters of the model, but can leverage the differences and commonness between scenarios. To address the data fairness issue caused by manual intervention, we propose a Fairness Coefficient to measure the importance of samples in the scenario. It acts as the weight of the sample in the loss function and a useful feature in the bias-expert net of the expert network, which will be removed when deployed online. In this way, the proposed SAR-Net can not only fully learn the differences and relationship between scenarios, but also reduce the impact of intervention bias.

The main contributions of this work are summarized as follows:

- We propose a novel SAR-Net that can predict users' cross-scenario interest given the scenario features and item features and extract important scenario-specific information across different scenarios adaptively.
- We investigate the data fairness problem caused by manual intervention in recommender systems and propose a simple yet effective solution through the design of network structure and loss function.
- Evaluation on both the offline dataset and online A/B test demonstrate the superiority of the proposed SAR-Net over representative methods. SAR-Net has been serving all travel scenarios of Alibaba and brought more than 5% CTR increase.

2 RELATED WORK

Our proposed method specifically tackles the multi-scenario prediction problem and data fairness problem, so we briefly review the most related work from the following aspects: 1) Single-Scenario CTR Prediction, 2)Multi-Task Learning, 3)Bias and Unfairness in Recommender System.

2.1 Single-Scenario CTR Prediction

Existing CTR prediction works mainly focus on single scenario modeling from the following several aspects: 1) feature interaction (e.g., FM [24], deepFM [10]); 2) user historical behavior (e.g., DIN[43], DIEN [42]); and 3) combining matching and ranking (e.g., DMR [16]).

Factorization Machine (FM) is proposed to model feature interactions explicitly, while previous generalized linear models such as Logistic Regression (LR) [25] and Follow-The-Regularized-Leader (FTRL) [19] lack the ability to solve interaction issue. Wide&Deep [6] and DeepFM [10] combine wide part (low-order) and deep part (high-order) features to improve the performance. FmFM [29] makes each field feature have different embedding dimensions, so as to reduce the amount of model parameters and avoid over fitting problem. DIN [43] utilizes the attention mechanism to capture relative interests from the user behavior sequence with regard to the candidate item. DIEN [42] further uses a GRU structure to capture the evolution of user interest. Considering a single vector might be insufficient to capture complicated user patterns, DMIN [38] models user's multiple interests by a special designed extractor layer. DSIN [9] introduces a hierarchical view of behavior sequence by dividing it into sessions. DMR [16] considers the relevance between user and item to achieve better performance.

2.2 Multi-Task Learning

Multi-Task Learning (MTL) [4] aims to improve generalization by sharing knowledge across multiple related tasks. The shared knowledge and task-specific knowledge are explored to facilitate the learning of each task. There have been some studies applying the gate structure and attention network for information fusion. MOE [11] has a shared-bottom model structure, where the bottom hidden layers are shared across tasks. MMOE [17] extends MOE to utilize different gates network for each task to obtain different fusion weights in MTL. Similarly, MRAN [41] applies multi-head self-attention to learn different representation subspaces at different feature sets. Cross-Stitch [20] uses linear cross-stitch units to learn an optimal combination of task-specific representations. To address the seesaw phenomenon, PLE [30] separates shared components and task-specific components explicitly and adopts a progressive routing mechanism to extract and separate deeper semantic knowledge gradually.

2.3 Bias and Unfairness in RS

User behavior data are observational rather than experimental. leading to various biases in the data, such as exposure bias, popularity bias, and unfairness bias [5]. Blindly fitting the data without considering the inherent biases will cause many serious issues, e.g., the discrepancy between offline and online performance, and reducing user's satisfaction and trust on the recommendation service. Exposure bias happens as users are only exposed to a part of specific items so that unobserved interactions do not always represent negative preference. Popularity bias can be explained as popular items are recommended even more frequently than their popularity would warrant [1]. Unfairness bias can be explained as the system systematically and unfairly discriminates against certain individuals or groups of individuals in favor others [3, 8, 12, 15, 28].

The intervention bias investigated in this paper shares similarity with unfairness bias, both of which are caused by the unbalanced data distribution. Consequently, a model trained on the data is biased. The difference is that unfairness bias refers to the imbalance of users, which misleads the model to lean towards the interests of specific users, which will affect the performance for long-tail users. The intervention bias causes the data fairness issue where external manual intervention adjusts the ranking results, biasing the model towards certain items or merchants. Trained on such unbalanced data, the model will overfit over-represented items and reinforce them in the ranked results, resulting in a systematic discrimination that reduces the visibility of disadvantaged items.

3 PROBLEM FORMULATION

In a recommender system, the user-item interaction is typically formulated as a matrix $Y = \{y_{ui}\}_{M\times N}$, where M and N denote the numbers of users and items, respectively. The interaction y is either implicit feedback [2], e.g., click or explicit user rating [14]. In this work, we focus on the CTR prediction task, implying that the matrix Y consists of 0 and 1. Specifically, $y_{uis} = 1$ means that user u has clicked item i in scenario s, otherwise $y_{uis} = 0$. Moreover, each interaction is associated with a timestamp t that records the time of interaction. Therefore, the data in the recommender system are denoted by a set of quintuplets $\Gamma = \{(u, i, s, t, y)\}$, each of which includes the user $u \in U$ interacts with an item $i \in I$ in scenario s at a recommendation time t. For a target quadruple (u, i, s, t), an interaction probability should be predicted.

In this work, the CTR prediction model aims to estimate the probability of interaction $\hat{y}$ between a target user $u \in U$ and a candidate item $i \in I$ in scenario s, with the consideration of the scenario context and item context.

4 MODEL DESIGN

In this section we will detailedly present the SAR-Net, with its overall architecture illustrated in Figure 1. SAR-Net takes cross-scenario user behaviors, user basic profiles, contextual scenario features, and target item as input. It firstly embeds these input features as low-dimensional vectors by an embedding layer. Then, it extracts user cross-scenario interest transfer from users' historical behaviors by devising a Cross-Scenario Behavior Extract Layer after considering the scenario features and item features. Next, taking the encoded features of previous module as input, a scenario-wise linear transformation layer is adopted to strengthen the important information for each individual scenario. Finally, the Mixture of Debias Experts, i.e., scenario-specific experts and scenario-shared experts, output intermediate results independently, which are further fused into the final result by a multi-scenario gating module. Now, we will introduce each module in detail.

4.1 Fairness Coefficient

Intervention bias makes the data distribution biased towards the weighted items. When training on such unbalanced data, the recommendation models will inevitably tend to learn more about these over-represented items, resulting in unexpected self-reinforcement in the ranked results, i.e., systematic discrimination of disadvantaged items. Therefore, to address this issue, we propose the concept of Fairness Coefficient (*FC*), which measures the importance of individual sample and represents the degree of intervention of different items.

Assuming $\mathbb{D}$, $d_{i,s}$, N_s denotes the whole data of all scenarios in one day, the partial data of item i exposing in scenario s, the number of items in scenario s, respectively, where $d_{i,s} \in \mathbb{D}$. Then, we define $PV(i, s)$ and $F(i, s)$ as the number of samples from $d_{i,s}$, the sum of the values predicted by the proposed SAR-Net for all samples from $d_{i,s}$ with an offline manner, respectively. Therefore, the Fairness Coefficient, denoted as $w^{i,s}_{fairness}$, indicating the degree of intervention of different items in different scenarios, defined as follows:

$$w^{i,s}_{fairness} = \frac{F(i,s)/\sum_{i=0}^{N_s} F(i,s)}{PV(i,s)/\sum_{i=0}^{N_s} PV(i,s)}. \tag{1}$$

Where the numerator is a constant, which has no relationship with whether items intervened, while denominator affected by intervention. Therefore, as the increment of exposure volumes caused by intervention, the *FC* will become smaller, vice versa. *FC* will act as the weight of the sample in the loss function and a useful feature in the bias-expert net of the expert network.

4.2 Embedding Layer

There are five groups of features, i.e., user profiles, user cross-scenario behavior, scenario context feature, target item, and intervention bias. User profile contains features related to the user, e.g., user id, country, etc. Target item feature refers to the candidate item with corresponding features such as item id, category id, statistical offline scores, etc. Scenario context feature is a group of features including but not limited to time, current scenario id, current scenario type, etc. User cross-scenario behavior, with behavior type clicking, purchasing or add-to-cart, is a list of user interacted items in all

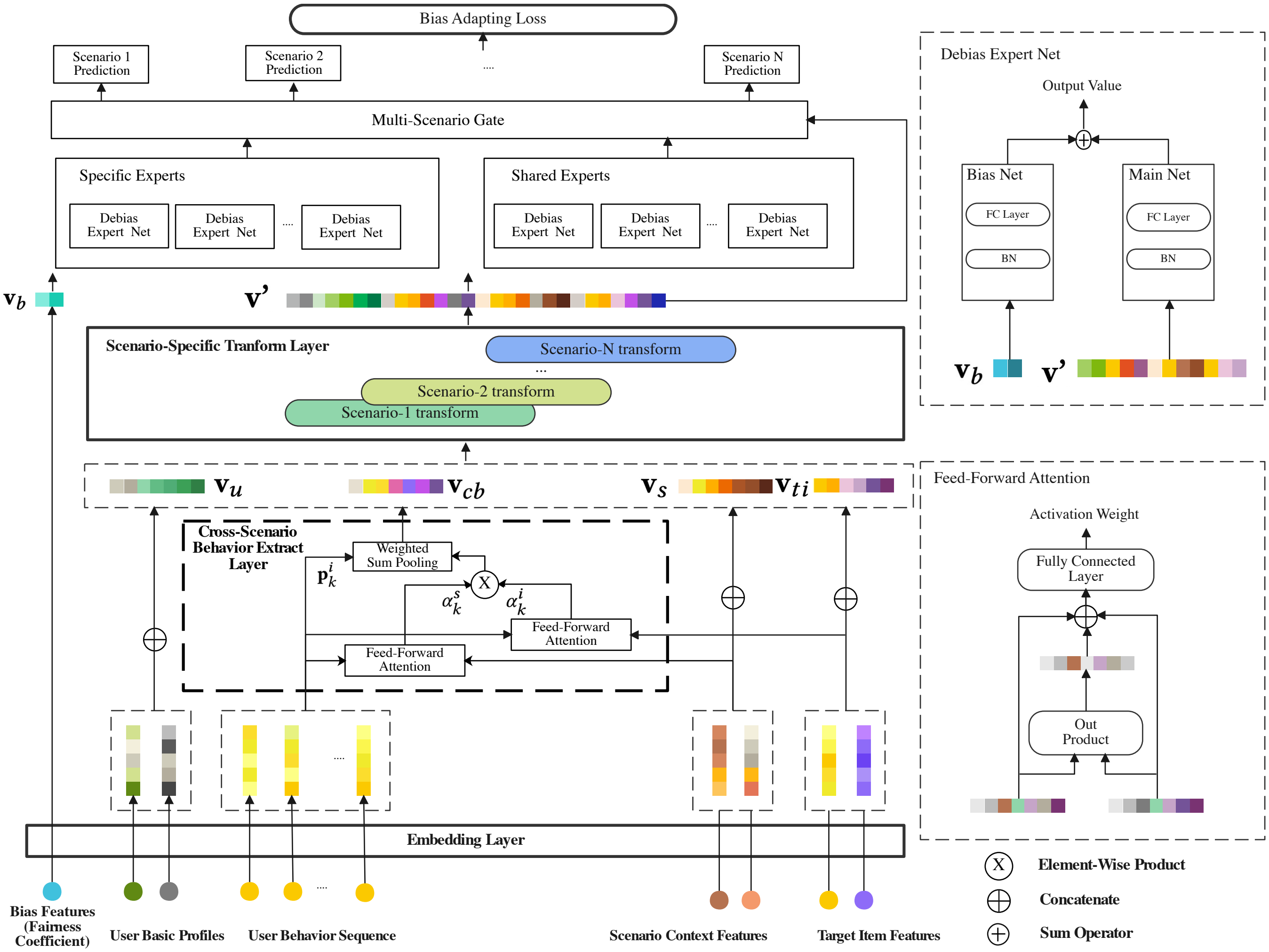

Figure 1: The overview architecture of our proposed model SAR-Net. Cross-Scenario Behavior Extract Layer harvests the abundant data from different scenarios by learning users' cross-scenario interest via two remarkable attention modules. Then, a Scenario-Specific Transform Layer is adopted to further extract scenario-specific information, followed by two groups of debias expert networks. We use a Fairness Coefficient to measure the importance of individual sample and use it to reweigh the prediction in the debias expert networks. A multi-scenario gating module is used to fuse these predictions into the final one.

scenarios, where each item in this list not only has same feature fields as the target item but also has the scenario context features at the moment that the behavior happened. Intervention bias feature is the Fairness Coefficient of each sample which is computed after one-day data is generated. Each feature can be encoded into an one-hot vector with high-dimension.

We first encode features into one-hot encodings. For the ith feature, its one-hot encoding is denoted as:

$$\mathrm{v}_i = onehot(i), \tag{2}$$

where $\mathrm{v}_i \in \mathbb{R}^N$ is a vector with 1 at the ith entry and 0 elsewhere, and N is the number of unique features. We then map the sparse and high-dimensional one-hot encodings to dense and low-dimensional embedding vectors that are suitable for neural networks. In particular, we define a learnable embedding matrix $\mathbf{E} \in \mathbb{R}^{D \times N}$, where $D \ll N$ is the dimension. The ith feature is then projected to its corresponding embedding vector $\mathbf{e}_i \in \mathbb{R}^D$ as:

$$\mathbf{e}_i = \mathbf{E}v_i. \tag{3}$$

4.3 Cross-Scenario Behavior Extract Layer

Existing multi-task modeling methods do not consider the user's interest transfer in different scenarios. In fact, most users reside in more than one scenario, with different preference for respective scenario. For example, users prefer items such as tickets of scenic spots in the travel-surrounding theme scenario, while prefer items with higher living quality or with a place suitable for children to play in the parent-child theme scenario. And in the couple-travel theme scenario, romantic scenery and travel experience will become the primary consideration. Therefore, for reflecting user's significant travel intention for different scenarios and portraying user's interest transfer, the modeling for user's cross-scenario behaviors is very critical.

With this in mind, we can aggregate user's cross-scenario behaviors into a unified representation. In general, the aggregated strategy can be defined as:

$$a_s = \sum_i \alpha_i x_i, \tag{4}$$

where α_i is a weight assigned to x_i, indicating its importance during aggregation. The remaining issue is how to compute the weight. A naive way is $\alpha_i = 1/|x_i|$, i.e., each one of clicked items has equal importance. It is clearly not a wise choice because some items may not be indicative for the target item. Inspired by this insight, DIN [43] applies the attention mechanism to extract relevant interests from user's behaviors, which considers the degree of correlation between historical behavioral items and target items. However, in multi-scenario modeling, the scenario context information at the moment that the behavior happened also carry a lot of information. For example, a user's historical behaviors in three themed scenarios, namely travel-surrounding, parent-child travel, and northwestern travel, provide informative context when users are looking for items they are interested in at the couple-travel theme scenario. The behavioral mentality of users in different scenarios in the past is different from that of the current scenario. Among them, the relevance of parent-child travel is weaker, therefore we need to consider the user's behavior in strong relevance scenarios to make recommendations. Specifically, we extract users' cross-scenario interests via two specific attention modules, which leverage the scenario features and item features to modulate the user behavior features, respectively.

Furthermore, a user's cross-scenario behavior can be split into two parts, i.e., item behavior sequence $\mathbf{p}(B^i) = \{\mathbf{p}_1^i, \mathbf{p}_2^i, \cdots, \mathbf{p}^i_{|\mathbf{p}(B^i)|}\}$ and scenario context sequence $\mathbf{p}(B^s) = \{\mathbf{p}_1^s, \mathbf{p}_2^s, \cdots, \mathbf{p}^i_{|\mathbf{p}(B^s)|}\}$, where $\mathbf{p}_k^i$ is obtained by concatenating the kth corresponding item feature embedding vectors, including item id, category, destination, etc, i.e., $\mathbf{p}_k^i = [\mathbf{e}_{itemId}||\mathbf{e}_{destination}||\mathbf{e}_{category}||\cdots]$. $\mathbf{p}_k^s$ is obtained by concatenating the kth corresponding scenario context feature embedding vectors, including scenario id, scenario type, behavior time, etc, i.e., $\mathbf{p}_k^s = [\mathbf{e}_{scenarioId}||\mathbf{e}_{scenarioType}||\mathbf{e}_{behaviorTime}||\cdots]$, where || is the vector concatenation operator.

Additionally, we define α_k^i and α_k^s , indicating the relevance between user's kth behavior item and the target item or target scenario, respectively, shown as follows.

$$\alpha_k^i = \frac{\exp(\psi(\mathbf{p}_k^i, \mathbf{p}_t^i))}{\sum_{l=1}^{|\mathbf{p}(B^i)|} \exp(\psi(\mathbf{p}_l^i, \mathbf{p}_t^i))}., \tag{5}$$

$$\alpha_k^s = \frac{\exp(\psi(\mathbf{p}_k^s, \mathbf{p}_t^s))}{\sum_{l=1}^{|\mathbf{p}(B^s)|} \exp(\psi(\mathbf{p}_l^s, \mathbf{p}_t^s))}, \tag{6}$$

Where, $\mathbf{p}_t^i$ and $\mathbf{p}_t^s$ representing the embeddings of target item and target scenario, respectively. And $\psi(x, y)$, taking two vectors x and y as input, output the weight value by employing the feed-forward attention operator, illustrated in Figure 1.

Finally, we aggregate $\mathbf{p}_k^i$ with the consideration of both α_k^i and α_k^s as follows to get the user cross-scenario interest transfer $\mathbf{v}_{cb}$.

$$\mathbf{v}_{cb} = \sum_{k=1}^{t} \alpha_k^i * \alpha_k^s * \mathbf{p}_k^i. \tag{7}$$

4.4 Scenario-Specific Transform Layer

After generating user interest transfer vector $\mathbf{v}_{cb}$, user basic profiles vector $\mathbf{v}_u$, target item feature vector $\mathbf{v}_{ti}$ and the scenario context feature vector $\mathbf{v}_s$, we can obtain the gathered representation $\mathbf{v}$ by concatenating the corresponding feature embedding vectors, i.e., $\mathbf{v} = [\mathbf{v}_{cb}||\mathbf{v}_u||\mathbf{v}_{ti}||\mathbf{v}_s]$. Next, for further extracting scenario-aware specific information, we apply a scenario-wise transform module to process the previous representation $\mathbf{v}$. Specifically, for the ith scenario, we compute **v'** as follows:

$$\mathbf{v'} = \mathbf{v} \otimes \beta_i + \gamma_i. \tag{8}$$

where, the vectors β_i and γ_i are scenario-aware parameters, which have the same dimension with $\mathbf{v}$, and $\otimes$ is an element-wise operator.

4.5 Mixture of Debias Experts

Because sharing the parameters among different tasks is difficult to describe the heterogeneity of different tasks and may potentially result in the negative transfer issue, we use a multi-expert network as the core structure of the feature extraction part. In order to further model the difference between homogeneous tasks, each scenario has some scenario-specific experts and all the scenarios share several common experts. Compared with the shared-only experts structure, our network alleviates the seesaw phenomenon, i.e., the model has a profit in one of the tasks but a negative profit in the other task.

In order to alleviate the influence of intervention bias on model prediction, we divide each expert network into two parts: Bias net and Main net. Both modules are composed of a fully connected network and a batch normalization layer. Main net takes user interest transfer vector $\mathbf{v}'_{cb}$, user basic profiles vector $\mathbf{v}'_u$, target item feature vector $\mathbf{v}'_{ti}$, and scenario context feature vector $\mathbf{v}'_s$ as input, and aims to predict the click-through rate of users on the target item. Bias net receives the input of Fairness Coefficient $\mathbf{v}_b$, taking one specific value from set w according to scenario-aware and item-aware principle, and predicts the degree of intervention bias to reweigh the predicted score of main net. Bias net is used only during training and will be removed when deployed online.

4.6 Multi-Gate Network & Prediction

After the Mixture of Debias Experts, we have the predicted scores from both scenario-specific experts and scenario-shared experts. Denoting x as the input representation, m_k as the quality of scenario k's scenario-specific experts, and m_s as the quality of scenario-shared experts, we have:

$$S^k(x) = [o_{k,1}, o_{k,2}, \cdots, o_{k,m_k}, o_{s,1}, o_{s,2}, \cdots, o_{s,m_s}]^T. \tag{9}$$

The structure of the multi-gate network is based on a single-layer feed-forward network with a SoftMax activation function. It acts as a selector to calculate the weighted sum of the selected predicting scores. More precisely, the output of scenario k's gating network is formulated as follows:

$$y^k(x) = w^k(x)S^k(x), \tag{10}$$

which is indeed the final predicted score of scenario k.

4.7 Bias Adapting Loss

To mitigate the intervention bias issue, we propose the concept of Fairness Coefficient to measure the importance of each sample and hope that the proposed model can get more information from samples with high Fairness Coefficients, due to the fact that these type of samples are less intervened. Furthermore, we use the binary cross entropy with the consideration of Fairness Coefficient, defined as below:

$$Loss = \sum_{l=1}^{|L_k|} w_{fairness}^{i,k} * loss_{k,l,i}, \tag{11}$$

where $loss_{k,l,i}$ is lth sample of scenario k and item i. $w_{fairness}^{i,k}$ is the Fairness Coefficient of scenario k and item i. $loss_{k,l,i}$ is computed as follows:

$$loss_{k,l,i} = -I_{k,l,i} \log p_{k,l,i} - (1 - I_{k,l,i}) log(1 - p_{k,l,i}), \tag{12}$$

where $p_{k,l,i}$ is the output of SAR-Net, $I_{k,l,i}$ is the label of lth sample of scenario k and item i. $I_{k,l,i} = 1$ indicates the current user will click the recommended item and 0 otherwise.

5 EXPERIMENTS

To comprehensively evaluate the proposed SAR-Net, we conduct experiments to answer the following research questions:
RQ1: How does SAR-Net perform compared with state-of-the-art models for multi-task CTR prediction?
RQ2: How about training a single model for each scenario using its own data or training a unified model for all scenarios using all data compared with the proposed SAR-Net?
RQ3: How about the impact of each part on the overall model?

5.1 Experimental Setups

5.1.1 Datasets. Due to the lack of public datasets for the multi-scenario CTR prediction task, we use Alibaba production data containing user click behaviors on 20 scenarios to perform the offline evaluation. The dataset contains users' logs from the travel platform of Alibaba in one month, which is collected from October 20th to November 20th, 2020, with the Double-Eleven Shopping Festival, one of the most important annual festivals of Alibaba Group, included during this period. The dataset is further organized into the training dataset and the testing dataset. The training dataset covers over 80 million users and 1.55 million travel items. Table 1 shows the percentage of training dataset and average CTR of each scenario. Table 2 shows the percentage of users that visited certain number of scenarios in past 30 days. It is obvious that different scenario has different distribution, and nearly 90% of users have visited multiple scenarios in past 30 days.

5.1.2 Competitors. We will compare our SAR-Net with multi-task models and single-scenario models. Since existing multi-task models are usually to model different tasks for the same scenario, we will adapt them to the multi-scenario CTR prediction task in this paper to model the same task (CTR) for different scenarios.

Specifically, **Multi-Task Models** include:

1) **Hard Parameter-Sharing**: HPS [4] is the most basic and commonly used MTL structure, where the parameters are straightforwardly shared between different tasks.
2) **Cross-Stitch**: Cross-Stitch [20] proposes to learn static linear combinations to fuse representations of different tasks.
3) **MMOE**: MMOE [17] applies gating networks to combine bottom experts based on the input to handle the differences between tasks.
4) **CGC**: Compared with MMOE, CGC [30] separates the expert layer into shared experts and unique experts, enabling different types of experts to concentrate on learning different knowledge efficiently without interference.
5) **PLE**: Compared with CGC, PLE [30] adopts a progressive routing mechanism to extract and separate deeper semantic knowledge gradually.

and **Single-Scenario Models** include:

1) **Wide&Deep**: Wide&Deep [6] combines LR (wide part) and DNN (deep part).
2) **PNN**: PNN [23] automatically learns feature representations and high-order feature interactions.
3) **DIN**: Deep Interest Network [43] models dynamic user interest based on historical behavior for CTR prediction.

5.1.3 Parameter Settings. Adam [13] is used as the optimizer with the learning rate of 0.001 for all methods and the batch size is 2048. For all methods, the truncation length of user behavior is 50. For SAR-Net, each scenario has 2 specific experts, and 8 experts are shared for each scenario. DIN, PNN and Wide&Deep use single-scenario data and mix-scenario data for training respectively. We run each method 10 times and report the average results.

5.1.4 Metrics. (1) **AUC**: AUC denotes the Area Under the ROC Curve over the test set. It is a widely used metric for CTR prediction, which reflects the probability that a model ranks a randomly chosen positive instance higher than a randomly chosen negative instance. The larger AUC is, the better the CTR prediction model performs. It is noteworthy that a small improvement in AUC is likely to lead to a significant increase in online CTR [43]. Concretely, we use the AUC of each scenario and overall AUC (mixing samples from all scenarios to calculate the overall AUC) as the metrics. (2) **RelaImpr**: RelaImpr is introduced in [39] to measure the relative improvement of a target model over a base model. Since the AUC of a random model is 0.5, RelaImpr is defined as:

$$RelaImpr = \left(\frac{AUC(target\ model) - 0.5}{AUC(base\ model) - 0.5} - 1 \right) \times 100\%. \tag{13}$$

5.2 Experimental Results: RQ1 and RQ2

As illustrated in Table 3, the consistent improvement of our SAR-Net over different contenders validates its efficacy. Note that the overall performance of Mix-Scenario models are worse than the single-scenario model and the multi-scenario models, which proves obscuring scenario difference hurts the modeling of multi-scenario CTR prediction. Besides, HPS, Cross-Stitch, MMOE, CGC, and PLE

Table 1: The percentage of training dataset and average CTR of each scenario.

Scenario	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
Percentage	11.20%	2.01%	5.30%	3.61%	2.39%	7.52%	3.44%	12.17%	8.51%	1.39%	4.33%	7.21%	3.99%	1.49%	6.63%	8.52%	4.25%	0.15%	2.78%	3.11%
CTR	5.22%	2.14%	6.39%	3.52%	4.55%	6.83%	4.32%	5.01%	9.39%	1.23%	9.34%	3.61%	5.25%	2.21%	5.29%	6.52%	4.58%	3.66%	4.79%	6.12%

Table 2: The percentage of users that visited certain number of scenarios (NoS) in past 30 days.

NoS	<2	2~4	4~6	6~8	8~10	>10
Percentage	11.21%	13.40%	25.30%	32.10%	11.52%	6.47%

all achieve better overall performance than single-scenario methods and mix-scenario methods, demonstrating the importance of exploiting the distinctions and relationship between scenarios.

Although single-scenario methods all achieve better performance than mix-scenario methods, it is notable that in scenario #18, the AUCs of single-scenario methods are worse than mix-scenario methods. We suspect that it is because the data of scenario #18 which is 0.15 percent of training dataset as illustrated in Figure 1, is not sufficient to train a reasonable model, while mix-scenario method has enough data to make it. On the other hand, in scenario #8, the DIN model trained in the single-scenario manner achieves better performance than HPS, Cross-Stitch, MMOE, CGC, and PLE, which is opposite to the overall performance results, it shows that in some scenarios, existing multi-task models fail to extract deeper information about the scenarios and users. In contrast, the proposed SAR-Net exhibits superior performance across all scenarios compared with single-scenario methods and mix-scenario methods. Besides, SAR-Net also achieves consistent improvement than HPS, Cross-Stitch, CGC, and PLE, which shows the superiority of explicitly modeling the user's interest transfer and extracting scenario-specific information.

5.3 Ablation Study: RQ3

5.3.1 Cross-Scenario Behavior Extract Layer. In this section, we investigate the impact of the attention mechanism in the Cross-Scenario Behavior Extract Layer. The base model is SAR-Net*, which extracts user behavior using a mean pooling operator (No Attention). In particular, we consider the following settings: 1) Target Attention: taking the target item as the query and the behavior item as the key in the attention mechanism; 2) Scenario Attention: taking scenario context feature as the query and the scenario context information at the moment that the behavior happened as key; 3) Concatenate Attention: concatenating target item feature and scenario context feature as query; 4) Hierarchical Attention: using a two-layer attention. In the first layer, scenario context feature concatenated with the target item is used as query. In the second layer, scenario context feature is used as the query. Multiplication of the weights of the two attention layers is used to generate the final pooling weights; and 5) Attention mechanism used in SAR-Net: target item and scenario context feature are respectively used as queries to learn the two-layer weights and element-wise item is used to generate the final pooling weights.

As illustrated in Table 4, "No attention" perform the worst, showing that useful signals could be easily buried in noise without distilling. In addition, either target item attention or scenario attention can improve the AUC compared with the base model, demonstrating that considering the relevance of target item or scenario context information both can bring gains. The third and fourth attention mechanisms that consider both target item and scenario context features perform better than the base model but worse than target attention, implying that directly concatenating target item and scenario context feature as the query could not fully attend and exploit the useful information. In contrast, SAR-Net learns attention weights from the perspective of target item and scenario context respectively and achieves the best performance. It shows that learning weights separately will avoid mutual interference, and can extract users' interest transfer across different scenarios.

5.3.2 Bias Net and Bias Adapting Loss. In this section, we investigate the impact of the Bias Net and Bias Adapting Loss in SAR-Net, which are used to mitigate the intervention bias issue. In particular, we try the loss without the weight of fair coefficients and the model without the bias net structure in the Expert Net as the base model, called SAR-Net†. We consider the following Settings: 1) sub-sampling items that are overexposed due to manual intervention. Sampling ratios were 0.9,0.8,0.7,0.6; 2) introducing the bias net to the expert network; 3) using the bias adapting loss; and 4) using the bias net and bias adapting loss.

As can be seen from Table 5, when the biased items being manual intervened are sub-sampled, the performance was improved slightly at first and then decreased. When the sub-sampling ratio is 0.7, the performance is the best. This is because sub-sampling will reduce the proportion of intervened items in the dataset, so that the model can learn the information of each type of items in a more fair way. However, when the proportion gradually decreases, the performance of the model will decrease, showing that excessive sub-sampling leads to the reduction of samples and the interaction information between users and items is not fully utilized. Bias net that is introduced to each expert net to reweigh the prediction of expert during training achieves better performance compared with base model and the sub-sampling method. Besides, bias adapting loss can make the model adaptive to the biased data and learn information according to the importance of samples. Our SAR-Net with both bias net and bias adapting loss can further boost the performance, demonstrating the effectiveness of the Bias Adapting Loss and Bias Net and their complementarity.

5.3.3 Scenario-Specific Transform Layer. Scenario-specific transform layer further models the differences and relations between scenarios by strengthening the key information of different scenarios. In this section, we take SAR-Net without the scenario-wise transform layer as the base model SAR-Net‡ and consider the following settings: 1) referring to PLE [30], a multi-layer expert network

Table 3: AUC of different models on the offline Alibaba production dataset. Single-Scenario denotes that models are trained using single scenario data independently. Mix-Scenario denotes that models are trained using all-scenario data. Multi-Scenario denotes that models are trained based on multi-task learning. "Overall" denotes that mixing samples from all scenarios to calculate AUC.

	Single-Scenario			Mix-Scenario			Multi-Scenario Model					
Scenario	Wide&Deep	PNN	DIN	Wide&Deep	PNN	DIN	HPS	Cross-Stitch	MMOE	CGC	PLE	**SAR-Net**
#1	0.6610	0.6526	0.6621	0.6556	0.6499	0.6568	0.6701	0.6731	0.6733	0.6752	0.6768	**0.6811**
#2	0.6740	0.6681	0.6751	0.6686	0.6654	0.6698	0.6801	0.6821	0.6831	0.6852	0.6858	**0.6901**
#3	0.6744	0.6685	0.6755	0.6690	0.6658	0.6702	0.6805	0.6816	0.6809	0.6808	0.6822	**0.6921**
#4	0.6842	0.6843	0.6883	0.6788	0.6816	0.683	0.6903	0.6952	0.6958	0.6961	0.6968	**0.7053**
#5	0.6660	0.6661	0.6701	0.6606	0.6634	0.6648	0.6721	0.6742	0.6745	0.6755	0.6769	**0.6829**
#6	0.6748	0.6749	0.6789	0.6694	0.6722	0.6736	0.6809	0.6828	0.6838	0.6855	0.6861	**0.6902**
#7	0.6745	0.6746	0.6786	0.6691	0.6719	0.6733	0.6806	0.6816	0.6822	0.6818	0.6827	**0.6936**
#8	0.6876	0.6901	0.6977	0.6796	0.6824	0.6838	0.6911	0.6915	0.6959	0.6962	0.6975	**0.7085**
#9	0.6440	0.6441	0.6481	0.6386	0.6414	0.6428	0.6501	0.6551	0.6593	0.6552	0.6568	**0.6681**
#10	0.6840	0.6841	0.6881	0.6786	0.6814	0.6828	0.6901	0.6921	0.6935	0.6962	0.6958	**0.7001**
#11	0.6745	0.6746	0.6786	0.6691	0.6719	0.6733	0.6806	0.6815	0.6811	0.6809	0.6829	**0.6929**
#12	0.6942	0.6943	0.6983	0.6888	0.6916	0.6930	0.7003	0.7052	0.7058	0.7061	0.7068	**0.7154**
#13	0.6460	0.6461	0.6501	0.6406	0.6434	0.6448	0.6521	0.6542	0.6545	0.6555	0.6569	**0.6729**
#14	0.6742	0.6743	0.6783	0.6688	0.6716	0.6730	0.6803	0.6822	0.6831	0.6857	0.6862	**0.6905**
#15	0.6748	0.6749	0.6789	0.6694	0.6722	0.6736	0.6809	0.6821	0.6831	0.6819	0.6825	**0.6934**
#16	0.6950	0.6951	0.6991	0.6896	0.6924	0.6938	0.7011	0.7015	0.7059	0.7062	0.7075	**0.7185**
#17	0.6860	0.6861	0.6901	0.6806	0.6834	0.6848	0.6921	0.6924	0.6979	0.6972	0.6985	**0.7095**
#18	0.6350	0.6351	0.6391	0.6357	0.6371	0.6411	0.6411	0.6551	0.6583	0.6592	0.6595	**0.6691**
#19	0.6840	0.6841	0.6881	0.6786	0.6814	0.6828	0.6901	0.6921	0.6935	0.6971	0.6988	**0.7021**
#20	0.6345	0.6346	0.6386	0.6291	0.6319	0.6333	0.6406	0.6465	0.6511	0.6509	0.6539	**0.6729**
Overall	0.6640	0.6642	0.6682	0.6587	0.6615	0.6629	0.6702	0.6759	0.6762	0.6791	0.6801	**0.6997**

Table 4: Ablation study of the Cross-Scenario Behavior Extract Layer. SAR-Net* is the base model by replacing the Cross-Scenario Behavior Extract Layer with mean pooling.

	Overall AUC	RelaImp
SAR-Net*	0.6925	0
SAR-Net*+Target Attention	0.6956	1.610
SAR-Net*+Scenario Attention	0.6934	1.004
SAR-Net*+Concatenate Attention	0.6944	0.987
SAR-Net*+Hierarchical Attention	0.6955	1.558
SAR-Net	**0.6997**	**3.741**

is used to extract features. And the number of layers is set to 2, 3, 4, 5, respectively; and 2) a single layer expert network is used and the scenario-wise transform layer is added before the expert network (SAR-Net). It can be seen from Table 6 that the benefit of the multi-layer extraction network structure converges gradually with the increase of the number of layers. Compared with the multi-layer expert extraction structure, scenario-specific transform layer achieves better results with less parameters.

Table 5: Ablation study of the Bias Net and Bias Adapting Loss. SAR-Net† denotes the base model that removes the bias net and uses the naive binary cross-entropy loss.

	Overall AUC	RelaImp
SAR-Net†	0.6911	0
SAR-Net†+sub-sampling(0.9)	0.6915	0.209
SAR-Net†+sub-sampling(0.8)	0.6921	0.523
SAR-Net†+sub-sampling(0.7)	0.6909	-0.104
SAR-Net†+sub-sampling(0.6)	0.6899	-0.628
SAR-Net†+Bias Net	0.6955	2.302
SAR-Net†+Bias Adapting Loss	0.6964	2.773
SAR-Net	**0.6997**	**4.500**

5.4 Online Deployment Test

We conduct the online A/B test by deploying our SAR-Net to handle real traffic in the personalized scenarios of Fliggy, Taobao and Alipay for seven days in November 2020, where the base model is MMOE [17]. The online evaluation metric is real CTR, which defined as the number of clicks over the number of item impressions. The experimental results are shown in Figure 2. It is clear that SAR-Net outperforms the base model MMOE [17] consistently, demonstrating the effectiveness of SAR-Net in practical multi-scenario CTR tasks. SAR-Net has been deployed in the online travel marketing platform in Alibaba and is now serving hundreds of travel scenarios.

Moreover, we analyzed the performance of SAR-Net as per different categories compared with the base model. As illustrated

Table 6: Ablation study of the Scenario-Specific Transform Layer. SAR-Net‡ denotes the base model without the Scenario-Specific Transform Layer.

	Overall AUC	RelaImp
SAR-Net‡	0.6948	0
SAR-Net‡+multi-layer experts(2)	0.6951	0.256
SAR-Net‡+multi-layer experts(3)	0.6956	0.523
SAR-Net‡+multi-layer experts(4)	0.6958	0.359
SAR-Net‡+multi-layer experts(5)	0.6957	0.301
SAR-Net	**0.6997**	**2.515**

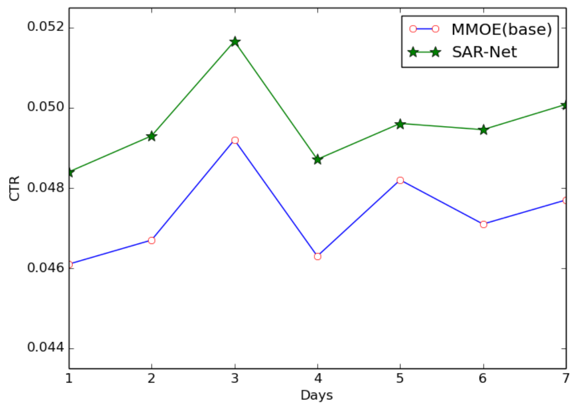

Figure 2: Online CTRs of SAR-Net and the base model in seven days in November 2020.

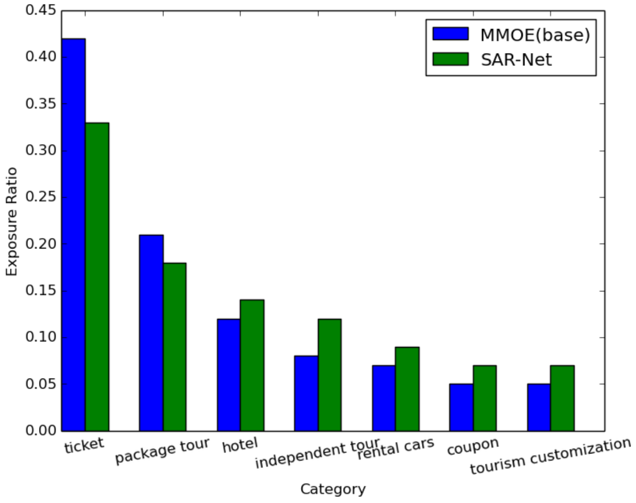

Figure 3: Online exposure ratios of SAR-Net and the base model as per category in seven days in November 2020. The intervened samples are removed from the statistics. Exposure ratio denotes the number of category's exposure over the number of all test data.

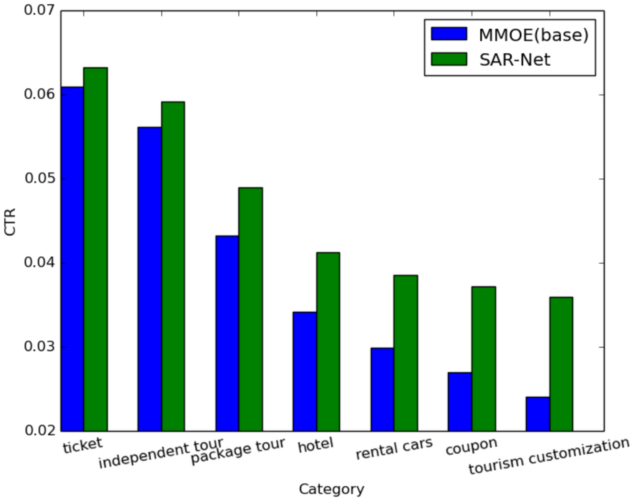

Figure 4: Online CTRs of SAR-Net and the base model as per category in seven days in November 2020. The intervened samples are removed from the statistics.

in Figure 3 and Figure 4. We found that: 1) SAR-Net makes the exposure ratio of each category more even; 2) SAR-Net achieves consistent improvement in all categories; and 3) the improvement is more obvious on the categories with smaller traffic. These results demonstrate that SAR-Net effectively mitigates the intervention bias issue and achieves fair recommendation for each item.

6 CONCLUSION

In this paper, we propose a novel Scenario-Aware Ranking Network (SAR-Net) to address two issues encountered in the context of Alibaba travel marketing platform, i.e., multi-scenario modeling issue and data fairness issue. SAR-Net harvests the abundant data from different scenarios by learning users' cross-scenario interests via two specific attention modules. Then, a scenario-specific transformation layer is adopted to further extract scenario-specific features, followed by two groups of debias expert networks. Furthermore, above intermediate results are fused into the final result by a multi-scenario gating module. In addition, we propose the concept of Fairness Coefficient to measure the importance of individual sample and use it to reweigh the prediction in the debias expert networks. In this way, SAR-Net can address above two issues efficiently. The experimental results on both offline dataset and from online A/B test demonstrates the superiority of SAR-Net over representative methods for multi-scenario prediction. SAR-Net has been deployed in the online travel marketing platform of Alibaba and is serving hundreds of travel scenarios, bringing a 5% improvement on CTR. In the future, we intend to investigate the impact of introducing more user fine-grained behaviors.

REFERENCES

[1] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided exposure bias in recommendation. *arXiv preprint arXiv:2006.15772* (2020).

[2] Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. 2009. Spatio-temporal models for estimating click-through rate. In *Proceedings of the 18th international conference on World wide web*. 21–30.

[3] Batya, Friedman, Helen, and Nissenbaum. 1996. Bias in computer systems. *ACM Transactions on Information Systems* (1996).

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

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *arXiv preprint arXiv:2010.03240* (2020).

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

[7] 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.

[8] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated Experiments on Ad Privacy Settings. *Proceedings on Privacy Enhancing Technologies* 2015, 1 (2015), 92–112.

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

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1725–1731.

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

[12] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. 2018. Homophily influences ranking of minorities in social networks. *Scientific reports* 8, 1 (2018), 1–12.

[13] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[14] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 447–456.

[15] Anja Lambrecht and Catherine Tucker. 2019. Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads. *Management Science* 65, 7 (2019), 2966–2981.

[16] Ze Lyu, Yu Dong, Chengfu Huo, and Weijun Ren. 2020. Deep Match to Rank Model for Personalized Click-Through Rate Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 156–163.

[17] 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.

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[19] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1222–1230.

[20] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3994–4003.

[21] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems* 20 (2007), 1257–1264.

[22] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for selection bias in learning-to-rank systems. In *Proceedings of The Web Conference 2020*. 1863–1873.

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

[24] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[25] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. 521–530.

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

[27] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems*. 213–220.

[28] Ana-Andreea Stoica, Christopher Riederer, and Augustin Chaintreau. 2018. Algorithmic Glass Ceiling in Social Networks: The effects of social recommendations on network diversity. In *Proceedings of the 2018 World Wide Web Conference*. 923–932.

[29] Yang Sun, Junwei Pan, Alex Zhang, and Aaron Flores. 2021. FMˆ2: Field-matrixed Factorization Machines for Recommender Systems. *arXiv preprint arXiv:2102.12994* (2021).

[30] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *Fourteenth ACM Conference on Recommender Systems*. 269–278.

[31] Can Wang, Jiawei Chen, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2020. SamWalker++: recommendation with informative sampling strategy. *arXiv preprint arXiv:2011.07734* (2020).

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

[33] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 115–124.

[34] Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. 2019. Multi-level deep cascade trees for conversion rate prediction in recommendation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 338–345.

[35] Hong Wen, Jing Zhang, Fuyu Lv, Wentian Bao, Tianyi Wang, and Zulong Chen. 2021. 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*.

[36] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2377–2386.

[37] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3119–3125.

[38] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep Multi-Interest Network for Click-through Rate Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2265–2268.

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

[40] Jing Zhang and Dacheng Tao. 2021. Empowering Things With Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet of Things Journal* 8, 10 (2021), 7789–7817. https://doi.org/10.1109/JIOT.2020.3039359

[41] Jiejie Zhao, Bowen Du, Leilei Sun, Fuzhen Zhuang, Weifeng Lv, and Hui Xiong. 2019. Multiple relational attention network for multi-task learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1123–1131.

[42] 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.

[43] 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.