Scenario-aware and Mutual-based approach for Multi-scenario Recommendation in E-Commerce

Yuting Chen[1,2*] , Yanshi Wang[2*], Yabo Ni[2], An-Xiang Zeng[2], Lanfen Lin[1]
[1]Zhejiang University, Hangzhou, China
[2]Alibaba Group, Hangzhou, China
21851477@zju.edu.cn, {yanshi.wys, yabo.nyb}@alibaba-inc.com, renzhong@taobao.com, llf@zju.edu.cn

***Abstract*—Recommender systems (RSs) are essential for e-commerce platforms to help meet the enormous needs of users. How to capture user interests and make accurate recommendations for users in heterogeneous e-commerce scenarios is still a continuous research topic. However, most existing studies overlook the intrinsic association of the scenarios: the log data collected from platforms can be naturally divided into different scenarios (e.g., country, city, culture). We observed that the scenarios are heterogeneous because of the huge differences among them. Therefore, a unified model is difficult to effectively capture complex correlations (e.g., differences and similarities) between multiple scenarios thus seriously reducing the accuracy of recommendation results.**

In this paper, we target the problem of multi-scenario recommendation in e-commerce, and propose a novel recommendation model named Scenario-aware Mutual Learning (SAML) that leverages the differences and similarities between multiple scenarios. We first introduce scenario-aware feature representation, which transforms the embedding and attention modules to map the features into both global and scenario-specific subspace in parallel. Then we introduce an auxiliary network to model the shared knowledge across all scenarios, and use a multi-branch network to model differences among specific scenarios. Finally, we employ a novel mutual unit to adaptively learn the similarity between various scenarios and incorporate it into multi-branch network. We conduct extensive experiments on both public and industrial datasets, empirical results show that SAML consistently and significantly outperforms state-of-the-art methods.

***Index Terms*—Recommender Systems, E-Commerce, Personalization, Multi-Task Learning, Neural Networks**

I. Introduction

In the Internet era, large global e-commerce portals such as Amazon and AliExpress often serve customers all over the world and contain billions of items. It is particularly challenging for the recommender systems to meet the enormous needs of users with different preferences. Personalization techniques are critical for these systems in that modeling user's interests more precisely can help improve user experience and generate more business value.

In practice, the log data collected from e-commerce portals can be naturally divided into different scenarios (e.g., country, city, culture). Those scenarios are heterogeneous and there may be complex correlations between different scenarios, such as huge differences in user's interests, preferences, etc. On the contrary, in some cases users may have some similar interests as well (like countries with similar geographic locations).

There has been a large body of researches in recommender systems [1], [2]. Most of the researches are based on deep neural networks (DNNs) and recurrent neural networks (RNNs). More recently, attention mechanism [3]–[6] has been introduced for better performance. Many of these techniques have been successfully deployed in real-world applications [7]–[9]. However, existing recommendation methods mainly ignore the complex correlations between multiple scenarios and simply apply a general model for all scenarios, which may be sub-optimal as valuable information is not clearly captured across different scenarios.

Regarding the above issues, the intuitive consideration is to model for each scenario with the data of itself respectively. However, this may cause insufficient training problems on part of small-traffic scenarios and ignore the correlation between scenarios as well. In addition, Multi-Task Learning (MTL) may be a feasible solution. By treating each scenario as a separate task, a MTL scheme can be used to model the correlation between multiple scenarios, such as MMoE [10] that implicitly integrate information between relevant scenarios through MoE structure and gate units.

Different from existing research, this work focus on perceiving scenario awareness in an explicit manner. We propose **S**cenario-**a**ware **M**utual **L**earning (SAML) which targets at learning both global and scenario-specific representations across multiple scenarios simultaneously. The global representation can extract shared knowledge from various scenarios, and the scenario-specific representation can learn the specific representation of each scenario individually.

In practice, we first build both global and scenario-specific subspace for embedding and attention module, and then combine the features from each subspace respectively to construct two types of features, named scenario-independent and scenario-dependent features. Second, an auxiliary network and a multi-branch network are established upon the features to learn shared knowledge across scenarios as well as scenario-specific representations respectively. Finally, a novel mutual unit is designed and incorporated into multi-branch network to capture correlations among multiple scenarios, which not only maintain the dominance of the current scenario but also leverages the knowledge from some of the similar scenarios adaptively.

*Both authors contributed equally to this research. Work done when Yuting Chen was intern at Alibaba Group.

Extensive experiments have been conducted to verify the effectiveness of the scenario-aware mutual learning (SAML) method. Evaluations of Click-Through Rate (CTR) prediction on public and industrial datasets show that the proposed SAML can generate better performance for multi-scenario recommendation compared to most advanced recommendation methods (without an explicit perception of scenario). Furthermore, ablation studies and visualization analysis on real-world industrial datasets demonstrate the proposed SAML does have an effective generalization.

The main contributions of this paper are summarized as follows:

- Considering the feature diversity in multiple scenarios, we transform the embedding and attention module to map the features into both global and scenario-specific subspace, and then combine the feature vectors in the corresponding subspace to construct the scenario-independent and scenario-dependent features respectively.
- We propose to learn scenario-independent and scenario-dependent features separately, thus an auxiliary network, as well as a multi-branch network, are built to learn deep representations from corresponding feature spaces respectively.
- We propose to model the complex correlations between multiple scenarios in an explicit manner, thus introduce a mutual unit and incorporate it into the multi-branch network to simultaneously model the differences and similarities between scenarios, which can maintain the dominance of the current scenario and leverage the knowledge from some of the similar scenarios adaptively.
- We conduct extensive experiments on both public and industrial datasets. Experimental results show that our proposed SAML can generate more accurate results in the multi-scenario recommendation task, and can also generalize to other scenarios effectively.

The remaining parts of this paper are organized as follows: Section 2 introduces some related work. Section 3 describes and analyses the design of proposed SAML in detail. Experimental results and corresponding analysis are presented in Section 4 and conclusion in Section 5.

II. Related Work

In this section, we mainly introduce existing studies of feature representation and multi-task learning in recommender systems as well as deep mutual learning.

A. Feature Representation in Recommendation

In recommender systems, feature representation plays an important role in estimating the probability of corresponding user events, e.g., click and purchase. Enormous efforts have been put into modeling efficient features interactions and varied sequential behaviors.

Wide&Deep [11] combines the benefits of linear and deep representations, serving as a good solution for this task. DeepFM [12] replaces the wide component of Wide&Deep with factorization machines (FM) to model second-order feature interactions. DCN [13] further introduces a multi-layer residual [14] structure to learn high-order representation of features.

Besides, users' sequential behavior implies the dynamic and evolving interests and has been proven effective in tasks of user interest estimation. DIN [3] applies attention mechanism to learn the representation of users' historical behaviors towards the target item. DIEN [4] further introduces an auxiliary loss and AUGRU to capture the evolution trend of users' interests. DSIN [5] divides users' behaviors into different sessions and use self-attention to extract users' interests in each session. Most recently, BST [15] deploys Transformer to the E-commerce recommendation and verify its effectiveness.

However, these existing methods are mainly designed without consideration of multiple scenarios, thus the learned feature representation is from a global perspective and only in a homogeneous representation space, e.g., learning unified feature embedding and attention vector to express feature attributes and users' interests across various situations. This may become a bottleneck for distinguishing the interests of different users in multiple (heterogeneous) scenarios.

B. MTL in Recommendation

MTL [16], [17] have been actively researched in recommender systems and numerous deep learning applications benefit from the multi-objective optimization. DUPN [18] proposes a robust and practical representation learning framework, which learns sharing user representations in an end-to-end setting across multiple e-commerce tasks. Considering the sequential pattern of user actions, ESMM [19] introduces two auxiliary networks for CTR and CTCVR tasks, tackling the challenges of *sample selection bias* and *data sparsity* problems. ESM^2 [20] further decomposes the post-click behavior for modeling CVR task in e-commerce recommender system. MMoE [10] uses computational efficient Mixture-of-Experts (MoE) [21] as shared-bottom as well as light-weight gating network to model task relationship, which proved can better handle the scenario where tasks are less related.

In the context of our problem, although we can build individual networks for each scenario on top of a shared-bottom structure, and do multi-objective optimization as classical MTL methodology, thereby modeling complex correlations between multiple scenarios. However, when scenarios in recommender system share the same item candidates and label space, the consistency and discrepancy of scenarios are coupled with each other tightly, thus the sophisticated relationship between different scenarios are hard to capture.

C. Deep Mutual Learning

Deep mutual learning [22] is proposed for knowledge distillation [23], which builds an ensemble of student networks to teach each other with distillation losses of Kullback-Leibler divergence. Inspired by this learning strategy, we explore a different idea to solve multi-scenario problem with a novel mutual unit to learn collaboratively scenario correlations in

recommender systems. The essential difference is that our dataset shares a consistent label domain and the mutual unit uses a tailored similarity and gate mechanism to control the learning process instead of indirect approximation of data distribution by distillation.

III. Methods

In this section, we elaborate on the design of Scenario-aware Mutual Learning (SAML) model. First, we recapitulate the basic structure of deep learning based recommendation model from two aspects: feature representation and multi-layer perceptron. And then we introduce the overall structure of SAML corresponding to the above two aspects respectively.

A. Feature Representation

There are four categories of features in our recommender system: *User Profile*, *Item Profile*, *User Behavior* and *Context*. Each category of feature has several fields, *User Profile* contains *user_id*, *gender*, *age* etc.; *Item Profile* contains *item_id*, *shop_id*, *price*, etc.; *User Behavior* is the sequential list of user behavior, which contains the user's interacted items with corresponding features such as *item_id*, *shop_id*, etc.; *Context* contains *time*, *matchtype*, *scenario* and so on.

1) Embedding Module: Features in each field are numerical value or categorical id. For numerical features, we use normalization to transform them into the same scale. For categorical features, which typically represented by one-hot vectors, we use embedding technology to transform them into low-dimension dense vectors. For example, the embedding matrix of *item_id* can be represented by $E_{item} = [e_1; e_2; ...; e_K] \in R^{N \times K}$, where N is the total number of different items, K is the dimension size of the embedding, $e_i \in R^K$ represents an embedding vector with dimension K.

2) Attention Module: Most of the advanced recommender systems use attention mechanism to capture user interests, especially on the representation learning of user sequential behavior. We follow BST [6] and use multi-head self-attention [15] to learn deep representation of user interests based on the *User Behavior* features, which can be formulated as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, ..., head_H)W^O \quad (1)$$

$$\begin{aligned} head_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ &= \text{Softmax}(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}})VW_i^V \end{aligned} \quad (2)$$

where Q, K, V are embedding matrices of *User Behavior* features, which are converted through linear projection. H denotes the number of attention heads, d is the last dimension of embedding, W_i^Q, W_i^K, W_i^V, W^O are all linear projection matrices.

B. Multi-layer Perceptron (MLP)

As most of the recent deep models in recommender systems, the output of *Embedding Module* and *Attention Module* are concatenated, then fed into MLP with fully connected layers for final prediction.

The widely used loss function in recommendation is negative log-likelihood function, which can be defined as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{(x,y) \in D} (y \log p(x) + (1 - y) \log(1 - p(x))) \quad (3)$$

where x is the training sample, $y \in \{0, 1\}$ is the corresponding label, represents whether user clicks target item. $p(\cdot)$ is the predicted output of model.

C. Scenario-aware Feature Representation

Most existing recommendation methods mainly ignore the complex correlation between multiple scenarios, we first build both global and scenario-specific subspace for embedding and attention module, and then combine the features from each subspace respectively, thus construct two types of feature which we named scenario-independent and scenario-dependent features.

1) Embedding Module: As depicted in Figure 1, the embedding module explicitly embeds each feature into both global and scenario-specific subspace in parallel. And then combine the feature vectors from each subspace respectively to construct two types of embedding vectors. The motivation is to realize the perception of features to the global and specific scenarios. As a comparison, that is more effective than directly increasing the embedding size, because no matter how to expand the dimension, the information between various scenarios is still mixed together without distinction. While our method has an explicit distinction between different scenarios. The ablation study in section IV-H also confirms this.

2) Attention Module: On top of the embedding module, the attention module is subsequently enriched to capture use interests both in global and specific scenarios. According to formula 1, we can define the formula here as: MultiHead(Q_g, K_g, V_g) and MultiHead(Q_l, K_l, V_g), where subscript g and l indicate the source of embedding vector (i.e., global and scenario-specific). It is notable that we share the same embedding vector V_g above but the attention weights are computed from each scenario separately, which bridges the universal and specific knowledge as well as leveraging the rich diversity of user behaviors in global.

D. Scenario-mutual Network

The scenario-mutual network contains two subnetworks: an auxiliary network and a multi-branch network, which are used to learn the scenario-independent and scenario-dependent features in parallel. In addition, a novel mutual unit is incorporated in the multi-branch network to model the complex correlations (e.g., differences and similarities) between multiple scenarios in an explicit manner. Now we introduce these three components in detail.

1) Auxiliary Network: In MMoE, all experts learn the shared knowledge together, if a definite domain knowledge exists for each task, integrating it into the expert is not convenient in practice. Therefore, we build the auxiliary network on top of scenario-independent features, which is used to learn shared knowledge from a global perspective. Specifically, we not

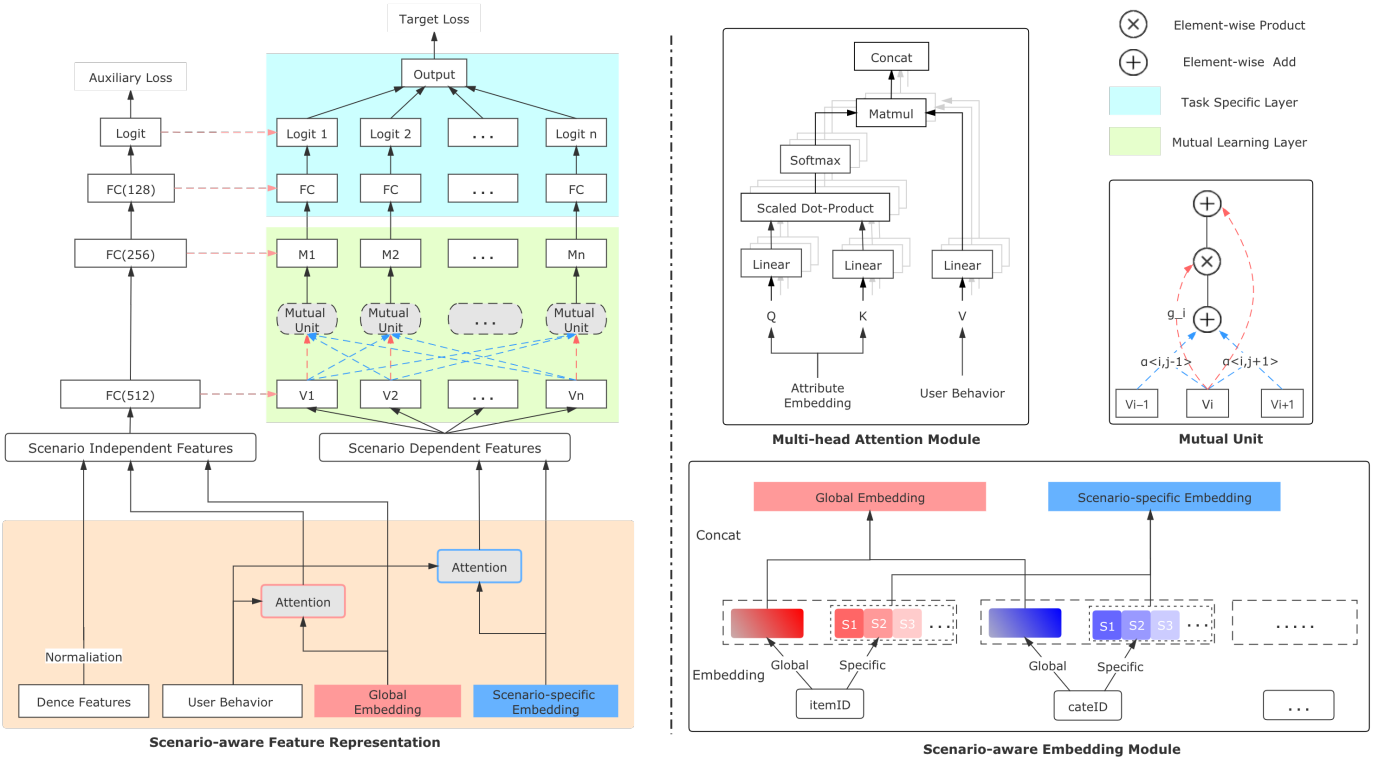


Fig. 1: The overview of our proposed SAML model. The left part describes the model structure, the right part makes detail description of the modules involved. From the bottom up, SAML has two main components: scenario-aware feature representation and scenario-mutual network. In the first component, the raw features first go through the embedding module to obtain global and scenario-specific embedding vectors. Then part of features will enter the attention module to calculate the weights, and multiply the embedding of the user behavior to obtain the attention vector. Finally, the feature vectors are concatenated and then fed into the two sub-networks of scenario-mutual network respectively. The detail description of scenario-mutual network will be introduced in section III-D.

only obtain its final output (for supervised learning), but also extract its hidden layer representation and use as an additional input to the multi-branch network. As shown in Figure 1, the knowledge propagates from auxiliary network to multi-branch network unidirectionally. The advantage is that we can extract some universal and scenario-independent knowledge to enhance the global perception of each specific scenario. An auxiliary loss function of negative log-likelihood is used to supervise the learning process.

2) Multi-branch Network: As shown in Figure 1, the input of each layer in multi-branch network contains two parts: output from the previous layer and hidden layer representation transferred from the auxiliary network. Taking the i th branch and l th layer as example, the process can be formulated as

$$\begin{aligned} V_a^l &= \delta(V_a^{l-1} W_a^l + b_a^l) \\ V_{m_i}^l &= \delta([V_{m_i}^{l-1}, V_a^l] W_{m_i}^l + b_{m_i}^l) \end{aligned} \tag{4}$$

where V_a^l is the l th layer output from auxiliary network, $V_{m_i}^l$ is the l th layer output from i th branch in multi-branch network, W_a^l, b_a^l, $W_{m_i}^l$, $b_{m_i}^l$ are the corresponding weight and bias, δ is the activation function.

In order to make each branch clearly correspond to a specific scenario, and can only be optimized by the data of the scenario itself, we implement this by adding a mask on the connection between networks to stop the gradient back-propagation. Thus the gradient of instance t which belongs to scenario S_i will only back-propagate to update the parameters in branch i. Finally, the total loss can be calculated as:

$$\begin{aligned} \mathcal{L}_{total} &= \mathcal{L}_{target} + \mathcal{L}_{aux} \\ &= \sum_i^N \mathcal{L}_i^t \cdot I_i^t + \mathcal{L}_{aux} \\ where \quad I_i^t &= \begin{cases} 1 & if\ t \in S_i \\ 0 & otherwise \end{cases} \end{aligned} \tag{5}$$

where $\mathcal{L}_{target}$ and $\mathcal{L}_{aux}$ are the losses of multi-branch network and auxiliary network respectively, N is the number of branches, t is the training sample, S_i is the data set of the i th scenario, $\mathcal{L}_i^t$ is the loss of the i th branch on sample t, $I(\cdot)$ is used as indicator function to constraint the gradient.

3) mutual unit: By combining the learned representation from auxiliary and multi-branch network, we can take advantage of both global and scenario-specific features. However, the branches are independent of each other, which means that we actually isolate the relationship between scenarios and ignore the fact that similarity exists between part of scenarios to some extent. In order to simultaneously model the differences and similarities between scenarios, we introduce a novel mutual unit that can enhance the representation learning

by considering the similarity among multiple scenarios and alleviating the problem of insufficient training on part of scenarios as well.

As shown in Figure 1, the mutual unit use the hidden layer output V_i from i th branch to calculate the cosine distance with other $V_{j,j\neq i}$ to capture the similarities between different scenarios. A light-weight gate network is designed to control the degree learning from other similar scenarios. The learning procedure can be defined as follows:

$$M_i = V_i + g_i * \sum_{j=1,j\neq i}^{N} (\alpha_{ij} * V_j) \tag{6}$$

$$g_i = \text{Sigmoid}(W_i V_i + b_i) \tag{7}$$

$$\alpha_{ij} = \text{Softmax}(\frac{\cos < V_i, V_j >}{\sum_{j=1,j\neq i}^{N} \cos < V_i, V_j >}) \tag{8}$$

where $V_i \in \mathbb{R}^D$, D is the dimension of the hidden layer output of each branch, $g_i \in \mathbb{R}$ is the gate coefficient learned from V_i. $\alpha_{ij} \in \mathbb{R}$ is the normalized similarity coefficient indicating the similarity between scenario i and j. $W_i \in \mathbb{R}^D$, $b_i \in \mathbb{R}$ are the linear matrix and bias of gate network.

Finally, M_i of each branch are sent to the next layer respectively, which benefits from the assistance of the similar scenarios, having the advantages of: (1) it maintain the dominance of the current branch to the greatest extent, so it can accurately model the differences between scenarios; (2) it can leverage the knowledge from some of the similar scenarios to enhance itself adaptively and vice versa. Note that when gate coefficient g equal to 0, the similarity between scenarios is not considered, thus the network degenerates into multi-branch network with independent branches.

IV. Experiments and Analysis

In this section, we present our experiments in detail, including datasets, competitors, experimental setup, evaluation metrics, and the corresponding analysis. The experiments are intended to answer the following questions:

- **Q1**: How does our proposed model SAML compare with state-of-the-art methods on the recommendation task?
- **Q2**: Does SAML really help to improve the recommendation results on each scenario?
- **Q3**: How is the effectiveness of critical technical designs in SAML?
- **Q4**: How do different experimental settings (i.e., embedding size, number of attention heads, etc.) influence the performance of SAML?
- **Q5**: How does SAML provide effective recommendation results intuitively?

A. Datasets

We conduct experiments on public and industrial datasets respectively, both of them are collected from real-world e-commerce platforms. Table I summarizes the statistics of datasets used in this paper.

Public Dataset[1]**.** It's a public dataset released by Alimama, an online advertising platform in China. The dataset consists of 8 days of ad display/click logs from 2017-05-06 to 2017-05-12. We use the first 7 days for training and the last day for testing, and set behavior sequence length to 15. We filter the samples of which user profile is missing, and then divide the scenario according to *City_level*.

Industrial Dataset. It's an industrial dataset collected from the online recommender system of AliExpress, a cross-border e-commerce platform. Logs from 2019-08-24 to 2019-08-30 are used for training and 2019-08-31 is for testing, users' recent 30 behaviors are also recorded in logs. In this dataset, more than two hundred countries are contained, the performance of each country varies greatly, thus we divide the scenario according to *Country_id*.

B. Competitors

To evaluate the performance of proposed method, we compare SAML with the following models:

- **Wide&Deep** [11] contains a wide part for memorization and a deep part for generalization, we implement the wide part by linear regression (LR) and the deep part by MLP.
- **DeepFM** [12] replace LR with factorization machines (FM) and combine with MLP to model low- and high-order feature interactions. The two components share embedding space and summed up their outputs as the final prediction.
- **DCN** [13] introduces a novel cross network that is more efficient in learning certain bounded-degree feature interactions. Our implementation follows the same structure as original paper and stacks only three cross layers.
- **DIN** [3] represents user interest with regard to the target item by adaptively learning the attention weight.
- **MMoE** [10] is a multi-task learning approach that use Multi-gate Mixture-of-Experts to model task relationships from data. We implement each expert with a two-layer MLP to learn representation from multiple scenarios.
- **BST** [6] utilizes the powerful Transformer model to capture the sequential signals underlying users' behavior sequences for recommendation. We regard it as a base model for comparison in this paper.

C. Experimental Setup

We implement our experiments on a distribution TensorFlow framework[2]. All competitors in the experiments use ReLU activation function and Adam [24] optimizer. For each dataset, their settings are as follows:

Public settings. Learning rate is tuned and set to be 5e-4, mini-batch size is set to 128. The hidden layer size of MLP involved are set by 128 $\times$ 64. The size of global embedding and scenario-specific embedding for each attribute is set to 12 and 4 respectively. The number of attention heads is set to 4. Corresponding to *Country_level* attribute, the scenarios are divided into 5 parts.

Industrial settings. Learning rate is tuned and set to be 1e-4, mini-batch size is set to 1024. The hidden layer size of MLP

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

[2]https://www.aliyun.com/product/bigdata/product/learn

TABLE I: STATISTICS OF PUBLIC AND INDUSTRIAL DATASETS

Dataset	User	Item	Click	Conversion	Samples
Public	1.32M	1.08M	1.23M	-	24.6M
Industrial	35.9M	111M	307M	1.90M	2.85B

involved is set by 256×128. The size of global embedding and scenario-specific embedding for each attribute are set to 8 and 2 respectively. The number of attention heads is set to 4. Corresponding to *Country_id* attribute, the traffic of top 9 countries exceeds sixty percent, thus we select top-9 countries with the highest traffic volume and group the rest into one.

D. Evaluation Metrics.

AUC. We use AUC (Area under ROC Curve) as our metric for measurement of model performance. It is defined as:

$$AUC = \frac{1}{|D^+||D^-|} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} I(f(x^+) > f(x^-)) \quad (9)$$

where D^+ and D^- donate the collection of positive and negative samples respectively, $|D^+|$ and $|D^-|$ donate the number of samples in D^+ and D^-, $f(\cdot)$ is the output of model, $I(\cdot)$ is the indicator function.

RelaImpr. We follow [25] to introduce RelaImpr metric to measure relative improvement over models. For a random guesser, the value of AUC is 0.5. Hence RelaImpr is defined as below:

$$RelaImpr = \left(\frac{AUC(measured\ model) - 0.5}{AUC(base\ model) - 0.5} - 1 \right) \times 100\% \quad (10)$$

E. Overall Performance (Q1)

We conduct experiments of multi-scenario recommendation on both public and industrial datasets. The corresponding results are present in table II, which are recorded from comparison models learning through the whole scenarios, and then testing the overall performance.

From above results, we have several important observations: (1) DIN performs better than previous models, mainly attributed to the attention mechanism which captures the user interest with regard to target item; (2) MMoE performs better than DIN and previous models, especially on industrial dataset of which expert networks can implicitly model correlations between scenarios, indicating that discrepancy between scenarios can not be neglected; (3) BST is a strong competitor, and is just slightly worse than MMoE, indicating the effectiveness of incorporating Transformer into the recommendation model; (4) SAML achieves the best performance on both datasets, demonstrates its effectiveness and generalization capability on multiple scenarios. Note that on industrial dataset, SAML achieves 0.0096 absolute AUC gain over BST, which is a significant improvement for business.

TABLE II: MODEL COMPARISON ON PUBLIC AND INDUSTRIAL DATASETS

Model	Public		Industrial	
	AUC	RelaImpr[a]	AUC	RelaImpr[a]
W&D	0.6320	-2.07%	0.7245	-7.61%
DeepFM	0.6329	-1.40%	0.7308	-5.02%
DCN	0.6333	-1.11%	0.7308	-5.02%
DIN	0.6343	-0.37%	0.7409	-0.86%
MMoE	0.6357	0.66%	0.7449	0.78%
BST	0.6348	0.00%	0.7430	0.00%
SAML	**0.6392**	**3.26%**	**0.7526**	**3.95%**

[a]RelaImpr is based on BST.

TABLE III: SINGLE SCENARIO COMPARISON (AUC) ON INDUSTRIAL DATASET

Scenario	BST-Individual	BST	SAML
RU	0.7391	0.7475	0.7574
BR	0.7399	0.7572	0.7683
ES	0.7261	0.7499	0.7600
US	0.7128	0.7407	0.7517
FR	0.7219	0.7502	0.7592
PL	0.7199	0.7496	0.7608
NL	0.7083	0.7403	0.7520
CL	0.7107	0.7466	0.7550
UA	0.7051	0.7429	0.7500
Others	0.7356	0.7421	0.7523

F. Single Scenario Performance (Q2)

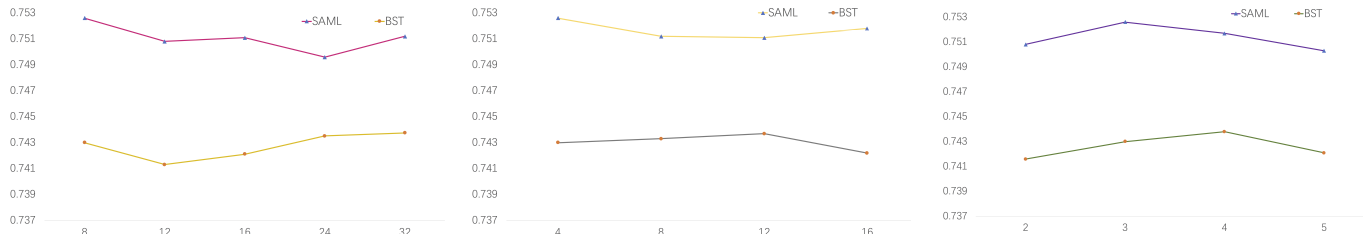
To verify whether our proposed SAML can really help to improve the recommendation results in each scenario, we take BST and SAML as comparison, and conduct experiments on industrial dataset according to the following procedure: (1) BST-Individual trains with the data of each scenario individually; (2) BST trains with the data combined from all scenarios; (3) SAML trains with the data combined from all scenarios and incorporates the critical technical designs we proposed above. And then all the models are tested on each scenario separately.

Table III shows the comparison results of several models in each scenario. According to the results, we have several observations: (1) By learning from multiple scenarios, BST is better than BST-Individual, indicating that the information between different scenarios can be mutually used to promote each other; (2) SAML consistently outperforms BST on each scenario, demonstrating the effectiveness of our scenario-aware mutual learning approach, which explicitly consider the differences and similarities between scenarios.

TABLE IV: ABLATION RESULTS OF VARIANT MODELS ON INDUSTRIAL DATASET

Model	AUC	RelaImpr[a]
SAML	**0.7526**	**3.95%**
SAML w/o gate	0.7489	2.42%
SAML w/o aux	0.7472	1.72%
SAML w/o gate&mut	0.7457	1.11%
BST	0.7430	0.00%

[a]RelaImpr is based on BST.



(a) embedding size (b) attention heads (c) depth of network

Fig. 2: The results of different experimental settings in BST and SAML.

G. Ablation Study of Critical Technical Designs (Q3)

To further verify the effectiveness of critical technical designs in SAML, such as the effectiveness of learning scenario-aware feature representation and the benefits of modeling differences and similarities between scenarios, we conduct ablation experiments to compare SAML with the following variant models:

- SAML w/o gate: removes the mutual unit between scenarios, equal to fix the gate coefficient g as 0, which means that it will not consider the similarity between scenarios.
- SAML w/o aux: removes the auxiliary network upon scenario-independent features, combine the two features together then fed into mutual network, thus the differences between scenarios are not fully considered.
- SAML w/o gate&mut: removes the mutual network upon scenario-dependent features, thus the features are combined and fed into auxiliary network (a unify MLP), which means that it considers neither similarities nor differences between scenarios.

Table IV shows the performance of SAML and different variants, the state-of-the-art recommendation model BST is also included as a base model. Based on the experiment results, we have the following observations:

- SAML w/o gate performs worse than SAML about 0.0037 absolute AUC, indicating the effectiveness of mutual unit which can automatically select similar representations from other scenarios for enhancement.
- SAML w/o aux declines by about 0.0054 absolute AUC than SAML, demonstrating the effectiveness of separation of scenario-specific representation learning from the global representation learning.
- From the comparison between SAML w/o gate&mut and BST, we can get a rough view of the contribution from scenario-aware feature representation, which achieves 0.0024 absolute AUC improvement than base model.

H. Ablation Study of Experimental Settings (Q4)

In addition to the ablation study of critical technical designs in SAML, we also study the sensitivity of the model to different experimental settings includes embedding size, number of attention heads, and depth of network layers.

The embedding and attention module in scenario-aware feature representation are different from directly increasing the embedding size or number of attention heads in that the feature diversity between scenarios are explicitly considered. However, one may question that the reason for the improvement we obtained may be purely due to the increase in embedding size or attention heads rather than the perception of scenario. Therefore, we compared the two models of BST and SAML, first fixed other settings, and then tune the embedding size and attention heads from 8, 12, 16, 24, 32 and 4, 8, 12, 16, respectively. Similarly, we also compared different depths of network layers, including 2, 3, 4, and 5, respectively.

From the results in Figure 2(a) and Figure 2(b), we can see that SAML model incorporates with scenario awareness consistently outperform the comparison model in various setting of embedding size and attention heads, while the improvement of purely increasing the embedding size or attention heads are all very limited. We consider that since increasing the embedding size or attention heads do have improvement to some extent, while the lack of scenario awareness may limit the expression capability of each attribute, hence resulting in limited improvement.

Increasing the depth of network layers can enhance the model capacity but also potentially leads to over-fitting. As can be seen from Figure 2(c), at the beginning stage, i.g., from two layers to four layers, increasing the number of hidden layers consistently improves the model's performance. However, it saturates at five layers that increasing more layers even marginally decreases the AUC scores, where the model may overfit the training set. Therefore, we use three/two hidden layers for SAML in industrial/public dataset.

I. Visualization Analysis (Q5)

The key to understanding how SAML provides effective recommendation results is to understand how mutual units help optimizing scenario correlation and enhance representation learning for each scenario. Thus we conduct experiments on industrial dataset to visualize the accumulated probability of α and g in mutual units. The similarity coefficient α is used to learn the similarity between scenarios and the gate coefficient g is used to control the degree of learning from other similar scenarios, which is based on the learning situation of each scenario itself.

Taking Spain (ES) and Ukraine (UA) as example, the distributions of α are dramatically different as shown in Figure 3. The relevance of ES to Brazil (BR) is much higher than that

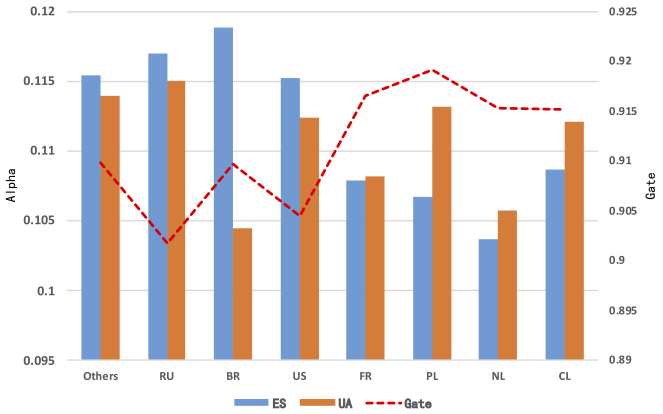


Fig. 3: Visualization of α and g in mutual unit.

of UA because of the sport-related categories. In contrast, the similarity of UA with Poland (PL) is higher than that of ES for the geographical location. What's more, both of them are similar to Russia (RU), United States (US) and Others because these three scenarios have relatively larger volumes of traffic to affect model learning.

The gate coefficient g and the similarity coefficient α show a certain correlation, which further validates our assumptions: (1) scenarios with more traffic such as RU and BR can learn better representation through themselves thus tend to suppress the g to reduce the effect from other scenarios; (2) scenarios with low traffic such as FR, NL, etc., due to insufficient learning of their own representations, thus tend to increase the g to rely more on the representations of other similar scenarios.

V. Conclusion

In this paper, a novel recommendation model named Scenario-aware Mutual Learning (SAML) is proposed in the field of e-commerce recommendation to capture the complex correlations (e.g., differences and similarities) between multiple scenarios. First, we introduce scenario-aware feature representation to learn feature representations both in global and scenario-specific. Then we introduce an auxiliary network to model the shared knowledge across all scenarios, and use a multi-branch network to model the differences among specific scenarios. Finally, we employ a mutual unit to adaptively learn the similarity of user's interests between various scenarios. An extensive set of experiments are provided to show the competitive performance of SAML and transferability of the learning framework. Detailed discussion of ablation studies and visualization analysis are also provided to show the insight of how the SAML works in real-world datasets.

References

[1] Y. Zhang, Q. Ai, X. Chen, and W. B. Croft, "Joint representation learning for top-n recommendation with heterogeneous information sources," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 1449–1458.

[2] J. Lian, F. Zhang, X. Xie, and G. Sun, "Towards better representation learning for personalized news recommendation: a multi-channel deep fusion approach." in *IJCAI*, 2018, pp. 3805–3811.

[3] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1059–1068.

[4] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5941–5948.

[5] Y. Feng, F. Lv, W. Shen, M. Wang, F. Sun, Y. Zhu, and K. Yang, "Deep session interest network for click-through rate prediction," *arXiv preprint arXiv:1905.06482*, 2019.

[6] Q. Chen, H. Zhao, W. Li, P. Huang, and W. Ou, "Behavior sequence transformer for e-commerce recommendation in alibaba," in *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, 2019, pp. 1–4.

[7] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers *et al.*, "Practical lessons from predicting clicks on ads at facebook," in *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 2014, pp. 1–9.

[8] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*. ACM, 2016, pp. 191–198.

[9] F. Borisyuk, L. Zhang, and K. Kenthapadi, "Lijar: A system for job application redistribution towards efficient career marketplace," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1397–1406.

[10] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1930–1939.

[11] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 2016, pp. 7–10.

[12] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.

[13] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*. ACM, 2017, p. 12.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[16] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[17] A. Argyriou, M. Pontil, Y. Ying, and C. A. Micchelli, "A spectral regularization framework for multi-task structure learning," in *Advances in neural information processing systems*, 2008, pp. 25–32.

[18] Y. Ni, D. Ou, S. Liu, X. Li, W. Ou, A. Zeng, and L. Si, "Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 596–605.

[19] X. Ma, L. Zhao, G. Huang, Z. Wang, Z. Hu, X. Zhu, and K. Gai, "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 1137–1140.

[20] H. Wen, J. Zhang, Y. Wang, W. Bao, Q. Lin, and K. Yang, "Conversion rate prediction via post-click behaviour modeling," *arXiv preprint arXiv:1910.07099*, 2019.

[21] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton *et al.*, "Adaptive mixtures of local experts." *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[22] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[23] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] L. Yan, W.-J. Li, G.-R. Xue, and D. Han, "Coupled group lasso for web-scale ctr prediction in display advertising," in *International Conference on Machine Learning*, 2014, pp. 802–810.