Scenario-Adaptive and Self-Supervised Model for Multi-Scenario Personalized Recommendation

Yuanliang Zhang*
Alibaba Group
Hangzhou, China
kubert.zyl@alibaba-inc.com

Xiaofeng Wang*
Alibaba Group
Hangzhou, China
aron.wxf@alibaba-inc.com

Jinxin Hu
Alibaba Group
Hangzhou, China
jinxin.hjx@alibaba-inc.com

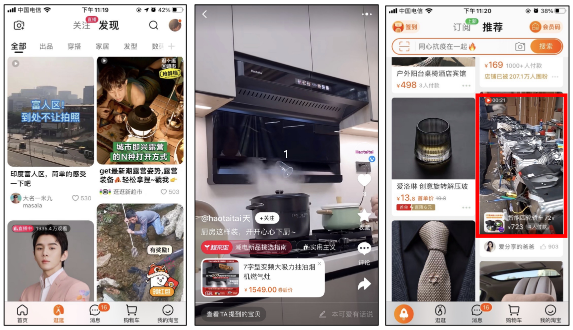
Ke Gao
Alibaba Group
Hangzhou, China
gaoke.gao@alibaba-inc.com

Chenyi Lei
Alibaba Group
Hangzhou, China
chenyi.lcy@alibaba-inc.com

Fei Fang
Alibaba Group
Hangzhou, China
mingyi.ff@alibaba-inc.com

ABSTRACT

Multi-scenario recommendation is dedicated to retrieve relevant items for users in multiple scenarios, which is ubiquitous in industrial recommendation systems. These scenarios enjoy portions of overlaps in users and items, while the distribution of different scenarios is different. The key point of multi-scenario modeling is to efficiently maximize the use of whole-scenario information and granularly generate adaptive representations both for users and items among multiple scenarios. we summarize three practical challenges which are not well solved for multi-scenario modeling: (1) Lacking of fine-grained and decoupled information transfer controls among multiple scenarios. (2) Insufficient exploitation of entire space samples. (3) Item's multi-scenario representation disentanglement problem. In this paper, we propose a **S**cenario-**A**daptive and **S**elf-**S**upervised (**SASS**) model to solve the three challenges mentioned above. Specifically, we design a **M**ulti-**L**ayer **S**cenario **A**daptive **T**ransfer (**ML-SAT**) module with scenario-adaptive gate units to select and fuse effective transfer information from whole scenario to individual scenario in a quite fine-grained and decoupled way. To sufficiently exploit the power of entire space samples, a two-stage training process including pre-training and fine-tune is introduced. The pre-training stage is based on a scenario-supervised contrastive learning task with the training samples drawn from labeled and unlabeled data spaces. The model is created symmetrically both in user side and item side, so that we can get distinguishing representations of items in different scenarios. Extensive experimental results on public and industrial datasets demonstrate the superiority of the SASS model over state-of-the-art methods. This model also achieves more than 8.0% improvement on Average Watching Time Per User in online A/B tests. SASS has been successfully deployed on multi-scenario short video recommendation platform of Taobao in Alibaba.



A:Main Feed B:Immersive Feed C:Homepage Feed

Figure 1: Short video recommendation scenarios in Taobao.

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

Recommendation System;Multi-Scenario Learning;Scenario-Adaptive; Self-Supervised Learning

ACM Reference Format:
Yuanliang Zhang, Xiaofeng Wang, Jinxin Hu, Ke Gao, Chenyi Lei, and Fei Fang. 2022. Scenario-Adaptive and Self-Supervised Model for Multi-Scenario Personalized Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557154

1 INTRODUCTION

In recent years, the *Multi-Scenario Personalized Recommendation Problem*[4, 17, 29, 30, 38], which focuses on retrieving relevant candidates in multiple scenarios, is well-known and ubiquitous in most industrial recommendation systems, such as Taobao in Alibaba, Amazon, TikTok etc. The *scenario* can be treated as a specific recommendation domain of users and items. As shown in Figure 1, there are diverse scenarios in short video recommendation platform of Taobao. From the user perspective, a user may access some of the scenarios and watch different videos. Form the video perspective, a short video may be pushed to different users in different scenarios.

*Both authors contributed equally to this research.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557154

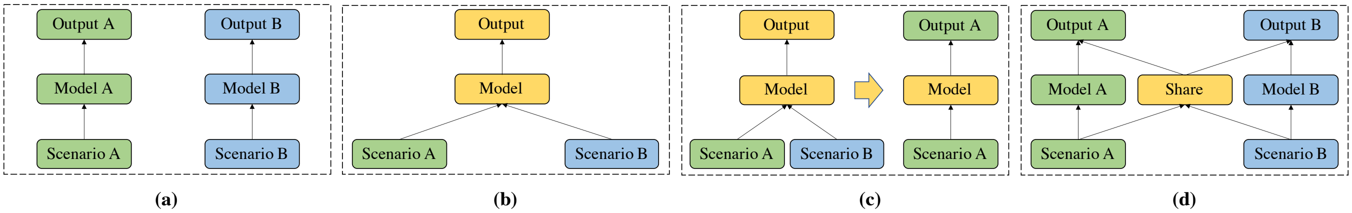


Figure 2: Overview of existing strategies for multi-scenario modeling. (a) Training a single model for each scenario with its own data. (b) Training a common model using multi-scenario data. (c) Pre-training a common model with multi-scenario data and fine-tuning with single scenario data for each scenario. (d) Building a unified model with the framework of multi-task learning.

There are some common users and videos among different scenarios, making it reasonable and beneficial to share information for model learning. However, each scenario will also have its own unique users and videos. Besides, The behaviors are diverse for the same user in different scenarios, and the exposure abilities of the same video in different scenarios are also distinguishing. Therefore, it is challenging to model the commonalities and distinctions of different scenarios when solving the multi-scenario problem.

Various types of strategies have been proposed to tackle the multi-scenario problem: (1) Training a separate model for each scenario (depicted as Figure 2(a)). The shared information among multiple scenarios is neglected in this strategies. It is challenging for new and minor scenarios with limited data to learn proper models. Besides, developing a separate model for each scenario will consume tremendous resources. (2) Training one common model with mixture of samples coming from multiple scenarios (depicted as Figure 2(b)). The operation of sample mixture destroys the original data distribution, making it hard for one model to make appropriate predictions for each scenario. Moreover, minor scenario may be dominated by major scenario. (3) Training a model with whole-scenario samples and fine-tuning scenario-specific models with samples from corresponding scenario, depicted as Figure 2(c). This method can enjoy whole-scenario data and get scenario-specific results. However, little attentions are paid to model the correlations among multiple scenarios. (4) Building a unified framework based on multi-task learning and learning commonalities and correlations among multiple scenarios[17, 29, 38] (depicted as Figure 2(d)), which has become a mainstream strategy to solve multi-scenario problem. However, previous works overlook three crucial challenges:

- **Lacking of fine-grained and decoupled information transfer controls**. The key point of learning scenario correlations is to model the information transfer among scenarios. Existing works conduct information transfer in quite implicit and coarse-grained ways, such as parameter factorization[30], mixture-of-experts mechanisms[17, 29] and dynamic weights network[38]. it is hard to determine the precise magnitude of transferred information from one scenario to another.
- **Insufficient exploitation of entire space samples**. Most existing multi-scenario models are trained only with labeled data, making huge amount of unlabeled data (users or items without interactive behaviors in a specific scenario) under no use. These unlabeled data are of great potential for reflecting scenario characteristics. It is also challenging to model some spare or new scenarios with little labeled data. Although ZEUS[9] has made some attempts to model unlabeled data through a self-supervised *Next-Query Prediction* task, it only focuses on query-side modeling. Besides, there are little unions between the self-supervised task and multi-scenario modeling in ZEUS.
- **Item's multi-scenario representation disentanglement problem**. From the perspective of item side, an item may have distinguishing characteristics and behaviors in different scenarios. It is quite suitable to generate distinguishing representations for items in different scenarios. To the best of our knowledge, previous methods mainly focus on scenario-aware intention modeling from user perspective, with little concerns on the item side.

In order to tackle the challenges mentioned above, we propose the Scenario-Adaptive and Self-Supervised (SASS) model and demonstrate its rationality and effectiveness. We design a Multi-Layer Scenario Adaptive Transfer (ML-SAT) module to learn user and item representations in scenario-specific scope and whole-scenario scope. In ML-SAT a scenario-adaptive gate unit with explicit gate controls is proposed to regulate and facilitate the fine-grained information transfer from global shared network to scenario-specific network. To adequately exploit the entire space samples, we introduce a two-stage training process (pre-training and fine-tune). In pre-training stage, we build a scenario-supervised contrastive learning task with the training samples drawn from labeled and unlabeled data spaces. In fine-tune stage, the user-item similarity matching objective is achieved with a scenario-specific and a global-auxiliary matching task. The model architecture of SASS is created symmetrically both in user side and item side, so that we can get distinguishing representations for items in different scenarios. We creatively combine the multi-scenario problem and self-supervised contrastive learning problem in a unified paradigm and tackle the challenges of multi-scenario modeling mentioned above.

The main contributions of this work are summarized as follows:

- We propose an effective Scenario-Adaptive and Self-Supervised (SASS) framework to solve multi-scenario problem. In SASS, we design a Multi-Layer Scenario Adaptive Transfer module with a scenario-adaptive gate unit to regulate and facilitate the fine-grained information transfer from global shared network to scenario-specific network.
- We introduce a two-stage training process to strengthen the exploitation of entire space samples, especially unlabeled data in corresponding scenario. We design a novel scenario-supervised contrastive learning task and closely correlate the scenario-supervised task with multi-scenario problem.

- The proposed model structure is symmetrically designed in item side, so that we can generate distinguishing representations for items in different scenarios.
- Extensive experimental results on public and industrial datasets demonstrate the superiority of the SASS model over state-of-the-art methods. SASS has also been successfully deployed on multi-scenario short video recommendation platform of Taobao in Alibaba and achieved more than 8.0% improvement on Average Watching Time Per User in online A/B tests. We believe the strategies proposed in SASS are universally applicable in most multi-scenario recommendation systems.

2 RELATED WORK

2.1 Multi-Scenario Recommendation Models

As mentioned above, the mainstream strategies of tackling the multi-scenario problem is to create a unified framework to model all scenarios simultaneously. Thus, we mainly survey works related to this paradigm. Specifically, HMoE[17] utilizes multi-gate mixture-of-experts[20] to implicitly model commonalities and distinctions among multiple scenarios. SAML[4] distinguishes user behaviors in different scenarios with scenario-specific attention mechanisms and proposes a scenario-mutual unit to learn differences and similarities between scenarios. ICAN[35] treats each data channel as a scenario and designs a scenario-aware contextual attention layer to generate distinguishing user representations in different scenarios. SAR-Net[29] proposes a unified multi-scenario architecture and introduces two scenario-aware attention modules to extract scenario-specific features in user side. After that, SAR-Net implements implicit scenario information transfer with the gate fusion of scenario-specific experts and scenario-shared experts. STAR[30] designs a star topology framework, with one centered network to maintain whole-scenario commonalities and a set of domain-specific networks to distinguish scenario distinctions. The combination strategy of element-wise product of layer weights is treated as the information transfer mechanism from whole scenarios to individual scenario. M2M[38] pays attention to advertiser modeling in multiple scenarios and proposes a dynamic weights meta unit to model inter-scenario correlations. The methods mentioned above learn scenario information transfer in quite implicit ways, making it hard to determine the precise impacts among multiple scenarios. Besides, these models are only trained with labeled data, without making full use of the entire space samples. Although ZEUS[9] tackles the feedback loop problem with a *Next-Query Prediction* self-supervised manner on unlabeled data, the pre-training task is mainly based on user spontaneous query sequences. Moreover, all of the previous works overlook the problem of generating distinguishing representations of items in different scenarios.

2.2 Self-Supervised Learning in Recommendation System

Self-supervised learning which focuses on learning feature embeddings and initial network weights on unlabeled data, is widely utilized in areas of Compute Vision[2, 3, 27] and Natural Language Processing[7, 10]. Many efforts based on self-supervised learning have also been made in the area of Recommendation Systems. [21] proposes a sequence-to-sequence self-supervised training strategy for sequential recommendation, while [31] models user behavior sequences by predicting the random masked items in the sequence with bidirectional self-attention. SGL[33] proposes a self-supervised task on user-item graph with various operators to generate different graph views. To tackle the label sparsity problem, [36] introduces the self-supervised strategy based on the contrastive learning on the item side. CLRec[39] also utilizes contrastive learning to reduce the exposure bias in deep candidate generation. S^3-Rec[40] models sequence recommendation by building four auxiliary self-supervised objectives with the mutual information maximization principle. Moreover, ZEUS[9] proposes a *Next-Query Prediction* self-supervised task on user's query sequences for spontaneous search Learning-To-Rank tasks.

3 PROBLEM FORMULATION

We propose a unified framework to solve multi-scenario problem, which is definitely applicable to serve both matching and ranking tasks in recommendation system. In this paper, we describe and demonstrate our method in matching task. Few modifications can be made when it is served for ranking tasks, which we remain for future research.

3.1 Multi-Scenario Matching Problem

A **Scenario** can be treated as a specific recommendation domain, denoted as D_s in this paper. Given a set of scenarios $D = \{D_s\}_{s=1}^{|D|}$, which share a common feature space $\mathcal{F}$ and label space $\mathcal{Y}$. For scenario D_s, the labeled training data are drawn from a domain-specific distribution $\mathcal{P}_s$ over $\mathcal{F} \times \mathcal{Y}$. Although there may be many common users and items between two scenarios, the distribution $\mathcal{P}_s$ are quite distinguishing in different scenarios. The **Multi-Scenario Matching Problem** can be formulated as follows:

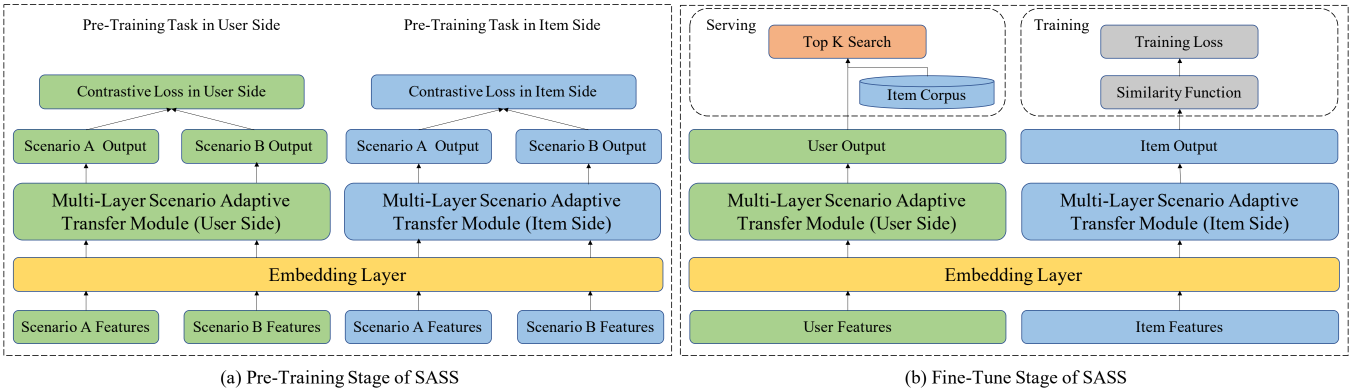
$$\widetilde{V_s} = max_{1 \leq k \leq K}(sim(e_u^s, e_v^s)), v \in V \tag{1}$$

where V denotes the large-scale candidate item set. e_u^s, e_v^s is the user and item representation vectors in scenario D_s, $sim(e_u^s, e_v^s)$ is the relevance score between user u and item v, and $\widetilde{V_s}$ is the final top K matching results for D_s.

The multi-scenario problem can be treated as a specific case of **multi-domain learning problem** with three important characteristics: (a) All scenarios have the same user/item type and enjoy a common feature schema; (b) All scenarios share the same learning objective (multiple inputs and one objective); (c) The key points of multi-scenario learning focus on the information sharing and transfer among scenarios, so that all scenarios' performances can be improved simultaneously. It should be emphasized that the multi-scenario problem is quite different from cross domain problem[12, 18, 26, 34] (mainly focusing on improving target domain's performance), multi-task learning problem[20, 24, 32] (one input and multiple objectives) and heterogeneous multi-domain task[11] with different item types.

4 PROPOSED METHOD

In this section, we describe the proposed Scenario-Adaptive and Self-Supervised Model (SASS). SASS has two stages, Pre-Training Stage and Fine-Tune Stage:


Pre-Training Task in User Side
Pre-Training Task in Item Side
Contrastive Loss in User Side
Contrastive Loss in Item Side
Scenario A Output
Scenario B Output
Multi-Layer Scenario Adaptive Transfer Module (User Side)
Multi-Layer Scenario Adaptive Transfer Module (Item Side)
Embedding Layer
Scenario A Features
Scenario B Features
Serving
Top K Search
Item Corpus
Training
Training Loss
Similarity Function
User Output
Item Output
User Features
Item Features

(a) Pre-Training Stage of SASS

(b) Fine-Tune Stage of SASS

Figure 3: Overall architecture of SASS model. (a) Pre-training stage of SASS. (b) Fine-tune stage of SASS

- **Pre-Training Stage**, as shown in Figure 3(a). There are user-side and item-side pre-training task in this stage. Both of the two tasks have the same model structure. The embedding layer is shared between the two tasks. The pre-training stage is based on a self-supervised contrastive learning strategy, which will be described in detail in subsection 4.1.
- **Fine-Tune Stage**, as shown in Figure 3(b). The model structure in fine-tune stage utilizes a dual double-tower framework to generate user and item representation vectors separately. The fine-tune and online serving operations will be illustrated in subsection 4.2 and 4.3. The embedding layer and network weights in fine-tune stage are restored from the pre-training stage, so that the model in fine-tune stage can reuse well-trained information from the entire sample space.

Both of the two stages utilize a **M**ulti-**L**ayer **S**cenario **A**daptive **T**ransfer Module (**ML-SAT**) to regulate the fine-grained and decoupled information transfer from whole scenarios to specific scenario. We minutely describe it in subsection 4.1.2.

4.1 Self-Supervised Framework Based on Contrastive Learning

We introduce the paradigm of contrastive learning for our pre-training task and creatively correlate the multi-scenario matching framework and contrastive learning framework in a unified way. Training samples in pre-training stage are drawn from both labeled(clicked) and unlabeled(exposed but not clicked) data space.

As shown in Figure 3(a), in the user side, behaviors of the same user in different scenario can be treated as a mechanism of data augmentation. After that, a Multi-Layer Scenario Adaptive Transfer Module (ML-SAT) is proposed to generate distinguishing representation vectors for the same user in different scenarios. Finally, the contrastive loss function[2] is introduced as the optimization loss to maximize agreements between different latent representations of same user in different scenarios. If a user u accessed k scenarios, we simply split these scenarios and make a combination of two to generate $C_k^2 = \frac{k*(k-1)}{2}$ samples for contrastive learning.

In the item side, exposures and interactions of the same item in different scenario can also be treated as a mechanism of data augmentation. Siamese model network and contrastive loss function will be adopted to the pre-training process for the same item. Similar operations of sample combination are conducted to generate multiple samples for item side, just as user side did.

4.1.1 Feature Composition and Embedding Layer. In the user side, every training sample contains user profiles, scenario context features and two groups of user-scenario cross behavior features which are drawn from two scenarios separately. Specifically, user profiles contain age, gender, etc. Scenario context features mainly include scenario ID. User-scenario cross behavior features including user behavior sequences, category preferences and user statistic characteristics in the corresponding scenario.

For user behavior sequence features, the list of feature field embeddings will be concatenated to form the item embedding for each item in the behavior sequence. The final user behavior sequence embeddings will be generated by sequence pooling strategies. We utilize self-attention mechanism[7] as our sequence pooling operation, other strategies[19, 29] can be investigated for better performance, which we remain for future research.

In the item side, item profiles contain item ID, item category ID, account ID, etc. item-scenario cross features mainly include item statistic characteristics in the corresponding scenario.

To highlight the significance of scenario context features, we introduce a separate auxiliary network to model scenario characteristic, as formulated below:

$$a = f(W_a x_a + b_a) \tag{2}$$

where x_a is the embedding of scenario context features, $f(*)$ is a multi-layer perceptron.

4.1.2 Multi-Layer Scenario Adaptive Transfer Module (ML-SAT). We propose a novel network to extract representation vectors in corresponding scenario both for users and items. The structures between user side and item side are nearly the same, as shown in Figure 3. Thus, we only describe the user side network in the following of the paper.

To make full use of whole-scenario information and regulate the fine-grained information transfer from whole scenarios to specific scenario, we introduce a *global shared network* to learn the information of all scenarios, and propose a **M**ulti-**L**ayer **S**cenario **A**daptive **T**ransfer Module(**ML-SAT**) as the *scenario specific network* for each individual scenario. As shown in Figure 4(a), the *global shared network* is a multi-layer perceptrons shared by all the

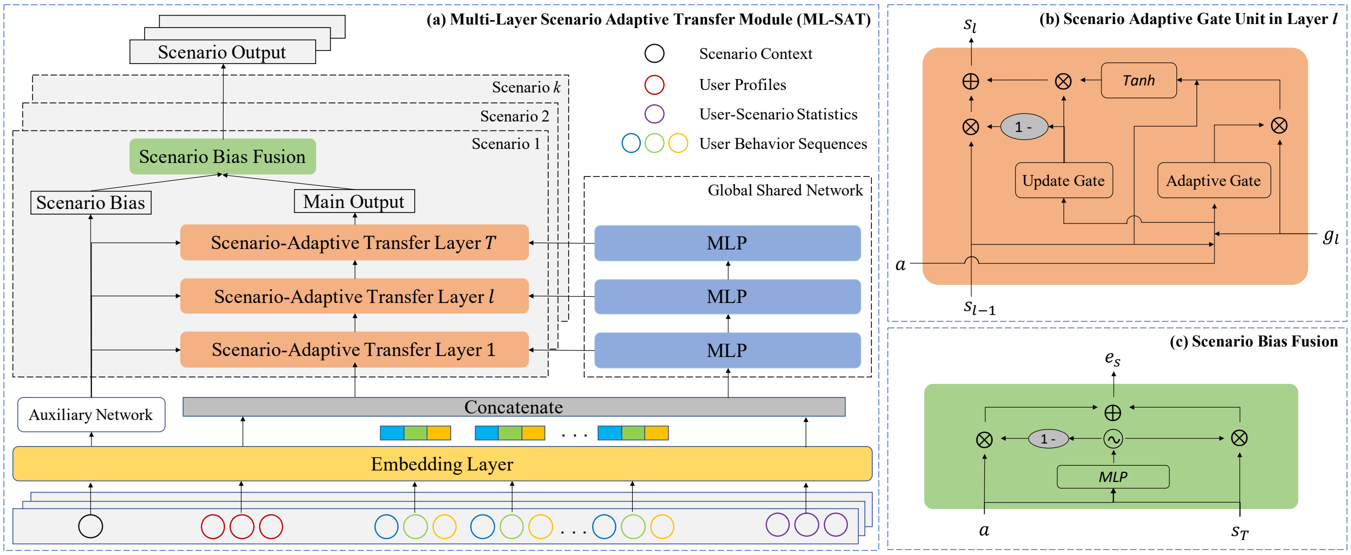


Figure 4: Overview of Multi-Layer Scenario Adaptive Transfer Module. We depict the model structure in user side. Item side's structure is identical with this. We describe the framework in one scenario in detail. Other scenarios will follow the same paradigm to generate corresponding vectors. (a) Multi-Layer Scenario Adaptive Transfer module. (b) Scenario-Adaptive Gate Unit in lth layer of ML-SAT. (c) Scenario bias fusion mechanism

scenarios. Training samples coming from the whole scenarios will be fed into *global shared network* to train a whole-scenario model. ML-SAT, which is parameter-specific in separate scenario, is trained with training samples coming from corresponding scenario.

Motivated by GRU[5], in each network layer, we design a *scenario-adaptive gate unit* with explicit gate mechanisms to regulate the fine-grained information transfer from whole scenarios to specific scenario. As shown in Figure 4(b), the *scenario-adaptive gate unit* can be formulated as follows:

$$r_l = \sigma(W_r^l[g_l, s_{l-1}] + W_{br}a) \tag{3}$$

$$h_l = tanh(W_h^l[r_l \cdot g_l, s_{l-1}]) \tag{4}$$

$$z_l = \sigma(W_z^l[g_l, s_{l-1}] + W_{bz}a) \tag{5}$$

$$s_l = (1 - z_l) \cdot s_{l-1} + z_l \cdot h_l \tag{6}$$

where g_l denote the lth layer output of *global shared network*. s_{l-1} denote the $(l-1)$th layer output of *scenario specific network*. W_z^l and W_{bz} are the projection matrix weights and bias matrix weights of *update gate* z_l in lth layer. W_r^l and W_{br} are the projection matrix weights and bias matrix weights of *adaptive gate* r_l in lth layer. a is the output of the scenario auxiliary network, which is introduced as a strong scenario indicator bias.

In Equation (3), *adaptive gate* r_l decides the degree of useful information the *global shared network* can transfer. h_l reflects the new adaptive state transfer information, considering the correlation between g_l and s_{l-1}, as depicted in Equation (4). With the *update gate* z_l as fusion weights, the new output s_l can be updated by the fusion of s_{l-1} and adaptive transfer output h_l. The network layer with *scenario-adaptive gate unit* can be stacked to multiple layers for more progressive layered extraction[32].

4.1.3 *Scenario Bias Fusion.*

After the multi-layer network with *scenario-adaptive gate unit*, we can get the scenario main output of the corresponding scenario. Considering the significant importance of scenario context features in distinguishing scenario characteristics, we treat the output of scenario auxiliary network as scenario bias and fuse it with scenario main output to generate the final scenario-specific representation vector, as shown in Figure 4(c).

$$e_s = \alpha \cdot s_T + (1 - \alpha) \cdot a \tag{7}$$

$$\alpha = \sigma(W_o[s_T, a]) \tag{8}$$

where e_s is the final output, s_T is the scenario main output, with T as the final layer number. a is the output of the scenario auxiliary network. σ is the sigmoid function.

4.1.4 *Self-supervised Optimization Objective.*

For each training sample with two groups of scenario-specific features in corresponding scenario, we can get two output vectors for the same user or item separately, denoted as e_s^i and e_s^j. The objective of pre-training task is to extract agreements and model corrections between different scenarios. Thus, we adopt the same self-supervised contrastive loss of [2] as our optimizing loss. Specifically, for a minibatch training samples with batch size N, we can get $2N$ vectors after ML-SAT. we treat e_s^i and e_s^j as a positive pair. The other $2(N-1)$ scenario vectors are regarded as negative vectors. The loss function for a positive pair (e_s^i, e_s^j) is defined as

$$\mathcal{L}_{ij} = -log \frac{exp(sim(e_s^i, e_s^j)/\tau)}{\sum_{k=1, k\neq i}^{2N} exp(sim(e_s^i, e_s^k)/\tau)} \tag{9}$$

where $sim(e_s^i, e_s^j) = \frac{(e_s^i)^T e_s^j}{\|e_s^i\| \|e_s^j\|}$. τ is the temperature parameter. The final loss is the sum of all losses in the minibatch, as denoted in equation (10).

$$\mathcal{L} = \sum_{k=1}^{N} (\mathcal{L}_{ij})^{(k)} \tag{10}$$

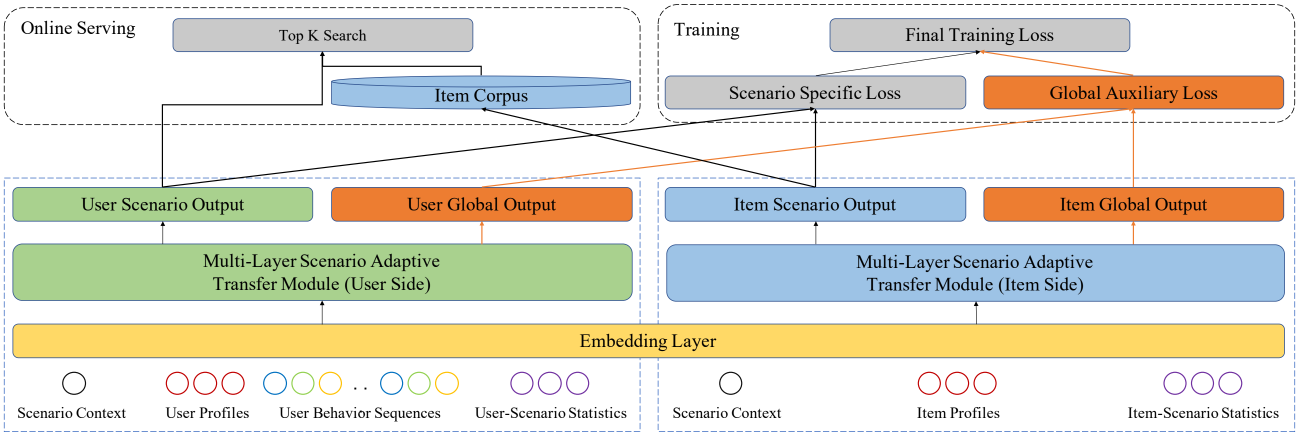


Figure 5: Fine-tune stage of SASS. Every user and item will output two vectors respectively, with the scenario specific loss and global auxiliary loss as optimization objectives. For online serving, we only employ scenario output of users and items.

4.2 Fine-Tune Stage

Most model components in fine-tune stage are identical with pre-training stage, such as ML-SAT module and Scenario Bias Fusion module. For brevity, we only describe components different from pre-training stage in the following subsections.

4.2.1 Feature Composition and Embedding Layer. The training samples in fine-tune stage are labeled data drawn from target scenario. A user can only access one scenario at a time, so the features in one training sample only contain features in a single scenario. User side features contain user profiles, scenario context feature (scenario ID), user's behavior sequences, statistic features and prefer features of user in target scenario. Item side features contain item profiles, scenario context feature (scenario ID), statistic features and prefer features of item in target scenario. Then, as shown in Figure 5, user side features and item side features are separately fed into embedding layer and upper ML-SAT modules to generate final scenario-specific vectors in corresponding scenario both for users and items. It should be emphasized that the embedding layer and networks weights of ML-SAT in fine-tune stage are restored from pre-training stage.

4.2.2 Fine-Tune Optimization Objective. The final optimization loss in fine-tune stage is the combination of *Scenario Specific Loss* and *Global Auxiliary Loss*.

Scenario Specific Loss Function: Similar with other matching tasks[13, 25, 37], we adapt pairwise loss to optimize our fine-tune matching task. For a scenario s, the kth training sample in fine-tune stage is a triplet (u_s^k, p_s^k, n_s^k). u_s^k denotes user representation vector, p_s^k and n_s^k denote corresponding positive and negative item vector, respectively. Negative items are randomly sampled from whole candidate set with negative sampling strategy[23]. The scenario-specific loss function is defined as

$$\mathcal{L}_{scenario} = \frac{1}{N}\sum_{k=1}^{N} log(1 + \sigma(sim(u_s^k, n_s^k) - sim(u_s^k, p_s^k))) \quad (11)$$

where σ is sigmoid function and $sim(*)$ is cosine function.

Global Auxiliary Loss Function: In fine-tune stage, the *global shared network* will also output a representation vector g_T for each sample, as shown in Figure 5. The *global shared network* is trained with samples from all scenarios. So the output g_T can be treated as the user or item representations in a global perspective. Modeling the similarity of users and items in the global scope is beneficial to the convergence of training and performance improvement. So we introduce a global auxiliary loss defined as:

$$\mathcal{L}_{auxiliary} = \frac{1}{N}\sum_{k=1}^{N} log(1 + \sigma(sim(u_g^k, n_g^k) - sim(u_g^k, p_g^k))) \quad (12)$$

where u_g^k, p_g^k, n_g^k denote the representation vectors of a triplet (user, positive item, negative item) respectively, which are all generated though *global shared network*. Finally, the loss function in fine-tune stage is formulated as below with a hyper-parameter β:

$$\mathcal{L} = \mathcal{L}_{scenario} + \beta \cdot \mathcal{L}_{auxiliary} \quad (13)$$

4.3 Online Serving

When SASS is trained, the whole model in fine-tune stage can be deployed for online serving. For a specific scenario s, all the items with their features will be fed into the model and generate item vectors e_s from corresponding *scenario specific network* in the item side architecture. Then, all the item vectors are saved as an item corpus. During online serving, when a user accesses scenario s, the user features are fed into SASS and the user vectors u_s is generated from corresponding *scenario specific network* in the user side architecture. Finally, an online real-time top-k retrieval operation based on approximate near neighbor algorithms[15] is implemented. These retrieved results are treated as candidates for subsequent ranking tasks.

5 EXPERIMENTS

To adequately evaluate the proposed SASS model, we conduct experiments to answer the following research questions:

- How about SASS model compared with state-of-art matching models trained with one scenario data for each scenario or trained with whole-scenario data for all scenarios?
- How about SASS model compared with other state-of-art multi-scenario matching models?
- How about the impact of each part on the overall model?

5.1 Experimental Settings

5.1.1 Datasets. We conduct experiments on our industrial dataset and two public datasets. Table 1 summarizes the basic information of these datasets.

- **Ali-MSSV**. Our industrial Muiti-Scenario Short Video (MSSV) dataset in Taobao. Data from 2022-03-24 to 2022-04-04 is utilized for training and 2022-04-05 for testing. We evaluate models on two dense scenarios (denoted as #A1 and #A2) with abundant user behaviors and two spare scenarios (#A3 and #A4) with sparse data.
- **Ali-CCP**[22]. A public dataset released by Taobao with prepared training and testing set. We split the dataset into 3 scenarios according to *scenario id*, denoted as #B1 to #B3 for simplicity.
- **Ali-Mama**[8]. A public dataset released by Alimama, an online advertising platform in China. Data from 2017-05-06 to 2017-05-011 is utilized for training and 2022-05-11 for testing. We arrange the dataset into 5 scenarios according to the *city level*, denoted as #C1 to #C5 for simplicity.

5.1.2 Competitors. We release two types of SASS models to compare with other methods.

- **SASS-Base**: It is the model trained with labeled data without pre-training. We will evaluate the performance of SASS-Base on all of the three datasets.
- **SASS**: It is our proposed model with two stages. The Ali-CCP dataset is unsuitable for pre-training task due to the lack of indispensable features, so we only evaluate the performance of SASS on our Ali-MSSV dataset and Ali-Mama dataset.

The compared *Single-Scenario* matching models (trained with single-scenario data) are listed as follows:

- **YoutubeDNN**: YoutubeDNN[6] adopts average pooling to extract user's interest with a sampled softmax loss to optimize similarities between users and items.
- **DSSM**: DSSM[14] builds a relevance score model to extract user and item representations with double-tower architecture.
- **BST**: BST[1] leverages transformer to build the user behavior sequence. In this paper we use the inner product of user and item representations instead of MLP.
- **MIND**: MIND[16] clusters users' multiple interests by capsule network to improve the effect of multi-interest promotion.

The model ment above will also be trained with all-scenario data for *Mix-Scenario* versions.

The multi-scenario matching models are listed as follows. To the best of our knowledge, most of the existing multi-scenario models are mainly proposed for ranking problems, Therefore, for original multi-scenario ranking modes, some necessary but slight modifications are made for unified evaluation on matching tasks, denoted by a postfix **-M** for each model.

- **SAR-Net-M**: Modified version of SAR-Net[29] which propose a multi-scenario architecture for scenario information migration with scenario-specific users' behaviors and attention mechanism;
- **STAR-M**: Modified version of STAR[30] which constructs star topology for multi scenarios and operate an element-wise operation to control information transferred from central network to specific networks;

Table 1: Basic information of three datasets

	Scenario	User	item	Samples
Ali-MSSV (#A)	#A1	10.1M	2.5M	630M
	#A2	30.1M	5.6M	1.2B
	#A3	2.1M	0.53M	100.2M
	#A4	1.5M	0.39M	100M
Ali-CCP (#B)	#B1	0.13M	1.98M	0.63M
	#B2	0.18M	2.44M	1M
	#B3	50k	0.2M	13k
Ali-Mama (#C)	#C1	40K	50k	94k
	#C2	0.15M	0.13M	0.42M
	#C3	80K	89k	0.23M
	#C4	65K	77k	0.19M
	#C5	0.14M	0.12M	0.35M

- **HMoE-M**: Modified version of HMoE[17] which constructs relationship between multiple scenarios in label implicit space through stacked model;
- **ZEUS-M**: Modified version of ZEUS[9] which learns unlabeled data through a self-supervised *Next-Query Prediction* task;
- **ICAN**: ICAN[35] is one of the SOTA models for multi-domain matching, which is most related works of our task. It highlights the interactions between feature fields in different domains for cold-start matching.

5.1.3 Parameter Settings and Metrics. For all methods, the truncation length of user behavior is 50. AdamGrad is used as the optimizer with learning rate of 0.001 for all methods and the batch size is 512. Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) are adopted as the performance metrics. We set HR@20 and NDCG@20 in top-20 matching results as final metrics.

5.2 Overall Experimental Results: RQ1 & RQ2

We evaluate our methods with other compared models, As illustrated in Table 2. We can summarize three significant observations: (1) The experiment performances of mix-scenario models are generally better than single-scenario models in sparse scenarios (such as scenario #A4, #B1). We suspect that it is challenging for sparse scenarios to train perfect models with limited training samples. Additional samples coming from other scenarios can partly improve the performance of sparse scenarios. However, in dense scenarios (such as #A1, #C2), mix-scenario models perform worse than single-scenario models. One possible reason may be that the crude mixture of multi-scenario samples introduces non-negligible noise data, which is harmful to the model performances. These two observations are consistent with the conclusions in SAR-Net[29]. (2) Unified multi-scenario modes all get better performances than single-scenario models and mix-scenario models. The results indicate that the strategies with information share and modeling of scenario commonalities and distinctions are beneficial to solving multi-scenario problems. (3) SASS-Base (without pre-training) outperforms other compared models (single-scenario, mix-scenario and multi-scenario) in nearly all scenarios (with comparable performance with SAR-Net-M and STAR-M in scenario #A2). It shows that SASS-Base has a leading performance over other multi-scenario models in building commonalities and characteristics of different

Table 2: Performance of different models on three datasets. Single-scenario models are trained with data in individual scenario independently while mix-scenario models are trained with data from all scenarios. Multi-Scenario models are trained with multi-scenario data in unified frameworks. Due to the lack of indispensable features for pre-traing task on Ali-CCP dataset, we only evaluate the performance of SASS on Ali-MSSV and Ali-Mama datesets.

		single-scenario models				mix-scenario models				multi-scenario models						
		MIND	DSSM	BST	YoutubeDNN	MIND	DSSM	BST	YoutubeDNN	SAR-Net-M	STAR-M	HMoE-M	ZEUS-M	ICAN	SASS-Base	**SASS**
#A1	HR@20	0.052	0.039	0.042	0.034	0.044	0.032	0.035	0.027	0.079	0.082	0.061	0.072	0.072	0.082	**0.087**
	NDCG@20	0.021	0.012	0.014	0.015	0.024	0.010	0.012	0.009	0.033	0.036	0.022	0.031	0.025	0.039	**0.042**
#A2	HR@20	0.032	0.016	0.029	0.023	0.027	0.012	0.023	0.016	0.035	0.034	0.027	0.032	0.031	0.034	**0.043**
	NDCG@20	0.014	0.009	0.013	0.012	0.011	0.007	0.008	0.004	0.014	0.012	0.011	0.014	0.016	0.019	**0.029**
#A3	HR@20	0.041	0.032	0.036	0.029	0.045	0.033	0.041	0.028	0.042	0.045	0.041	0.039	0.039	0.047	**0.068**
	NDCG@20	0.023	0.017	0.021	0.015	0.021	0.017	0.019	0.013	0.021	0.019	0.017	0.016	0.013	0.025	**0.037**
#A4	HR@20	0.017	0.010	0.012	0.006	0.023	0.011	0.012	0.009	0.027	0.021	0.029	0.031	0.025	0.032	**0.047**
	NDCG@20	0.007	0.003	0.007	0.002	0.011	0.007	0.004	0.005	0.012	0.013	0.010	0.012	0.010	0.017	**0.023**
#B1	HR@20	0.153	0.112	0.144	0.120	0.164	0.133	0.156	0.137	0.185	0.211	0.194	0.212	0.175	**0.234**	-
	NDCG@20	0.061	0.043	0.057	0.049	0.091	0.052	0.069	0.062	0.082	0.102	0.097	0.117	0.083	**0.137**	-
#B2	HR@20	0.191	0.143	0.167	0.136	0.212	0.168	0.193	0.151	0.241	0.227	0.219	0.231	0.199	**0.252**	-
	NDCG@20	0.121	0.093	0.103	0.092	0.137	0.112	0.117	0.096	0.117	0.106	0.094	0.123	0.104	**0.131**	-
#B3	HR@20	0.043	0.029	0.037	0.021	0.067	0.042	0.058	0.039	0.074	0.069	0.081	0.079	0.069	**0.092**	-
	NDCG@20	0.019	0.013	0.013	0.011	0.029	0.019	0.028	0.021	0.032	0.041	0.043	0.037	0.033	**0.051**	-
#C1	HR@20	0.213	0.176	0.192	0.172	0.232	0.179	0.193	0.132	0.239	0.242	0.219	0.221	0.203	0.243	**0.269**
	NDCG@20	0.117	0.098	0.114	0.079	0.125	0.093	0.106	0.056	0.119	0.107	0.129	0.114	0.105	0.132	**0.137**
#C2	HR@20	0.179	0.127	0.142	0.114	0.155	0.107	0.123	0.081	0.147	0.173	0.183	0.192	0.204	0.209	**0.227**
	NDCG@20	0.083	0.054	0.078	0.042	0.069	0.042	0.067	0.033	0.084	0.088	0.092	0.097	0.099	0.107	**0.119**
#C3	HR@20	0.256	0.237	0.217	0.193	0.251	0.241	0.224	0.201	0.241	0.269	0.243	0.261	0.271	0.274	**0.289**
	NDCG@20	0.131	0.119	0.095	0.088	0.127	0.119	0.107	0.081	0.132	0.139	0.137	0.122	0.138	0.142	**0.151**
#C4	HR@20	0.225	0.193	0.201	0.165	0.231	0.201	0.215	0.177	0.239	0.241	0.251	0.247	0.221	0.259	**0.267**
	NDCG@20	0.131	0.099	0.123	0.077	0.127	0.113	0.093	0.081	0.132	0.114	0.132	0.129	0.142	0.134	**0.142**
#C5	HR@20	0.191	0.167	0.188	0.143	0.185	0.155	0.147	0.132	0.204	0.212	0.209	0.189	0.217	0.227	**0.241**
	NDCG@20	0.082	0.077	0.112	0.059	0.104	0.082	0.069	0.055	0.093	0.112	0.103	0.104	0.121	0.132	**0.137**

Table 3: Ablation study of Scenario-Adaptive Gate Unit.

	#A1		#A3	
	HR@20	NDCG@20	HR@20	NDCG@20
SASS*	0.054	0.021	0.029	0.019
SASS*+Production Gate	0.059	0.022	0.033	0.018
SASS*+Simnet Gate	0.065	0.027	0.037	0.017
SASS*+Sigmoid Gate	0.071	0.029	0.042	0.023
SASS-Base	**0.082**	**0.039**	**0.047**	**0.025**

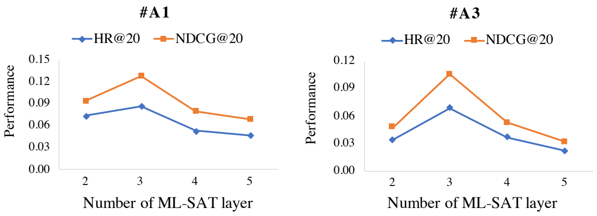


Figure 6: Ablation study of different layer number.

scenarios. Moreover, SASS (with pre-traing) can further improve the performance, especially for sparse scenarios. The performance gain of SASS in scenarios #A3 is much higher than that in other scenarios, indicating the excellent improvements of pre-training task in sparse scenarios.

5.3 Ablation Study: RQ3

5.3.1 Scenario-Adaptive Gate Unit. Scenario-Adaptive Gate Unit controls the information transfer from whole scenarios to specific scenario, which is essential for modeling scenario commonalities and distinctions. In this subsection, we investigate different transfer gate mechanisms and compare their performance on the SASS-Base model (SASS without pre-training stage). The baseline model is SASS*, which is a SASS-Base version without transfer gate. SASS* can be treated as a variant framework of MOE[28]. In particular, we compare the following settings: (1) Sigmoid Gate: concatenating scenario specific layer's output s_l with global shared layer's output g_l and feeding the results into MLP with sigmoid to control information transfer. SASS*+Sigmoid Gate can be treated as a variant framework of MMoE[20] or Cross-Stitch[24], which is also the fundamental transfer and fusion structure in SAR-Net[29] and HMoE[17]. (2) Production Gate: taking the element-wise production of scenario specific layer's output s_l and global shared layer's output g_l to generate the final scenario output, which can be partly treated as an equivalent of information mapping of STAR[30]. (3) Simnet Gate: an updated version for the fusion of information with the concatenations of element-wise production, element-wise subtraction and element-wise addition as the input. Then the input is feed into MLP with sigmoid. It can be summarized from Table 3 that (1) SASS* performs worst and SASS* with other gate mechanisms are beneficial to scenario-specific modeling both in #A1 and #A3. (2) The proposed SASS-Base model achieves the best performance. It shows that the fine-grained and decoupled gate mechanism in ML-SAT can get better control on information transfer from whole scenarios to individual scenario.

5.3.2 Different layer number of ML-SAT. Motivated by PLE[32], the Scenario-Adaptive Transfer Layer can be stacked multiple layers for

Table 4: Ablation study of different pre-training strategies. *Next Video Prediction* is the strategy introduced by ZEUS

	#A1		#A3	
	HR@20	NDCG@20	HR@20	NDCG@20
SASS-Base	0.082	0.039	0.047	0.025
Next Video Prediction	0.083	0.033	0.052	0.031
SASS	**0.087**	**0.042**	**0.068**	**0.037**

Table 5: Ablation study of different item representation strategies

	#A1		#A3	
	HR@20	NDCG@20	HR@20	NDCG@20
Single Item Embedding	0.081	0.037	0.054	0.033
SASS	**0.087**	**0.042**	**0.068**	**0.037**

better performances. Thus, we set SASS-Base model with different layers for comparison. The results can be shown in Figure 6. The performances become better when the number of layers are stacked from 2 to 3, while getting worse with the further increase of layer number. We suspect that, with the layer number increasing, the distinctions of scenario-specific representation are decreasing if much more information is transferred from whole scenarios to specific scenario.

5.3.3 Self-supervised Learning based Pre-training. In this part, we investigate the performances of different self-supervised strategies in pre-training stage. We set SASS-Base (model without pre-training) as baseline model. We compare SASS-Base with SASS (with pre-training task) and another self-supervised *Next Video Prediction* task, which is introduced in ZEUS[9]. The results shown in Table 4 exhibit the superior performance of SASS, especially in spare scenarios (such as #A3).

5.3.4 Representations for items in different scenarios. To certify the performance of generating distinguishing item representations in different scenarios, we implement a variant of SASS with only one single item representation vector shared in multiple scenarios. Results in Table 5 indicate that the performances gets better when generating different item representations in multiple scenarios. The reason is that when we consider the different representations of items in various scenarios, the characteristics of individual scenario can be more directly captured.

5.3.5 Scenario Bias Fusion & Global Auxiliary Loss Task. Scenario Bias Fusion is proposed to highlight the importance of scenario context information, while the global auxiliary loss task is introduced to strengthen the learning of global shared network. Both of the two strategies are expected to improve the whole performances of SASS. Experimental results in Table 6 demonstrate our hypothesises.

Table 6: Ablation study of scenario bias fusion and global auxiliary loss task

	#A1		#A3	
	HR@20	NDCG@20	HR@20	NDCG@20
No Global Auxiliary Task	0.045	0.018	0.036	0.021
No Scenario Bias Fusion	0.075	0.031	0.039	0.024
SASS	**0.087**	**0.042**	**0.068**	**0.037**

Table 7: AWT and CTR gains in online short video recommendation platform of Taobao, Alibaba

Scenarios	#A1		#A2		#A3		#A4	
Metrics	AWT	CTR	AWT	CTR	AWT	CTR	AWT	CTR
Gains	+8.3%	+16.3%	+4.5%	+5.3%	+3.1%	+1.2%	+15.2%	+16.2%

5.4 Online Deployment Test

Since August 2021, we have conducted online A/B tests and successfully developed SASS model on Taobao in Alibaba, which contains multiple short video recommendation scenarios. We collect the overall improvements of A/B tests in each industrial scenarios, where the base model is the double-tower matching model[13]. The online evaluation metric is AWT (average watching time per user) and CTR (the number of clicks over the number of video impressions). As shown in Table 7, the online results have demonstrated the feasibilities and effectiveness of our proposed SASS model on real industrial recommendation systems.

6 CONCLUSION

In this paper, we propose the Scenario-Adaptive and Self-Supervised (SASS) model to tackle three core problems of multi-scenario modeling mentioned above. To model multi-scenario commonalities and distinctions, SASS build a global shared network for all scenarios and a Multi-Layer Scenario Adaptive Transfer Module (ML-SAT) as scenario-specific network for each scenario. In ML-SAT, the Scenario-Adaptive Gate Unit is introduced to select and control information transfer from global shared network to scenario-specific network in a much fine-grained and decoupled way. To sufficiently exploit the power of entire space samples, a two-stage training framework including pre-training and fine-tune is introduced. The pre-training stage is based on a scenario-supervised contrastive learning task with the training samples drawn from labeled and unlabeled data spaces. Moreover, the model architecture of SASS is created symmetrically both in user side and item side, so that we can get distinguishing representations for items in individual scenarios. The experimental results on both offline datasets (industrial and public) and online A/B tests demonstrate the superiority of SASS over state-of-the-art methods for solving multi-scenario problems. SASS has been deployed on the online short video recommendation platform in Taobao, bringing more than 8% improvement on AWT.

REFERENCES

[1] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. 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*. 1–4.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[3] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15750–15758.

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

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation.

arXiv preprint arXiv:1406.1078 (2014).

[6] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. 293–296.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. 2017. Learning piece-wise linear models from large scale data for ad click prediction. *arXiv preprint arXiv:1704.05194* (2017).

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

[10] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964* (2020).

[11] Xiaobo Hao, Yudan Liu, Ruobing Xie, Kaikai Ge, Linyao Tang, Xu Zhang, and Leyu Lin. 2021. Adversarial Feature Translation for Multi-domain Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2964–2973.

[12] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 667–676.

[13] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[14] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[15] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[16] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.

[17] Pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang. 2020. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2605–2612.

[18] Pan Li and Alexander Tuzhilin. 2020. Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 331–339.

[19] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2635–2643.

[20] 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.

[21] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.

[22] 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.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[24] 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.

[25] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2876–2885.

[26] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. Minet: Mixed interest network for cross-domain click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2669–2676.

[27] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2536–2544.

[28] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).

[29] 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 & Knowledge Management*. 4094–4103.

[30] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4104–4113.

[31] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[32] 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.

[33] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.

[34] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. 2021. Contrastive Cross-domain Recommendation in Matching. *arXiv preprint arXiv:2112.00999* (2021).

[35] Ruobing Xie, Zhijie Qiu, Jun Rao, Yi Liu, Bo Zhang, and Leyu Lin. 2020. Internal and Contextual Attention Network for Cold-start Multi-channel Matching in Recommendation.. In *IJCAI*. 2732–2738.

[36] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised Learning for Large-scale Item Recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4321–4330.

[37] Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wen-Yun Yang. 2020. Towards personalized and semantic retrieval: An end-to-end solution for E-commerce search via embedding learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2407–2416.

[38] Qianqian Zhang, Xinru Liao, Quan Liu, Jian Xu, and Bo Zheng. 2022. Leaving No One Behind: A Multi-Scenario Multi-Task Meta Learning Approach for Advertiser Modeling. *arXiv preprint arXiv:2201.06814* (2022).

[39] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2021. Contrastive learning for debiased candidate generation in large-scale recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3985–3995.

[40] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1893–1902.