Towards Scenario-adaptive User Behavior Modeling for Multi-scenario Recommendation

Jin Huang(✉), Yu Qian, Zixu Yang, and Zhijun Sun

Dewu, Shanghai, China
{huangjin01,qianyu01,yangzixu,zhangliang1}@shizhuang-inc.com

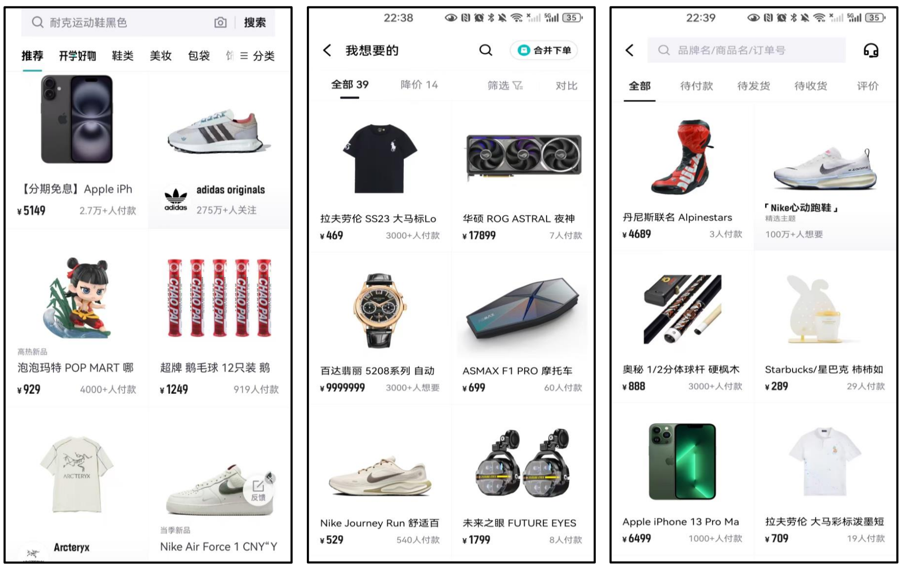
Abstract. Modern online service providers such as online shopping platforms often provide multiple scenarios to meet different user needs. Existing multi-scenario recommendation methods often train a unified model to serve all scenarios. Although much progress has been made, we argue that they ignore the importance of fusing scenario prior knowledge when modeling user's behaviors as the discrepancy of user's interests in different scenarios is significant. To solve this problem, we propose a novel **S**cenario-**a**daptive **I**nterest **Net**work, which explicitly incorporates scenario-related context into user behavior sequences, allowing for scenario-specific learning of interests refining. **SAINet** is stacked by a series of Scenario-adaptive Blocks. Each block adaptively integrates scenario-aware context into historical behaviors to distinguish the differences among scenarios and tailor interest representations to match current scenario of each instance. Extensive experiments on real-world datasets and online A/B test demonstrate the superiority of SAINet over state-of-the-art methods.

Keywords: Click-through Rate Prediction · Multi-scenario Learning · User Interest Modeling.

1 Introduction

In today's online platforms, the recommender system is broadly deployed to offer relevant products and fetch personalized services for users. Many online shopping platforms (e.g., Taobao and Amazon) develop multiple shopping scenarios to deliver high-quality recommendation results and boost the platform revenues. For example, Figure 1 shows several common scenarios in a shopping app (e.g., Dewu): (1) *Homepage*: users browse infinite recommended items by sliding screen just like waterfall. Various items are displayed in it to attract users. (2) *Cart page* or *Order page*: It suggests items to users after they cart or purchase some specific products and the recommended items depend more on the user's past purchased records. Clearly, these two typical scenarios are very different from each other. People in *Homepage* tend to have diverse interests (e.g., users prefer browsing more brands and categories of items, and the price selecting range of them is broader). Conversely, users may show more specific and concentrated preferences in *Order page*. For instance, if a user has recently bought a new phone in the

Order page, he may be more concerned about electronic devices such as headphones and phone cases. However, early industrial practices on multi-scenario recommendation neglect the significant interests discrepancy across scenario and the full utilization of scenario informations involved in behaviors.



(a) Homepage (b) Cart page (c) Order page

Fig. 1. Examples of multiple scenarios for the E-commerce platform

Traditional methods [4,5,8,13,14,27] typically train separate models for each scenario with scenario-specific data. However, this isolated model training approach usually requires more complex computation and maintenance cost. Also, separate models fail to capture interrelations across scenarios. Currently, state-of-the-art models generally train a unified model with merged data of all scenarios to learn commonalities and specific characteristics among scenarios. There are two mainstream unified multi-scenario approaches to model correlations and interrelations: 1) Scenario-specific network structures [20,9,28,17,29,26], inspired by multi-task learning (MTL) [7,15,10,19]. They implement a single model that deals with data in various scenarios and treat each scenario as a specific task. 2) Parameter adaptive network structures [3,24,22], influenced by LHUC [18] algorithm proposed in the field of speech recognition. They propose applying scenario context information as input and dynamically scaling the bottom-level embedding and top-level DNN hidden units through gate mechanisms to learn variety of scenarios.

All of them ignore a crucial issue: they regard user's entire historical behavior sequence as a whole and do not integrate the scenario knowledge into sequence modeling to obtain more effective recommendation results. As illustrated in Figure 2 (a), the coarse-grained weight adjustment methods (e.g., PEPNet[3]) uniformly apply the same weight to the representation aggregated from historical behavior. SAR-Net [16] in Figure 2 (b) attempts to fuse scenario context of current instance when calculating weights of each behavior in sequence by target attention mechanism [27]. They are incapable to distinguish which scenario each interaction in behavior sequence comes from as user behaviors differ greatly among different scenarios, which would inevitably reduce the accuracy of the recommendation.

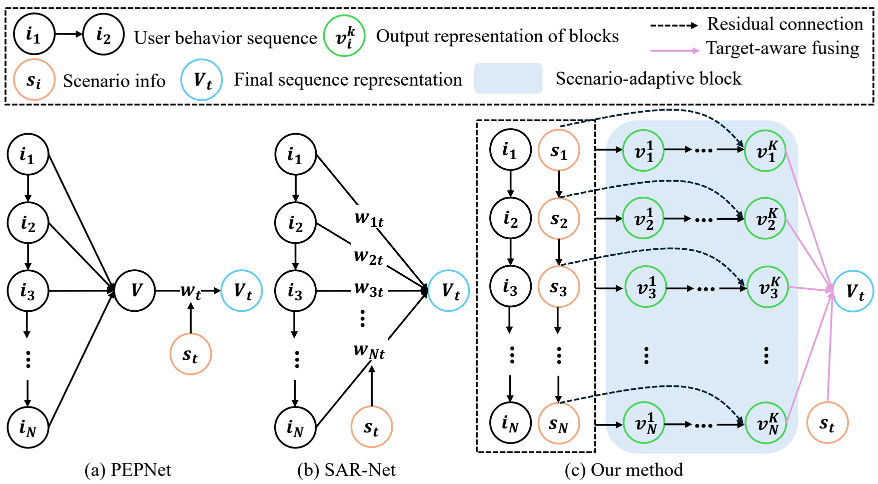


Fig. 2. Examples of previous methods and ours. In (a) $V_t = w_t V$, where $w_t = g(s_t)$ and $V = f(i_i, \cdots, i_N)$. (b) $V_t = f(w_{1t} i_1, \cdots, w_{Nt} i_N)$, where $w_{nt} = g(s_t)$. (c) $\{v_1^k, \cdots, v_N^k\} = f\left(\left(v_1^{k-1}, s_1\right), \cdots, \left(v_N^{k-1}, s_N\right)\right)$ and $V_t = g\left(\{v_1^k, \cdots, v_N^k\}, s_t\right)$, where $\{v_1^0, \cdots, v_N^0\} = \{i_1, \cdots, i_N\}$.

Therefore, to address the above issue, we propose a novel **S**cenario-**a**daptive **I**nterest **Net**work (**SAINet**, Figure 2 (c)), which explicitly fuses scenario adaptive knowledge in behaviors and models the discrepancy of user behaviors in different scenarios, then captures user's fine-grained interests to enhance recommendation accuracy coping with multi-scenario problem. Specifically, SAINet first stacks several Scenario-adaptive Blocks composed of Scenario-aware Interest Extracting (SIE) and Scenario Tailoring Module (STM). SIE adaptively injects scenario-aware context into user behavior sequence to capture user's interests more precisely. Meanwhile, STM tailors user's interest representations with target scenario information to further obtain user's interests which are significantly related to current scenario. By stacking blocks repeatedly, a deep network is built that progressively bolster its ability to model the behavior discrepancy of different scenarios. Then, Target-aware Interest Fusion (TIF) fused all interest representations to promote its integration with other features by attention mechanism. Finally, by leveraging the benefits of top-layer DNN hidden units dynamic scaling, we incorporate the Scenario-aware DNN Tower (SDT) at the prediction stage enhancing the model's capability. The contributions of our paper can be summarized as follows:

(1) To the best of our knowledge, SAINet is the first multi-scenario work to distinguish user's behaviors from different scenarios, modeling interests discrepancy and migration across scenarios effectively.

(2) SAINet proposes Scenario-adaptive Block to fuse scenario prior knowledge into behaviors to capture user's fine-grained interests and trim user's interest representations with target scenario information of current instance. By stacking the block, SAINet constructs a deep network that enhances its ability to model the differences between scenarios.

(3) Extensive offline and online experiments show that the proposed SAINet consistently and significantly outperforms all baselines. SAINet has already been deployed in industrial system.

2 Related Work

With the rapid growth of E-commerce platforms, various scenarios emerge simultaneously to better meet the user's diversified needs. Modeling multiple recommendation scenarios has become crucial. Recent methods adopt a unified ranking model to serve all scenarios. There are two typical categories of methods: 1) Scenario-specific network structures; 2) Parameter adaptive network structures.

Scenario-specific network structures are inspired by Multi-Task Learning (MTL)[2,1]. These methods apply specific networks for each scenario and output several scenario-specific ranking scores. MoE [7,15] proposes to select sub-expert based on the shared-bottom input. MMoE [10] adapts the MoE structure by sharing the subnets across all tasks, while having a light-weight gating network trained to optimize each task. However, MMoE suffers from the seesaw phenomenon (i.e., improvement of one task often leads to performance degeneration of the other tasks). To address the problem, PLE [19] separates shared components and task-specific components explicitly and adopts a progressive routing mechanism to extract and separate deeper semantic knowledge gradually, improving efficiency of joint representation learning and information routing across tasks in a general setup. HMoE [9] takes advantage of MMoE to implicitly identify distinctions and commonalities between scenarios. However, HMoE is difficult to capture the shared and specific information explicitly as multi-scenario datasets are complex. AESM^2 [29] proposes a novel expert network structure with automatic selection of fine granularity by calculating the KL divergence to select the most suitable sharing and specific experts.

However, scenario-specific network structures neglect the differences among scenarios in the bottom-level representations and are all composed of heavy networks. Consequently, Parameter adaptive network structures are proposed. AdaSparse [22] learns adaptive sparse structure for each scenario, enhancing generalization across domains. SASS [25] designs a scenario adaptive transfer module to inject useful information from whole scenario to individual scenario. DFFM [6] incorporates scenario-related information into the parameters of feature interaction and user behavior modules. PEPNet [3] takes scenario related features as input and dynamically scales the bottom-layer embeddings and the top-layer DNN hidden units through a gate mechanism. SFPNet [23] comprises a series of blocks named as scenario-tailoring block and integrates scenario information at a coarse-grained level by redefining fundamental features.

All of them regard user's entire historical behavior sequence as a whole and are incapable to distinguish which scenario each interaction in behavior sequence comes from. This may lead to insufficient modeling for user's interests, and then the capability of recommendation system may be suppressed. To address these issues, we proposed SAINet which integrates the scenario knowledge into sequence modeling to obtain more effective recommendation results of multi-scenarios.

3 Preliminaries

Let $\mathcal{I} = \{i_1, \ldots, i_{|\mathcal{I}|}\}$ be a set of $|\mathcal{I}|$ items, and $\mathcal{S} = \{s_1, \ldots, s_K\}$ denote a set of K scenarios. $\mathcal{B}^u = \{i_1, \ldots, i_N\}$ represents a chronological sequence of historical behaviors of user u and N represents the length of the sequence. Given target item $i_t \in \mathcal{I}$, target scenario $s_t \in \mathcal{S}$ and other features $\boldsymbol{o_f}$, the multi-scenario recommendation task aims to devise a unified ranking model to simultaneously provide accurate and personalized recommendations across K scenarios. We select click-through rate (CTR) prediction as our task, which can be formulated as:

$$\hat{y} = P(y = 1 | i_t, s_t, \boldsymbol{o_f}, \mathcal{B}^u) \tag{1}$$

CTR prediction is to predict the probability $\hat{y}$ that user u interacts with target item i_t in scenario s_t. We adopt the widely-used embedding technique [12] to transform sparse features into low-dimensional dense vectors. For instance, $\boldsymbol{e_{i_t}}$ and $\boldsymbol{e_o}$ represent the embedding of target item i_t and other features respectively.

4 The Approach

The architecture of SAINet is shown in Figure 3, mainly consisting of three components.

4.1 Scenario-adaptive Block

As illustrated in Figure 2, existing practices fail to distinguish the differences of user behaviors from different scenarios when performing multi-scenario recommendation. However, the scenario-aware prior knowledge contained in the behavior sequences has significant impact on improving the accuracy of recommendation results. Hence, we design Scenario-adaptive Block, which adaptively

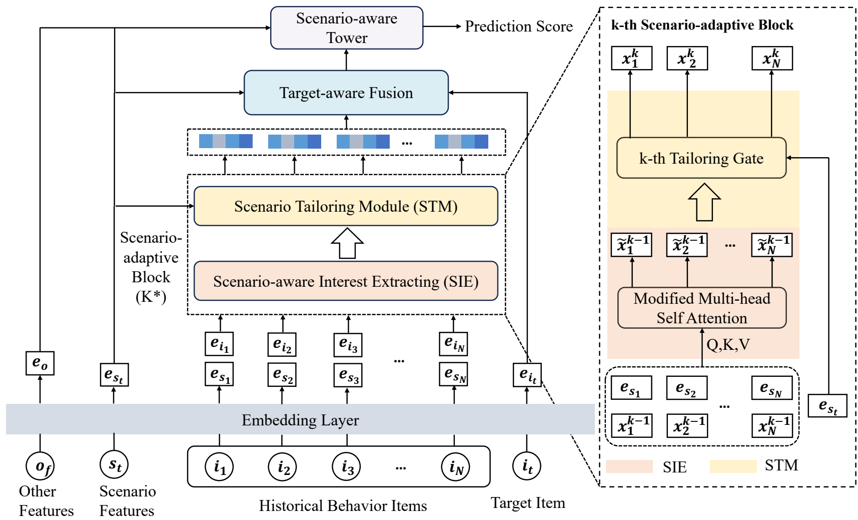


Fig. 3. Overall framework of the SAINet model.

injects scenario-aware context into user behavior sequences to acquire comprehensive and fine-grained interest representation. Meanwhile, it tailors user's interest representations with target scenario information to further capture user's interests which are markedly related to current scenario. Scenario-adaptive Block is stacked by L layers and each block comprises two modules. By stacking the block, SAINet constructs a deep network that progressively bolsters its ability to model the behavior discrepancy of different scenarios. To elucidate, we give the computation process within the l-th block, delineated as follows:

Scenario-aware Interest Extracting. The Scenario-aware Interest Extracting (SIE) module is devised to integrate scenario prior informations from historical behaviors and extract more fine-grained interests. We employ the modified multi-head self attention (MHA) [21] involved with specific scenario knowledge. $\boldsymbol{X^{l-1}} = \left[\boldsymbol{x_1^{l-1}}, \cdots, \boldsymbol{x_N^{l-1}}\right] \in \mathbb{R}^{N \times d}$ represents the output of the $(l-1)$-th block and N denotes the number of interacted items. Mathematically, The encoded interest matrix of $\boldsymbol{X^{l-1}}$ after MHA, denoted by $\boldsymbol{\hat{X}^{l-1}}$, is calculated as:

$$
\begin{aligned}
\boldsymbol{\hat{X}^{l-1}} &= MultiHead\left(\boldsymbol{X^{l-1}}\right) = concat\left(head_1, \cdots, head_h\right)\boldsymbol{W^O}, \\
head_i &= Attention\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = Softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V}
\end{aligned}
\tag{2}
$$

The $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ integrate the scenario embedding into the output of the $(l-1)$-th block to obtain more precise representations of interests. Let $\boldsymbol{M_S} = [\boldsymbol{e_{s_1}}, \cdots, \boldsymbol{e_{s_N}}] \in \mathbb{R}^{N \times d}$ denote the embedded matrix of scenarios corresponding to each behavior comes from. Then the integrating process is defined as:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{X^{l-1}} \boldsymbol{W_i^Q} + \boldsymbol{M_S} \boldsymbol{W_S^Q}, \\ \boldsymbol{K} &= \boldsymbol{X^{l-1}} \boldsymbol{W_i^K} + \boldsymbol{M_S} \boldsymbol{W_S^K}, \\ \boldsymbol{V} &= \boldsymbol{X^{l-1}} \boldsymbol{W_i^V} + \boldsymbol{M_S} \boldsymbol{W_S^V} \end{aligned} \tag{3}$$

where h represents the amount of heads. $\boldsymbol{W^O} \in \mathbb{R}^{hd_k \times d}$ denotes the weight matrix of output linear transformation with d$=h \times d_k$. $\boldsymbol{W_i^Q}, \boldsymbol{W_i^K}, \boldsymbol{W_i^V} \in \mathbb{R}^{d \times d_k}$ are projection matrices for the i-th head corresponding to query, key, and value respectively. $\boldsymbol{W_S^Q}, \boldsymbol{W_S^K}, \boldsymbol{W_S^V} \in \mathbb{R}^{d \times d_k}$ are transformation matrices for embedding of scenario knowledge.

Scenario Tailoring Module. Although SIE considers utilizing scenario information in historical behavior, it ignores the target scenario information, which is important to capture user's interests related to current scenario markedly. We propose Scenario Tailoring Module (STM) to further tailor user's interest representations (i.e., $\boldsymbol{\hat{X}^{l-1}} = \left[\boldsymbol{\hat{x}_1^{l-1}}, \cdots, \boldsymbol{\hat{x}_N^{l-1}}\right]$). STM consists of N light gated units. The i-th gated adjustment is calculated as:

$$\boldsymbol{v_i^{l-1}} = \gamma * Sigmoid\left(concat\left(\boldsymbol{\hat{x}_i^{l-1}}, \boldsymbol{e_{s_t}}\right) \boldsymbol{W_G} + \boldsymbol{b_G}\right) \in \mathbb{R}^d \tag{4}$$

$$\boldsymbol{x_i^l} = \boldsymbol{\hat{x}_i^{l-1}} \otimes \boldsymbol{v_i^{l-1}} \tag{5}$$

where $\boldsymbol{W_G}$ and $\boldsymbol{b_G}$ are projection matrices and bias respectively. γ is a scaling factor to further squash and magnify the tailoring signal. $\boldsymbol{e_{s_t}}$ denotes the embedding of the target scenario. $\otimes$ is element-wise product. Finally the interest representations of the l-th block can be defined as $\boldsymbol{X^l} = \left[\boldsymbol{x_1^l}, \cdots, \boldsymbol{x_N^l}\right] \in \mathbb{R}^{N \times d}$

The Scenario-adaptive Block is iterated L times to improve its ability to capture the behavior differences among scenarios. Specially, the original input of the first block (i.e., $\boldsymbol{X^0}$) is defined as:

$$\boldsymbol{X^0} = \left[\boldsymbol{x_1^0}, \cdots, \boldsymbol{x_N^0}\right] = \left[\boldsymbol{e_{i_1}}, \cdots, \boldsymbol{e_{i_N}}\right] \tag{6}$$

where $\left[\boldsymbol{e_{i_1}}, \cdots, \boldsymbol{e_{i_N}}\right]$ represents the embedding matrix of the user's behavior sequence $\mathcal{B}^u$.

4.2 Target-aware Interest Fusion

After Scenario-adaptive Block, several representations of interests is generated. All representations must first be fused to facilitate its integration with other feature vectors transferring into downstream DNN network. We employ Target-aware Interest Fusion (TIF) by an attention mechanism for fusing:

$$\boldsymbol{V_t} = \sum_{j=1}^{N} \mu_j \boldsymbol{W_V} \boldsymbol{x_j^l} \tag{7}$$

where $\boldsymbol{V_t}$ represents the fusing interest representation corresponding to target item and scenario. $\boldsymbol{W_V}$ is learning parameter. μ_j is the attention weight, which can be formulated as follows:

$$\mu_j = \frac{exp\left(\alpha_j\right)}{\sum_{m=1}^{N} exp\left(\alpha_m\right)}, \quad \alpha_j = \boldsymbol{W_Q z_t} \cdot \boldsymbol{W_K x_j^l} \tag{8}$$

where α_j is the relevance between $\boldsymbol{z_t}$ and user's j-th interest representation $\boldsymbol{x_j^l}$. $\boldsymbol{z_t} = concat\left(\boldsymbol{e_{i_t}}, \boldsymbol{e_{s_t}}\right)$ represents the concatenation of target item embedding and scenario embedding. $\boldsymbol{W_Q}$ and $\boldsymbol{W_K}$ are learning parameters.

4.3 Scenario-aware DNN Tower

Although we prioritize the bottom level's architecture (i.e., user's behaviors), the top level's optimization must not be overlooked [3]. Subsequently, we leverage the benefits of top-layer DNN hidden units dynamic scaling by incorporating Scenario-aware DNN Tower (SDT) at the prediction stage. We first concatenate all outputs:

$$\boldsymbol{H^0} = cocnat\left(\boldsymbol{V_t}, \boldsymbol{e_{i_t}}, \boldsymbol{e_o}\right) \tag{9}$$

Then the scenario-aware DNN tower is used to forecast the likelihood of users clicking on the target item:

$$\hat{y} = \sigma\left(sDNN\left(\boldsymbol{H^0}, \boldsymbol{e_{s_t}}\right)\right) \tag{10}$$

where the specific calculation of the j-th layer in the sDNN is:

$$\boldsymbol{H^j} = Relu\left(\boldsymbol{W_T^j}\left(\boldsymbol{H^j} \otimes \sigma\left(concat\left(\boldsymbol{H^j}, \boldsymbol{e_{s_t}}\right)\right)\right) + \boldsymbol{b_T^j}\right) \tag{11}$$

where σ is sigmoid activation function. $\boldsymbol{W_T^j}, \boldsymbol{b_T^j}$ are the weight and bias for the j-th layer respectively. For CTR task, we use the widely used cross entropy loss as the objective function:

$$L_{rec} = -\frac{1}{|D|}\sum_{i=1}^{|D|}\left(y_i log\left(\hat{y}_i\right) + \left(1 - y_i\right) log\left(1 - \hat{y}_i\right)\right) \tag{12}$$

where y_i is the ground truth of instance. $|D|$ is the cardinality of samples.

5 Experiments

In this section, we conduct plenty of experiments to validate the effectiveness of our proposed framework and answer the following questions:

- **RQ1.** How does SAINet perform compared with state-of-the-art baselines?
- **RQ2.** How does each module in SAINet work?
- **RQ3.** How the hyper-parameters in the proposed SAINet affect its performance?

5.1 Experiment Settings

Datasets. We conduct our experiments over two real-world datasets as follows. Statistics of them are listed in Table 1.

- **AliCCP[1].** AliCCP is a public dataset released by Taobao with training and testing set, which is widely used in the relevant literature [11] for recommendation area. We split the dataset into three scenarios (abbreviated as #C1 to #C3) according to the context feature value as previous work [11].
- **Industrial Dataset.** It involves user logs in five scenarios (denoted as #A1 to #A5), randomly sampled at an industrial APP from the date 20/08/2023 to the date 04/09/2023. We use logs of the last day in the dataset as testing set, and the remaining logs are used as training set.

Table 1. Statistics of two datasets. (M-million,K-thousand)

	Industrial Dataset					AliCCP		
Scenario	#A1	#A2	#A3	#A4	#A5	#C1	#C2	#C3
Users	0.97M	0.93M	0.57M	0.49M	0.37M	91K	2.6K	154K
Items	2.72M	2.4M	1.79M	1.69M	1.22M	0.54M	0.2M	0.54M
Samples	145.77M	136.73M	52.12M	39.05M	21.09M	32M	0.64M	85M

Metrics. We adopt widely-used accuracy metric, i.e, AUC, to verify model performance. AUC denotes the area under the ROC curve over the testing set. A small improvement of AUC is likely to lead to a significant increase in CTR at real industrial platform.

Baselines. To demonstrate the effectiveness of our proposed model, SAINet is compared with three categories of approaches in multi-scenario recommendation.

(1) General recommenders. Samples from all scenarios are merged to train a unified ranking model.

- **DNN**: It is a common model with all parameters sharing by a single DNN tower for prediction.
- **DeepFM[5]**: It combines the factorization machines and DNN components eliminating feature engineering work.

(2) Scenario-specific network structures. Each scenario is regarded as a distinct task with multiple scenario-specific networks.

- **SharedBottom (SBT)**: It shares all parameters at bottom level and employs multiple scenario-specific DNN towers at top level.

[1] `https://tianchi.aliyun.com/dataset/408`

- **MMoE[10]**: It transfers the original multi-task learning to multi-scenario learning. MMoE applies gating networks to adjust the bottom expert representations followed by scenario-specific towers and learns to model scenarios' relationships from data.
- **PLE[19]**: Scenario-shared experts and scenario-specific experts are introduced based on MMoE to alleviate the seesaw phenomenon effectively.
- **AESM^2[29]**: It proposes a novel expert network structure with automatic selection of fine granularity by calculating the KL divergence to select the most suitable sharing and specific experts dynamically.

(3) Parameter adaptive network structures. Scenario context is directly applied to the embedding layer and the DNN's hidden layers, and adjusts model parameters dynamically with considering scenario variations.

- **PEPNet[3]**: It takes scenario related features as input and dynamically scales the bottom-layer embeddings and the top-layer DNN hidden units in the model through a gate mechanism.
- **AdaSparse (ADS) [22]**: It learns sparse structure for each scenario adaptively, enhancing generalization across scenarios by pruning redundant neurons with learned weights.
- **SFPNet[23]**: It comprises a series of scenario-tailoring blocks and integrates scenario information at a coarse-grained level by redefining fundamental features, simultaneously incorporates target scenario information into behaviors to support scenario-aware user interest modeling.

Parameter Setting. We use Adam optimizer with batch size of 4096 and run the experiments with an initial learning rate of 0.001 for all comparison methods. A Gaussian distribution ($\mu = 0$ and $\sigma = 0.05$) is used to initialize the parameters in DNN. All methods use a four-layer feedforward neural network with hidden sizes of [512, 256, 128, 64] for prediction.

5.2 Comparison with Baselines (RQ1)

Table 2 shows the comparison results of all methods on both datasets. For each method, we repeat the experiment five times and report the averaged results. The statistical significance test is conducted by using $t-test$. Performance of our method against the best baseline is statistically significant at 0.05 level. We can make the following conclusions. All *General recommenders* methods consistently achieve inferior performance compared to other practices on two datasets. This is because all of them ignores interrelations and differences among scenarios. To address it, *Scenario-specific network structures* are proposed and achieve significant performance enhancements. The SharedBottom adds several scenario-specific DNN towers to utilize the scenario-specific knowledge, which is incapable for capturing the complex interplay between scenarios. MMoE uses experts and gating networks to extract commonalities of different scenarios and gets better results. But MMoE exhibits seesaw phenomenon (i.e., improvement

Table 2. Comparison of different methods on two datasets.

Method	Industrial dataset					AliCCP		
	#A1	#A2	#A3	#A4	#A5	#C1	#C2	#C3
DNN	0.6879	0.6918	0.7271	0.6945	0.6807	0.6053	0.6014	0.5527
DeepFM	0.6968	0.6962	0.7415	0.6985	0.6835	0.6126	0.6095	0.5923
SBT	0.6984	0.6985	0.7447	0.6956	0.6807	0.6195	0.6153	0.5926
MMoE	0.6994	0.6993	0.7455	0.6974	0.6813	0.6223	0.6156	0.593
PLE	0.7004	0.7005	0.7468	0.6986	0.6848	0.6227	0.6167	0.5937
AESM^2	0.7011	0.7034	0.7415	0.7044	0.6909	0.6233	0.6176	0.5969
PEPNet	0.7021	0.7045	0.7427	0.7061	0.6938	0.6237	0.6189	0.5973
ADS	0.7033	0.7052	0.7435	0.7086	0.6947	0.6253	0.6196	0.5985
SFPNet	0.7063	0.7077	0.7558	0.7106	0.7061	0.6266	0.6224	0.5995
SAINet	**0.7136**	**0.7117**	**0.7585**	**0.7152**	**0.7134**	**0.6315**	**0.6248**	**0.6026**

of one scenario often leads to performance degeneration of the other scenarios) across multiple scenarios. For instance, MMoE has unsatisfactory performance in the scenario #A4 and the scenario #A5 of industrial dataset compared with DeepFM as these scenarios have more different characteristics and uneven data distributions, and MMoE is insufficient for dealing with it. PLE alleviates the phenomenon and demonstrates significant improvement over MMoE on both datasets by splitting experts into two groups, i.e, scenario-shared and scenario-specific partly. AESM^2 further introduces expert automatic selection mechanism to gain better performance than PLE.

Scenario-specific network structures merely optimize the top level of networks. However, the differences among scenarios in the bottom-level representations are also crucial for achieving good performance. PEPNet derives good performance improvement on both datasets by taking scenario related features as input and dynamically scales the bottom-layer embeddings and the top-layer DNN hidden units in the model through a gate mechanism. AdaSparse's performance is worse compared with PEPNet as the mechanism of sparse hidden units is difficult to learn. The SFPNet shows best performance among all baselines in the experiment as it integrates scenario information redefining fundamental features, simultaneously incorporates target scenario information into behaviors to support scenario-aware user interest modeling. However, All of these works neglect utilizing scenario prior knowledge involved in historical behavior sequence and target item to effectively model user's interest discrepancy across scenarios. Our proposed SAINet explicitly models the differences of user behaviors in different scenarios by introducing abundant scenario context. In addition, interest migration across scenarios can be easily captured. SAINet outperforms all baselines across all scenarios on both datasets as shown in Table 2.

5.3 Ablation Study (RQ2)

To evaluate the effectiveness of each module in SAINet, we also compare SAINet with its variants. The following variants are considered:

- **w/o SIE** removes the SIE from Scenario-adaptive Blocks, which means that the block no longer emphasizes scenario information in historical behavior.
- **w/o STM** removes the STM from Scenario-adaptive Blocks, which implies that target scenario is not utilized to tailor interest representation.
- **w/o TIF** simply removes the TIF module from SAINet and replaces it with mean pooling operation.
- **w/o SDT** simply removes the SDT module from SAINet and replaces it with a normal DNN network.

As indicated by the Table 3, each module makes a considerable contribution to ensure the performance of of SAINet. Specifically, the absence of SIE (w/o SIE) impacts prediction performance across all scenarios, demonstrates that integrating scenario prior knowledge from historical behaviors is pivotal when enhancing the capability of modeling user interest discrepancy of different scenarios. Furthermore, the removal of STM (w/o STM) also leads to a measurable reduction in the model's prediction performance. This robustly validates the effectiveness of tailoring user's interest representations to capture user's preference which is markedly related to current scenario. Additionally, without the TIF module, model performance declines in both dataset partly. This reflects that interests fusion with introducing target attention mechanism rather than mean pooling is very helpful in ensuring the predictive accuracy. Lastly, removing the SDT will damage the performance of model. This reveals that the top level network adjustment with the assistance of scenario information can also not be overlooked to obtain superior results.

Table 3. Comparison of different SAINet variants on two datasets.

Models	Industrial dataset					AliCCP		
	#A1	#A2	#A3	#A4	#A5	#C1	#C2	#C3
SAINet	**0.7136**	**0.7117**	**0.7585**	**0.7152**	**0.7134**	**0.6315**	**0.6248**	**0.6026**
w/o SIE	0.7086	0.7074	0.7543	0.7118	0.7087	0.6278	0.6202	0.5977
w/o STM	0.7103	0.7087	0.7556	0.7135	0.7104	0.6296	0.6216	0.6001
w/o TIF	0.7114	0.7106	0.7567	0.7146	0.7121	0.6307	0.6226	0.6014
w/o SDT	0.7133	0.7114	0.7583	0.7161	0.7138	0.6312	0.6245	0.6027

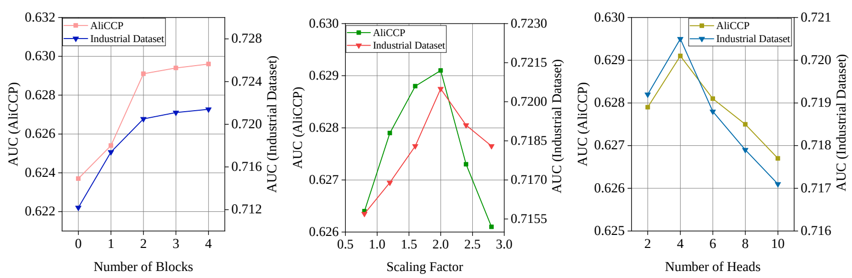
5.4 Hyper-Parameters Study (RQ3)

We conduct extensive experiments to examine the effects of several key hyperparameters, which include the numbers L for the Scenario-adaptive Block, scaling factor γ in the STM module, the numbers of heads h in the SIE module.

Effect of Block Numbers Figure 4 (a) illustrates the impact of different L. When the numerical value increases, AUC shows a trend of improvement. This is mainly due to the deeper interaction between interest representation and scenario context as L gets larger, while increasing L beyond 2 bringing no remarkable benefit.

Effect of Scaling Factor. The scaling factor is tuned carefully in {0.8, 1.2, 1.6, 2.0, 2.4, 2.8}. According to the curves shown in Figure 4 (b), when the value of factor equals to 2, SAINet achieves its best performance in terms of AUC, while increasing value beyond 2 reduces its performance. Hence, we set the scaling factor in SAINet and its variants to 2 in all experiments.

Effect of Head Numbers. We conducted experiments with varying h from {2, 4, 6, 8, 10}. Figure 4 (c) displays the impact of the numbers of heads for the multi-head attention mechanism in the SIE. When the numbers of heads equal to 4, AUC curve reaches its peak, while introducing more heads beyond 4 brings worse performance. Hence, h is set to 4 in all experiment.



(a) Number of Blocks L (b) Scaling Factor γ (c) Number of Heads h

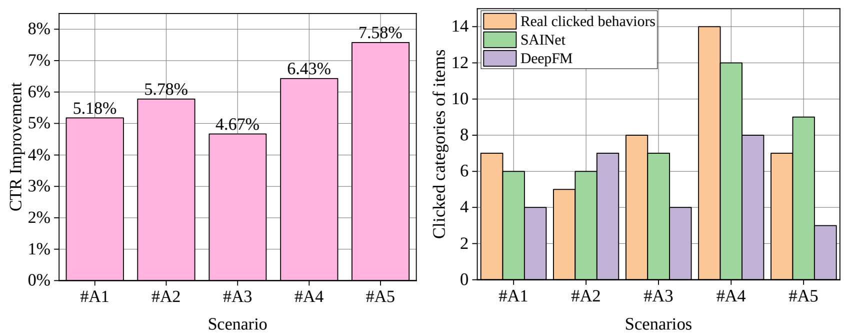
Fig. 4. Hyperparameter study of SAINet

6 Online A/B Test

We conduct fair online A/B test based on the real traffic in an online E-commerce platform. To be specific, we deploy SAINet and comparison methods in online serving system and execute inference tasks on daily requests of users. We take the averaged results in seven days, SAINet obtains 5.23% overall CTR gains over DeepFM online. 5.23% is a significant increase in a mature industrial system. Online test results compared with DeepFM of each scenario are illustrated in Figure 6 (a). It shows that SAINet has significant yet consistent improvement across the five scenarios.

7 Case Study

To further explain how SAINet models interests discrepancy by fusing scenario information, the representative case for a randomly selected real-world user is



(a) Online A/B test results (b) Case study of user1

illustrated in Figure 6 (b). Three histograms represent the average clicked categories of the items that users clicked in the past seven days by scenarios, the average categories of the items that SAINet and DeepFM recommend in recent three requests by scenarios, respectively. The distribution of categories that users have interacted with can represent coarse-grained interests of users partly. It is obvious that the number of items' categories recommended by SAINet is closer to the user's historical preferences than DeepFM in all scenarios, which proves that SAINet can more effectively distinguish user interest discrepancy across scenarios.

8 Conclusion

In this paper, we highlight the necessity of distinguishing the discrepancy of user behaviors in different scenarios for interest modeling and devise a novel model named SAINet. It first introduces a series of Scenario-adaptive Blocks, which fuse scenario prior knowledge into behaviors to capture user's fine-grained interests and tailor interest representations with target scenario context. By stacking the block, the ability to model the interests discrepancy of different scenarios can be enhanced. SAINet also utilizes Scenario-aware DNN Tower (SDT) to scale top-layer DNN hidden units achieving better predicting results. Extensive offline and online experiments demonstrate the superiority of SAINet in multi-scenario recommendation.

References

1. Argyriou, A., Pontil, M., Ying, Y., Micchelli, C.: A spectral regularization framework for multi-task structure learning. Advances in neural information processing systems **20** (2007)
2. Caruana, R.: Multitask learning. Machine learning **28**, 41–75 (1997)
3. Chang, J., Zhang, C., Hui, Y., Leng, D., Niu, Y., Song, Y., Gai, K.: 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. pp. 3795–3804 (2023)
4. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems. pp. 7–10 (2016)
5. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247 (2017)
6. Guo, W., Zhu, C., Yan, F., Chen, B., Liu, W., Guo, H., Zheng, H., Liu, Y., Tang, R.: Dffm: Domain facilitated feature modeling for ctr prediction. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 4602–4608 (2023)
7. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991)
8. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
9. Li, P., Li, R., Da, Q., Zeng, A.X., Zhang, L.: 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. pp. 2605–2612 (2020)
10. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: 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. pp. 1930–1939 (2018)
11. Ma, X., Zhao, L., Huang, G., Wang, Z., Hu, Z., Zhu, X., Gai, K.: 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. pp. 1137–1140 (2018)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26** (2013)
13. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295 (2001)
14. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The adaptive web: methods and strategies of web personalization, pp. 291–324. Springer (2007)
15. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
16. Shen, Q., Tao, W., Zhang, J., Wen, H., Chen, Z., Lu, Q.: Sar-net: A scenario-aware ranking network for personalized fair recommendation in hundreds of travel scenarios. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 4094–4103 (2021)
17. Sheng, X.R., Zhao, L., Zhou, G., Ding, X., Dai, B., Luo, Q., Yang, S., Lv, J., Zhang, C., Deng, H., et al.: One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 4104–4113 (2021)
18. Swietojanski, P., Li, J., Renals, S.: Learning hidden unit contributions for unsupervised acoustic model adaptation. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**(8), 1450–1463 (2016)

19. Tang, H., Liu, J., Zhao, M., Gong, X.: Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In: Proceedings of the 14th ACM Conference on Recommender Systems. pp. 269–278 (2020)
20. Tian, Y., Li, B., Chen, S., Li, X., Deng, H., Xu, J., Zheng, B., Wang, Q., Li, C.: Multi-scenario ranking with adaptive feature learning. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 517–526 (2023)
21. Vaswani, A.: Attention is all you need. Advances in Neural Information Processing Systems (2017)
22. Yang, X., Peng, X., Wei, P., Liu, S., Wang, L., Zheng, B.: Adasparse: Learning adaptively sparse structures for multi-domain click-through rate prediction. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4635–4639 (2022)
23. Zhang, M., Tang, Y., Hu, J., Zhang, Y.: Scenario-adaptive fine-grained personalization network: Tailoring user behavior representation to the scenario context. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1557–1566 (2024)
24. Zhang, Q., Liao, X., Liu, Q., Xu, J., Zheng, B.: Leaving no one behind: A multi-scenario multi-task meta learning approach for advertiser modeling. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. pp. 1368–1376 (2022)
25. Zhang, Y., Wang, X., Hu, J., Gao, K., Lei, C., Fang, F.: Scenario-adaptive and self-supervised model for multi-scenario personalized recommendation. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 3674–3683 (2022)
26. Zhao, P., Gao, X., Xu, C., Chen, L.: M5: Multi-modal multi-interest multi-scenario matching for over-the-top recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 5650–5659 (2023)
27. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., Gai, K.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1059–1068 (2018)
28. Zhou, J., Cao, X., Li, W., Bo, L., Zhang, K., Luo, C., Yu, Q.: Hinet: Novel multi-scenario & multi-task learning with hierarchical information extraction. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). pp. 2969–2975. IEEE (2023)
29. Zou, X., Hu, Z., Zhao, Y., Ding, X., Liu, Z., Li, C., Sun, A.: Automatic expert selection for multi-scenario and multi-task search. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1535–1544 (2022)