



IncMSR: An Incremental Learning Approach for Multi-Scenario Recommendation

Kexin Zhang*
zkx21@mails.tsinghua.edu.cn
Tsinghua Shenzhen International
Graduate School, Tsinghua University
Shenzhen, China

Yichao Wang*
wangyichao5@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Xiu Li†
li.xiu@sz.tsinghua.edu.cn
Tsinghua Shenzhen International
Graduate School, Tsinghua University
Shenzhen, China

Ruiming Tang†
tangruiming@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Rui Zhang
rayteam@yeah.net
ruizhang.info
Shenzhen, China

ABSTRACT

For better performance and less resource consumption, multi-scenario recommendation (MSR) is proposed to train a unified model to serve all scenarios by leveraging data from multiple scenarios. Current works in MSR focus on designing effective networks for better information transfer among different scenarios. However, they omit two important issues when applying MSR models in industrial situations. The first is the efficiency problem brought by mixed data, which delays the update of models and further leads to performance degradation. The second is that MSR models are insensitive to the changes of distribution over time, resulting in suboptimal effectiveness in the incoming data. In this paper, we propose an incremental learning approach for MSR (IncMSR), which can not only improve the training efficiency but also perceive changes in distribution over time. Specifically, we first quantify the pair-wise distance between representations from scenario, time and time-scenario dimensions respectively. Then, we decompose the MSR model into scenario-shared and scenario-specific parts and apply fine-grained constraints on the distances quantified with respect to the two different parts. Finally, all constraints are fused in an elegant way using a metric learning framework as a supplementary penalty term to the original MSR loss function. Offline experiments on two real-world datasets are conducted to demonstrate the superiority and compatibility of our proposed approach.

CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies → Transfer learning.

KEYWORDS

Incremental Learning, Multi-Scenario Recommendation, Metric Learning

*Both authors contributed equally to this research.

†Corresponding author



This work is licensed under a Creative Commons Attribution International 4.0 License.

WSDM '24, March 4–8, 2024, Merida, Mexico
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0371-3/24/03.
<https://doi.org/10.1145/3616855.3635828>

ACM Reference Format:

Kexin Zhang, Yichao Wang, Xiu Li, Ruiming Tang, and Rui Zhang. 2024. IncMSR: An Incremental Learning Approach for Multi-Scenario Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24)*, March 4–8, 2024, Merida, Mexico. ACM, Mérida, Yucatán, 10 pages. <https://doi.org/10.1145/3616855.3635828>

1 INTRODUCTION

Large-scale commercial platforms typically contain multiple scenarios to meet the diverse user groups. For example, a video-sharing platform may have dozens of scenarios including homepage, search discovery, guess what you like and etc. Within each scenario, users explore content driven by their individual motivations, exhibiting diverse user behaviors that reflect their interests. Conventional recommendation systems (RS) collect these behaviors from different scenarios and apply *Single-Scenario Recommendation (SSR)* which builds individual models for each scenario with data only from the corresponding scenario. This approach is inefficient due to the large number of models to maintain and is prone to sparsity issues when data is scarce in certain scenarios. To address these issues, *Multi-Scenario Recommendation (MSR)* [40] has been proposed. MSR builds a unified model that leverages data from multiple scenarios, improving efficiency, mitigating sparsity issues, and utilizing transferable information from various scenarios, thereby enhancing the effectiveness of multiple scenarios simultaneously. Consequently, MSR has gained wide adoption in real-world commercial RS.

Existing works in MSR [3, 25, 28, 33, 37] focus on better information transfer among different scenarios in the *Batch* mode, where the model is trained on a fixed-size window of training data. However, this approach faces two problems. Firstly, utilizing all data from multiple scenarios can construct comprehensive user profiles but degrades efficiency, hindering capturing the latest user preferences [34]. Secondly, these MSR models can not generalize well on the incoming data, because the data volumes and distributions of different scenarios change over time [18].

Incremental Learning [8, 20, 21, 36] has been proposed as a potential solution to the aforementioned issues encountered in SSR, updating models using only incoming data and based on the previous model. However, existing incremental learning methods in SSR can hardly be applied to MSR due to their inability to capture distribution changes across different scenarios. Figure 1 illustrates

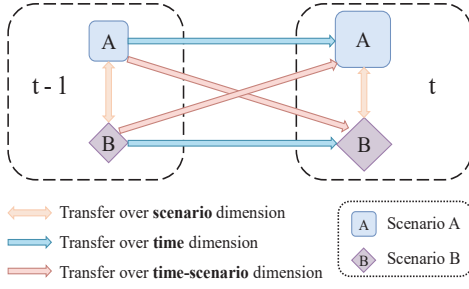


Figure 1: Illustration of different transfers over scenario, time and time-scenario dimensions.

the information transfer of MSR models in the *Incremental* mode. Specifically, when the MSR models update from step $t - 1$ to step t , there are three dimensions of information transfer. 1) Transfer between different scenarios at the same time step (scenario dimension). Different from batch mode, the impact of data imbalance between different scenarios is more serious when updating models with only incoming data. 2) Transfer between different time steps of the same scenario (time dimension). As the data comes incrementally, the distribution of the same scenario may change drastically. 3) Transfer between different scenarios at different time steps (time-scenario dimension). The incoming data of one scenario can degrade the existing shared parameters, which had been updated by the historical data from other scenarios. The coexistence of these three dimensions of information transfer makes it challenging to apply existing incremental algorithms to MSR models. Although some work addresses class-incremental transfer issues in computer vision [6, 13, 35], the research on achieving effective MSR in incremental mode remains limited.

In this paper, we propose IncMSR, a practical incremental learning approach for multi-scenario recommendation, which resolves the transferability issues over three dimensions in a unified way. We first quantify the pair-wise distance between representations in the scenario, time and time-scenario dimensions. Then, we decompose the MSR model into scenario-shared and scenario-specific parts, considering that the scenario-shared representations should contain relatively general and stable information for all scenarios while the scenario-specific representations should be more sensitive to the changes of scenario and time. Then we apply fine-grained constraints on the distances derived from the representations with respect to these two parts. Finally, all constraints are elegantly fused based on metric learning framework and as a supplementary penalty term to the original MSR loss function. The overall operations are conducted in an end-to-end manner.

The main contributions of this work are summarized as follows:

- To the best of our knowledge, it is the first work to achieve the multi-scenario recommendation in incremental mode.
- We analyze the specific issues and challenges in the incremental multi-scenario recommendation and propose IncMSR, a practical approach that quantifies distances between representations from three dimensions and applies fine-grained constraints on the distances quantified over two decomposed parts. Finally, we optimize the incremental MSR model in an end-to-end manner.

- The proposed IncMSR is model-agnostic and lightweight, which provides a supplementary penalty term to the original MSR loss function without increasing the number of parameters or model complexity.
- Experiments over two public datasets demonstrate the effectiveness and efficiency of our proposed approach.

2 RELATED WORK

2.1 Incremental Learning

Incremental learning (IL) has been proposed in SSR to improve the efficiency of model updates and to make the model adapt more quickly to the latest distribution. Existing IL methods can be broadly classified into three categories. Replay-based methods [9, 22] store historical data to maintain the memory. Regularization-based methods [8] introduce regularization terms to preserve prior knowledge. Model-based methods [20, 21] extract knowledge from historical models. SPMF [32] maintains historical data and combines it with new observations to update the model and make recommendations. IncCTR [34] consists of three decoupled modules to construct training data, handle features and fine-tune the model parameters. SML [36] transforms the old model to a new model via a neural network-based transfer learning component during training. However, the aforementioned works do not take into account the problem of multi-scenario recommendation, while our proposed IncMSR incorporates designs for transferring in multiple dimensions.

2.2 Multi-Scenario Recommendation

Multi-scenario recommendation optimizes all scenarios jointly by transferring information among them. It can be regarded as a special case of multi-task learning, where each scenario is viewed as a task. HMoE [15] utilizes Multi-gate Mixture-of-Experts [19] to model distinctions and commonalities among multiple scenarios in the latent space implicitly. SAR-Net [24] proposes a unified multi-scenario architecture that facilitates transfer across the scenario dimension by incorporating scenario-specific user behaviors with attention modules. STAR [25] adopts a star topology framework consisting of one central network that maintains scenario-shared commonalities and a set of scenario-specific networks to distinguish scenario distinctions. However, existing models suffer from the efficiency problem caused by the mixed data, resulting in delayed model updates and performance degradation. What's more, MSR models are insensitive to distribution changes over time.

2.3 Metric Learning

Metric learning aims to learn a suitable metric function to measure and compare the similarity and dissimilarity of feature distributions [26, 38]. CML [10] minimizes the Euclidean distance between users and items for fine-grained user preference. LRML [29] utilizes a memory-based attentive network explicitly to induce latent relations. DML [16] exploits bidirectional latent relations between users and items to transfer information across scenarios. Due to the capacity to capture important relationships among data, metric learning can be applied to quantify the distance between different distributions. IncMSR designs constraints from multiple dimensions for both scenario-shared and scenario-specific representations and leverages metric learning to enhance information transfer.

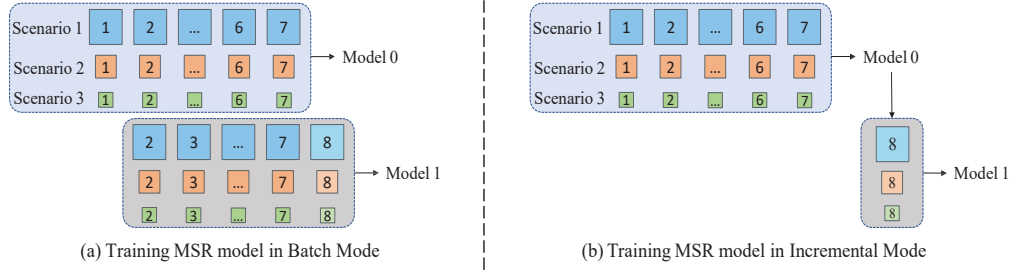


Figure 2: Comparison between Batch Mode (a) and Incremental Mode (b) for training MSR models: Batch Mode trains Model 1 from scratch with fixed window-size data, while Incremental Mode trains Model 1 using only incoming data based on Model 0.

3 PRELIMINARY

Existing MSR models typically learn a unified model in batch mode, which utilizes a fixed-size window of training data for iterative model training (Figure 2(a)). However, batch mode suffers from long training time due to the large data volume within each window. In contrast, incremental mode trains the model using only incoming data and the previous model, resulting in more efficient updates. As shown in Figure 2(b), "Model 0" as the initialized model, is trained using data of scenarios 1-3 from day 1 to day 7. Then, at the first incremental step, "Model 1" is trained based on the previous model "Model 0" and only data from day 8 is used.

In the context of IncMSR, we take (t, x, y, d) as input, where x is the common feature used by multiple recommendation scenarios like user historical behavior, user profile feature, item feature and context feature. As we focus on the click-through rate (CTR) task, $y \in \{0, 1\}$ is the label for each interaction indicating click or not. $d \in \{1, 2, \dots, N\}$ is the scenario indicator that indicates the scenario of the corresponding interaction. $D_{t,d}$ represents data collected from scenario d at incremental step t .

MSR models can be categorized into two main structures: hard parameter sharing and soft parameter sharing. Given input common features $x_t \in D_t$ from multiple scenarios, we can obtain scenario-shared representations $z_{t,d}^a$ and scenario-specific representations $z_{t,d}^s$ for the d -th scenario from scenario-shared layers and scenario-specific layers, respectively.

$$\begin{aligned} z_{t,d}^a &= x_{t,d} W_{t,d}^a + b_{t,d}^a \\ z_{t,d}^s &= z_{t,d}^a W_{t,d}^s + b_{t,d}^s \end{aligned} \quad (1)$$

where $W_{t,d}^a, b_{t,d}^a, W_{t,d}^s, b_{t,d}^s$ denote learnable weights and bias terms for scenario-shared and scenario-specific layers, respectively.

4 METHODOLOGIES

4.1 Overall Framework

IncMSR is first initialized with batch model M_0 , which is regarded as the warm-start model, and then updates subsequent models using only the incoming data and based on the previous model. An overview of IncMSR is shown in Figure 3(a). During model updates using data from day t , we first obtain $z_{t-1,d}^a$ and $z_{t-1,d}^s$ from previous model M_{t-1} , and then extract $z_{t,d}^a$ and $z_{t,d}^s$ from current training incremental model M_t . We apply different constraints on the scenario-shared and scenario-specific layers inspired by the metric learning objective to resolve transferability issues over three dimensions in a unified way. The idea behind this is that the scenario-shared representations should contain relatively general

and stable information for all scenarios while the scenario-specific representations should be more sensitive to the changes of scenario and time. All constraints are fused into a unified formulation and act as a supplementary penalty term to the existing MSR loss function, without increasing the number of parameters or model complexity.

4.2 Distance Quantification

Figure 3 (b) shows the distribution of CTR across four distinct scenarios over 13 consecutive days on the KuaiRand[5] dataset. A large shift in CTR can be found across scenarios and timestamps. Existing MSR models do not model these shifts and thus are insensitive to the changes of distribution over scenario and time dimensions.

Since calculating distributions directly with respect to scenarios and time is challenging, we quantify distances at the representation level. Eq. 2 gives the general definition of the distance function, where z_{t,d_i} is the representation of a specific scenario i at incremental step t . The two different representations are obtained from either scenario-shared layers or scenario-specific layers, depending on the imposed constraints in each dimension.

$$D(z_{t,d_i}, z_{t',d_j}) \triangleq d(z_{t,d_i}, f(z_{t',d_j})) \quad (2)$$

Here, d represents a similarity distance function, and for convenience, we employ the Euclidean distance. The summary function f is applied to z_{t',d_j} to compute summary statistics of the representation of scenario j at incremental step t' , e.g. compressing the representation into a real number with the commonly used squeeze functions such as max or mean methods [11, 12]. Specifically, we apply the mean pooling as the summary function:

$$f(z_{t',d_j}) = \frac{1}{n} \sum_{k=1}^n (z_{t',d_j})_k \quad (3)$$

4.3 Transfer over Scenario dimension

It is common to observe imbalanced label distributions, resulting in significant differences in average CTR values between different scenarios. To enhance information transfer across multiple scenarios in incremental mode, we design constraints for the scenario-shared and scenario-specific layers.

Scenario-shared layers are responsible for capturing common information across scenarios, and scenario-specific layers are dedicated to capturing scenario-specific characteristics. For the shared layers, we try to decrease the distance between scenario-shared representations, which aims at improving transferability and generalization across scenarios. On the other hand, for the scenario-specific layers, we increase the distance between scenario-specific

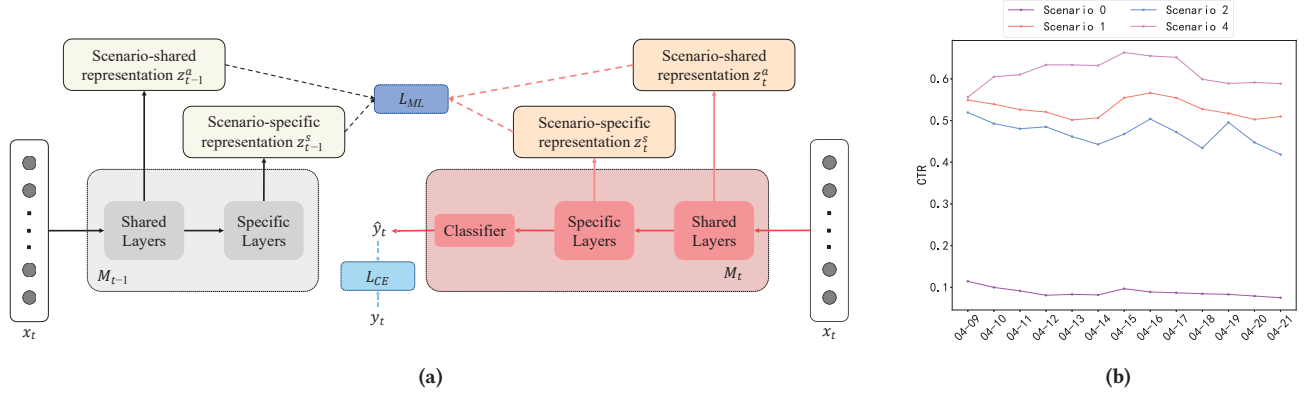


Figure 3: (a) Overview of IncMSR architecture with t indicating the incremental step. x_t is fed into previous model M_{t-1} to obtain representations from shared and scenario-specific layers. x_t is also used to update the current model M_t to obtain the updated representations. (b) CTR distribution of 4 scenarios over 13 consecutive days on the KuaiRand dataset [5].

representations to make them more discriminative for distinct scenarios. The distance between scenario-shared representations is computed as follows:

$$\mathcal{D}(z_{t,d_i}^a, z_{t,d_j}^a) = \sqrt{(z_{t,d_i}^a - f(z_{t,d_j}^a))^T (z_{t,d_i}^a - f(z_{t,d_j}^a))} \quad (4)$$

Similarly, the distance between scenario-specific representations is calculated as:

$$\mathcal{D}(z_{t,d_i}^s, z_{t,d_j}^s) = \sqrt{(z_{t,d_i}^s - f(z_{t,d_j}^s))^T (z_{t,d_i}^s - f(z_{t,d_j}^s))} \quad (5)$$

4.4 Transfer over Time dimension

In incremental learning, we ensure effective incorporation of new user behaviors, items and other changes that occur over time while maintaining prior knowledge from previous incremental step. To address catastrophic forgetting[17], we minimize the distance between step $t-1$ and t in scenario-shared representations for the same scenario i (Eq. 6). This preserves the stability of shared representations over time and ensures transferability.

$$\mathcal{D}(z_{t-1,d_i}^a, z_{t,d_i}^a) = \sqrt{(z_{t-1,d_i}^a - f(z_{t,d_i}^a))^T (z_{t-1,d_i}^a - f(z_{t,d_i}^a))} \quad (6)$$

Scenario-specific layers extract the scenario-specific representations, which can change differently over time according to different scenarios. As a result, we do not apply constraints over the scenario-specific layers.

4.5 Transfer over Time-Scenario dimension

When information is migrated from one scenario to another in incremental mode, we cannot only consider the migration of information from the scenario or time dimension independently but also the impact of the coexistence of two-dimensional migration. To address this, we propose a constraint for cross-dimensional migration. Specifically, we quantify the distance between the scenario-specific representation of scenario i at step $t-1$ and scenario j at step t :

$$\mathcal{D}(z_{t-1,d_i}^s, z_{t,d_j}^s) = \sqrt{(z_{t-1,d_i}^s - f(z_{t,d_j}^s))^T (z_{t-1,d_i}^s - f(z_{t,d_j}^s))} \quad (7)$$

The distance is designed to be maximized so that IncMSR can capture the distinctions between scenarios at different time steps and thus is more sensitive to changes in the distribution along with time and scenarios. The scenario-shared representations are

aligned in the aforementioned dimensions and thus are not taken into account in the time-scenario dimension.

4.6 Metric Learning Based Optimization

Metric learning-based optimization aims to improve the model's discriminative power by constructing a suitable metric function, such as MMD [2], to effectively capture the similarity and dissimilarity between different distributions. In our case, we utilize metric learning to facilitate the information transfer over scenario, time and time-scenario dimensions. By employing metric learning, we minimize positive distances (scenario-shared representations) and maximize negative distances (scenario-specific representations).

Our goal is to minimize the distance between $z_{t,d}^a$ which contains relatively general and stable information for all scenarios, while maximizing the distance between $z_{t,d}^s$ which should be more sensitive to the changes of time and scenario. We fuse these constraints in a unified formulation and denote the metric-based loss L_{ML} as:

$$L_{ML} = -1 \sum_{d_i \in D \setminus d_j} \log \frac{\exp(\mathcal{D}(z_{t-1,d_i}^a, z_{t,d_i}^a) + \mathcal{D}(z_{t,d_i}^a, z_{t,d_j}^a))}{\exp(\mathcal{D}(z_{t-1,d_i}^s, z_{t,d_j}^s) + \mathcal{D}(z_{t,d_i}^s, z_{t,d_j}^s))} \quad (8)$$

Besides, due to the different data volumes in different scenarios, scenarios with larger volumes of data may dominate the entire training process. We rescale the original \mathcal{D} in Eq. 2 by a factor of $1/N_{t,d_i}$, where N_{t,d_i} represents the data volume of scenario i at incremental step t , and the L_{ML} will be updated accordingly.

$$\mathcal{D}'(z_{t,d_i}, z_{t',d_j}) = \frac{\mathcal{D}(z_{t,d_i}, z_{t',d_j})}{N_{t,d_i}} \quad (9)$$

$$L_{ML} = -1 \sum_{d_i \in D \setminus d_j} \log \frac{\exp(\mathcal{D}'(z_{t-1,d_i}^a, z_{t,d_i}^a) + \mathcal{D}'(z_{t,d_i}^a, z_{t,d_j}^a))}{\exp(\mathcal{D}'(z_{t-1,d_i}^s, z_{t,d_j}^s) + \mathcal{D}'(z_{t,d_i}^s, z_{t,d_j}^s))} \quad (10)$$

Finally, we incorporate the metric learning-based loss L_{ML} into the cross entropy loss L_{CE} to enhance information transfer from multiple dimensions for both scenario-shared and scenario-specific representations. The hyper-parameter λ balances these different losses. The overall objective function is formulated as:

$$L = L_{CE}(Y, \hat{Y}) + \lambda L_{ML} + \mathcal{R} \quad (11)$$

$$L_{CE}(Y, \hat{Y}) = \sum_{y_i \in Y} L_{CE}(y_i, \hat{y}_i) \quad (12)$$

$$L_{CE}(y, \hat{y}) = -y \log \hat{y} - (1-y) \log(1-\hat{y}) \quad (13)$$

Algorithm 1 Training protocol in the t -th incremental step

Input: features of incoming data X_t ; labels of incoming data Y_t ; existing model M_{t-1}

Output: Model M_t

```

1: Initialize:  $L = \text{MaxInt}$ ;  $ep = 0$ ; Initialize  $M_t$  based on  $M_{t-1}$ 
2:  $\Theta \leftarrow$  feature embeddings using [34]  $\cup$  network parameters
3: for epochs do
4:    $z_{t-1,d}^a, z_{t-1,d}^s \leftarrow M_{t-1}(X_t)$  by Eq. 1
5:    $z_{t,d}^a, z_{t,d}^s \leftarrow M_t(X_t)$  by Eq. 1
6:    $M_t = \text{Train\_with\_IncMSR}(X_t, Y_t, z_{t-1,d}^a, z_{t-1,d}^s, z_{t,d}^a, z_{t,d}^s, \Theta)$ 
7: end for
8: Return:  $M_t$ 
9:
10: Train\_with\_IncMSR:
11:    $\hat{Y}_t = \text{Inference}(M_t, X_t)$ 
12:   Compute  $L_{ML}(z_{t-1,d}^a, z_{t-1,d}^s, z_{t,d}^a, z_{t,d}^s)$  by Eq. 10
13:   Compute cross entropy loss  $L_{CE}(Y_t, \hat{Y}_t)$  by Eq. 12
14:    $M_t \leftarrow \arg \min_{\Theta} (L_{CE}(Y, \hat{Y}) + \lambda L_{ML} + \mathcal{R})$ 
15:    $ep += 1$ 

```

where Y and \hat{Y} denote the ground-truth and outputs of the model, respectively, and \mathcal{R} denotes the regularization term. The overall algorithm is summarized in Algorithm 1.

5 EXPERIMENTS

In this section, we present experiments conducted on two datasets to answer the following questions:

- **RQ1:** What is the performance of IncMSR compared with baseline models?
- **RQ2:** How compatible is IncMSR with existing MSR models?
- **RQ3:** How does the information transfer over three dimensions affect the performance?

5.1 Experimental Setting

5.1.1 Dataset and Evaluation Protocols. We conduct extensive experiments on industrial and public datasets respectively, and both are collected from real-world commercial platforms.

- **KuaiRand** [5]. This dataset is collected from the recommendation logs of a video-sharing mobile app, Kuaishou. Each log contains comprehensive side information including explicit user IDs, interaction timestamps, and rich features for users and items. The dataset consists of 13 days of interactions logs from 2022-04-09 to 2022-04-21. We take attribute *tab* as the indicator of scenarios. Since the top four scenarios account for more than 96% of the total data volume, we select the top four scenarios in the dataset.
- **Taobao**.¹ This dataset is released by Alimama, an online advertising platform in China. Ad display/click logs from 2017-05-06 to 2017-05-13 are used. Following [4], we filter the samples of which user profile is missing, and then divide the dataset into 4 scenarios according to *City_level*.

Both datasets contain multiple scenarios and samples for several consecutive days, allowing us to perform daily incremental updates

for multi-scenario recommendation. We use the *click* as the target label. The statistics of the two datasets are shown in Table 1.

We apply the most commonly-used **AUC** (Area Under Curve) and **logloss** (cross-entropy) as our evaluation metrics for measurement of model performance. For industrial applications, even 0.1% improvement on AUC is significant [7, 39].

Table 1: The percentage of instances and average click-through rate (CTR) of each scenario.

Dataset	KuaiRand				Taobao			
Scenario	#0	#1	#2	#4	#1	#2	#3	#4
Percentage	13.64%	75.92%	3.57%	6.87%	10.26%	44.32%	25.20%	20.22%
CTR	9.20%	52.97%	48.00%	61.96%	5.04%	5.12%	5.06%	5.11%

5.1.2 Baselines and Implementation Details. To demonstrate the effectiveness and compatibility of the proposed framework, we compare IncMSR with three groups of models.

- **Batch-single.** Batch-single models are trained with data from a specific scenario using batch mode. Without loss of generality, we use a SOTA deep CTR model DCN [31] as backbone model. We train the batch model on the data in a fixed-size window $[s, s + \omega)$, $s \in [0, T - \omega]$. Batch- i ($i \in \{0, 1, 2, 3, 4\}$) represents the models with i days' delay at iteration i . Specifically, Batch-0 corresponds to the model trained on data from a specific scenario within the initial window $[0, \omega)$.
- **Batch-multi.** Batch-multi models are multi-task or multi-scenario models trained on data from all scenarios using batch mode. These methods jointly utilize samples from all scenarios for training. SharedBottom [23] shares parameters at bottom layers, and a specific three-layer fully-connected network is built for each scenario based on the shared parameters. PLE [27] is a state-of-the-art multi-task model that explicitly separates task-shared experts and task-specific experts and adopts a progressive routing mechanism to extract and separate deeper semantic knowledge gradually. STAR [25] is a state-of-the-art multi-scenario recommendation model that proposes a star topology structure consisting of shared and scenario-specific networks.
- **Incremental-single.** Incremental-single models are trained on data from a specific scenario in incremental mode. Without loss of generality, we select IncCTR [34] as the baseline for comparison. Because it is a classic incremental CTR method that consists of three decoupled modules to construct data, handle features, and fine-tune model parameters.

All incremental competitors adapt warm-start, which first train a model with batch mode on data in $[0, \omega)$, then update the subsequent models with only the data from the incoming day. We use adam [14] as the optimizer for all models and set the embedding size as 10. For each dataset, their settings are as follows.

- **KuaiRand settings.** ω is set to 7, T is set to 13. For batch mode and increment mode, learning rate is tuned from $[1e-3, 1e-4]$, batch size is set to 2000. The weight of L_{ML} is tuned from $[1e-1, 1e-2, 1e-3]$.
- **Taobao settings.** ω is set to 3, T is set to 8. For batch mode, learning rate is set to $1e-3$, batch size is set to 2000. For increment mode, learning rate is set to $1e-4$, batch size is set to 5000. The weight of L_{ML} is tuned from $[1e-1, 1e-2]$.

¹<https://tianchi.aliyun.com/dataset/dataDetail?dataId=56>

Table 2: Performance comparison of single-scenario and multi-scenario recommendation models in batch and incremental mode on the KuaiRand dataset. Batch-0 is the warm-start model. Comparisons are conducted between batch mode and incremental mode, e.g., Batch-1 vs. Incre-1. Boldface denotes the highest score. ★ represents significance level p-value < 0.05 over AUC.

Scenarios	Metrics	Batch-single(DCN)							Incremental-single(IncCTR)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Batch-6	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	Incre-6
#0	AUC	0.6801	0.6907	0.6601	0.5983	0.6187	0.5873	0.5726	0.6937	0.6698	0.6384	0.6741	0.6480	0.6248
	Logloss	0.3169	0.2840	0.2856	0.2938	0.2861	0.2738	0.2645	0.2847	0.2813	0.2829	0.2709	0.2705	0.2611
#1	AUC	0.7089	0.7037	0.6949	0.6869	0.6749	0.6666	0.6521	0.6988	0.6925	0.6973	0.6924	0.6941	0.6947
	Logloss	0.6189	0.6184	0.6287	0.6463	0.6498	0.6701	0.6887	0.6211	0.6270	0.6284	0.6352	0.6328	0.6308
#2	AUC	0.6857	0.6977	0.7012	0.7080	0.6509	0.5998	0.6067	0.6951	0.7055	0.7229	0.7211	0.6862	0.6773
	Logloss	0.6893	0.6964	0.6903	0.6843	0.6945	0.6898	0.6880	0.6940	0.6897	0.6842	0.6923	0.6865	0.6830
#4	AUC	0.6429	0.6452	0.6651	0.6464	0.6056	0.5968	0.6153	0.6438	0.6616	0.6599	0.6315	0.6493	0.6354
	Logloss	0.6213	0.6150	0.6076	0.6513	0.6638	0.6672	0.6568	0.6186	0.6091	0.6356	0.6525	0.6455	0.6496
Scenarios	Metrics	Batch-multi(SharedBottom)							IncMSR(SharedBottom)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Batch-6	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	Incre-6
#0	AUC	0.6797	0.6858	0.6650	0.6151	0.6175	0.6176	0.6092	0.6871*	0.6725*	0.6388*	0.6798*	0.6532*	0.6506*
	Logloss	0.5909	0.5833	0.6044	0.6423	0.6561	0.6620	0.6660	0.5892	0.5853	0.5832	0.5818	0.5782	0.5765
#1	AUC	0.7087	0.7053	0.6963	0.6878	0.6732	0.6669	0.6559	0.7066*	0.6946	0.6978*	0.6933*	0.6975*	0.6962*
	Logloss	0.6653	0.6563	0.6773	0.6916	0.6925	0.6934	0.6930	0.6615	0.6597	0.6627	0.6588	0.6570	0.6564
#2	AUC	0.6932	0.7214	0.7150	0.7361	0.7043	0.6900	0.6848	0.7275*	0.7280*	0.7564*	0.7280*	0.7083*	0.7240*
	Logloss	0.6528	0.6400	0.6553	0.6861	0.6927	0.6897	0.6888	0.6367	0.6263	0.5978	0.6238	0.6272	0.6260
#4	AUC	0.6706	0.6726	0.6795	0.6636	0.6300	0.6470	0.6560	0.6813*	0.6890*	0.6813*	0.6643*	0.6914*	0.6770*
	Logloss	0.6395	0.6252	0.6411	0.6824	0.6857	0.6863	0.6878	0.6372	0.6395	0.6403	0.6457	0.6392	0.6417
Scenarios	Metrics	Batch-multi(PLE)							IncMSR(PLE)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Batch-6	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	Incre-6
#0	AUC	0.6797	0.6694	0.6282	0.6014	0.5932	0.6062	0.5925	0.6748*	0.6703*	0.6383*	0.6552*	0.6419*	0.6542*
	Logloss	0.5969	0.5980	0.6204	0.6442	0.6555	0.6840	0.6819	0.5918	0.5880	0.5820	0.5832	0.5826	0.5806
#1	AUC	0.7084	0.7030	0.6923	0.6888	0.6706	0.6574	0.6505	0.6998	0.6934*	0.6955*	0.6922*	0.6944*	0.6923*
	Logloss	0.6692	0.6678	0.6911	0.6918	0.6926	0.6934	0.6930	0.6557	0.6542	0.6546	0.6535	0.6537	0.6538
#2	AUC	0.6978	0.7196	0.7223	0.7376	0.6947	0.6787	0.6884	0.7272*	0.7240*	0.7533*	0.7283*	0.6926*	0.7235*
	Logloss	0.6599	0.6585	0.6946	0.6906	0.6954	0.6906	0.6900	0.6466	0.6390	0.6102	0.6304	0.6385	0.6319
#4	AUC	0.6730	0.6733	0.6865	0.6653	0.6350	0.6520	0.6502	0.6785*	0.6878*	0.6772*	0.6646*	0.6876*	0.6758*
	Logloss	0.6353	0.6332	0.6665	0.6830	0.6851	0.6875	0.6886	0.6390	0.6327	0.6427	0.6485	0.6440	0.6481
Scenarios	Metrics	Batch-multi(STAR)							IncMSR(STAR)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Batch-6	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	Incre-6
#0	AUC	0.6845	0.6801	0.6508	0.5951	0.6062	0.6052	0.5968	0.6870*	0.6685*	0.6400*	0.6867*	0.6519*	0.6631
	Logloss	0.3671	0.4010	0.3754	0.4086	0.4866	0.3331	0.3938	0.3635	0.3343	0.4076	0.3275	0.3240	0.3142
#1	AUC	0.7062	0.7013	0.6904	0.6713	0.6510	0.6539	0.6458	0.6988	0.6922*	0.6948*	0.6912*	0.6950*	0.6916*
	Logloss	0.6219	0.6216	0.6338	0.6520	0.6938	0.6647	0.6624	0.6255	0.6414	0.6601	0.6401	0.6356	0.6402
#2	AUC	0.6903	0.7092	0.6893	0.7051	0.6719	0.6463	0.6740	0.7154*	0.7175*	0.7431*	0.7264*	0.6861*	0.7127*
	Logloss	0.6465	0.6351	0.6454	0.6211	0.7106	0.6809	0.6506	0.6254	0.6244	0.6021	0.6283	0.6438	0.6261
#4	AUC	0.6730	0.6782	0.6740	0.6478	0.6267	0.6376	0.6343	0.6695	0.6841*	0.6626*	0.6556*	0.6803*	0.6565*
	Logloss	0.6272	0.6329	0.6393	0.6770	0.7103	0.6842	0.6819	0.6229	0.6150	0.6423	0.6524	0.6335	0.6513

5.2 Overall Performance(RQ1)

The overall comparisons on KuaiRand and Taobao datasets are shown in Table 2 and Table 3, respectively. All experiments were repeated 5 times and the averaged results are reported. We summarize the observations as follows:

- Firstly, the results over both datasets show that IncMSR outperforms the baseline models. Compared with Incremental-single, IncMSR performs better, which proves that introducing data from other scenarios for joint modeling can bring significant improvement. According to the results in Table 2 and Table 4, we compare IncCTR(STAR) in Table 4 with the Batch-multi(STAR) model in Table 2. We observe that IncCTR(STAR) outperforms Batch-multi(STAR), which proves that incremental learning can update the model with users' latest preference and is more sensitive to the changes of the distribution. Compared with IncCTR, IncMSR obtains further improvement, which is due to the fine-grained design for transferring in scenario-shared and scenario-specific layers from different dimensions, specially, the time dimension and scenario-time dimension.
- Secondly, IncMSR exhibits more notable improvements in small scenarios. Specifically, on the KuaiRand dataset, IncMSR(STAR) shows absolute AUC gains of 4.38%, 3.42%,

1.83% over corresponding Batch-multi(STAR) in small scenarios #0. Similarly, the gains for IncMSR compared to Incremental-single are 0.81%, 1.55%, 2.12% in the same small scenarios. Small scenarios are particularly sensitive to changes like emerging interests, resulting in significant distribution shifts over time [30]. IncMSR improves the performance in small scenarios because it can provide more stability and adaptability in changing and sparse data environments, resulting in enhanced effectiveness.

- What's more, the results demonstrate IncMSR's ability to adapt to changing distribution over time as it receives new data. On the KuaiRand dataset, in the initial incremental step (Incre-1), the model is adjusting to the new data. IncMSR may experience a slight performance decline compared to Batch-1 for SharedBottom and PLE, while achieves comparable performance to Batch-1 for STAR. However, as the model continues to receive new data, IncMSR gradually adapts to the changing distribution, leading to improved performance compared to the corresponding Batch-Multi models from Incre-2 to Incre-6, compared to Batch-2 to Batch-6. These findings highlight IncMSR's capability to handle changing distribution and exploit the benefits of incremental learning, leading to enhanced performance in MSR models.

Table 3: Performance comparison of single-scenario and multi-scenario recommendation models in batch and incremental mode on the Taobao dataset.

Scenarios	Metrics	Batch-single(DCN)						Incremental-single(IncCTR)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	
#1	AUC	0.5994	0.5998	0.5987	0.5976	0.5987	0.6009	0.6069	0.6014	0.5971	0.6037	0.6124	
	Logloss	0.1980	0.1982	0.2001	0.2044	0.1916	0.1932	0.1957	0.1978	0.2020	0.1923	0.1945	
#2	AUC	0.6114	0.6089	0.6108	0.6092	0.6126	0.6103	0.6084	0.6103	0.6076	0.6134	0.6093	
	Logloss	0.2010	0.1996	0.2005	0.2018	0.1913	0.1972	0.1995	0.2002	0.2019	0.1912	0.1962	
#3	AUC	0.6069	0.6104	0.6099	0.6105	0.6127	0.6097	0.6076	0.6123	0.6109	0.6112	0.6111	
	Logloss	0.2007	0.1993	0.1980	0.1992	0.1918	0.1949	0.1983	0.1973	0.1981	0.1915	0.1944	
#4	AUC	0.6052	0.6082	0.6036	0.6054	0.6026	0.6074	0.6040	0.6076	0.6057	0.6031	0.6083	
	Logloss	0.2003	0.2002	0.2014	0.2036	0.1938	0.1962	0.1995	0.2006	0.2027	0.1933	0.1940	
Scenarios	Metrics	Batch-multi(SharedBottom)						IncMSR(SharedBottom)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	
#1	AUC	0.6105	0.6108	0.6114	0.6122	0.6131	0.6173	0.6167*	0.6116*	0.6116	0.6168*	0.6273*	
	Logloss	0.2086	0.2076	0.2257	0.2274	0.2196	0.2223	0.2059	0.2073	0.2098	0.1986	0.2005	
#2	AUC	0.6154	0.6122	0.6123	0.6098	0.6140	0.6128	0.6124*	0.6133*	0.6101*	0.6171*	0.6132*	
	Logloss	0.2123	0.2109	0.2261	0.2269	0.2193	0.2270	0.2088	0.2092	0.2102	0.1996	0.2045	
#3	AUC	0.6134	0.6145	0.6155	0.6166	0.6190	0.6161	0.6150*	0.6156*	0.6163*	0.6157*	0.6174*	
	Logloss	0.2110	0.2090	0.2263	0.2256	0.2180	0.2256	0.2076	0.2063	0.2071	0.2000	0.2027	
#4	AUC	0.6123	0.6145	0.6124	0.6093	0.6099	0.6126	0.6087	0.6141*	0.6098*	0.6108*	0.6136*	
	Logloss	0.2112	0.2106	0.2277	0.2288	0.2211	0.2266	0.2090	0.2095	0.2114	0.2016	0.2028	
Scenarios	Metrics	Batch-multi(PLE)						IncMSR(PLE)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	
#1	AUC	0.6139	0.6149	0.6088	0.6082	0.6144	0.6176	0.6151*	0.6122*	0.6101*	0.6184*	0.6282*	
	Logloss	0.2065	0.2051	0.2091	0.2290	0.2149	0.2030	0.2062	0.2072	0.2093	0.1992	0.2003	
#2	AUC	0.6139	0.6125	0.6123	0.6106	0.6144	0.6120	0.6109	0.6125*	0.6105	0.6159*	0.6113	
	Logloss	0.2110	0.2075	0.2107	0.2314	0.2203	0.2053	0.2094	0.2094	0.2098	0.2003	0.2040	
#3	AUC	0.6155	0.6131	0.6150	0.6148	0.6180	0.6154	0.6146*	0.6152*	0.6165*	0.6167	0.6164*	
	Logloss	0.2082	0.2093	0.2096	0.2295	0.2189	0.2047	0.2084	0.2068	0.2068	0.2006	0.2024	
#4	AUC	0.6126	0.6150	0.6111	0.6091	0.6098	0.6122	0.6104	0.6132*	0.6110*	0.6121*	0.6136*	
	Logloss	0.2108	0.2080	0.2112	0.2200	0.2198	0.2043	0.2092	0.2095	0.2104	0.2017	0.2019	
Scenarios	Metrics	Batch-multi(STAR)						IncMSR(STAR)					
		Batch-0	Batch-1	Batch-2	Batch-3	Batch-4	Batch-5	Incre-1	Incre-2	Incre-3	Incre-4	Incre-5	
#1	AUC	0.6125	0.6104	0.6116	0.6009	0.6071	0.6208	0.6136*	0.6085	0.6102*	0.6160*	0.6254*	
	Logloss	0.1972	0.1970	0.1986	0.2060	0.1917	0.1912	0.1951	0.1972	0.2001	0.1895	0.1908	
#2	AUC	0.6153	0.6109	0.6097	0.6074	0.6128	0.6111	0.6117*	0.6125*	0.6087*	0.6113	0.6067	
	Logloss	0.2011	0.2006	0.2027	0.2029	0.1919	0.1966	0.1992	0.1998	0.2017	0.1916	0.1962	
#3	AUC	0.6163	0.6118	0.6143	0.6092	0.6162	0.6122	0.6137*	0.6152*	0.6168*	0.6141	0.6126*	
	Logloss	0.1998	0.2014	0.1987	0.2001	0.1935	0.1962	0.1980	0.1970	0.1978	0.1917	0.1942	
#4	AUC	0.6156	0.6133	0.6112	0.6052	0.6077	0.6046	0.6104	0.6137*	0.6097*	0.6104*	0.6123*	
	Logloss	0.2004	0.2015	0.2019	0.2050	0.1960	0.1948	0.1991	0.2002	0.2026	0.1932	0.1939	

5.3 Compatibility(RQ2)

To evaluate the compatibility of IncMSR, we apply IncMSR to both multi-scenario model STAR and multi-task model PLE. The results in Table 2 and Table 3 show that significant improvements can be achieved when these models are applied with IncMSR.

The compatibility of IncMSR with existing MSR models arises from its ability to leverage incremental learning and capture transferable information. By seamlessly integrating with existing MSR models, IncMSR offers a practical framework that can leverage the strengths of multi-scenario recommendation models while incorporating incremental learning and information transfer. This compatibility ensures that IncMSR remains flexible and adaptable.

5.4 Ablation Study(RQ3)

To investigate the respective contributions of the three dimensions of information transfer in IncMSR, we conduct a series of ablation studies on KuaiRand dataset and take STAR as the backbone model. Models are trained with or without each of the three different dimensions of transfer and the results are compared in Table 4.

- **IncCTR** [34] denotes a commonly used incremental learning framework in SSR. In this case, we only minimize the cross-entropy loss to train the model in incremental mode.

- **IncMSR w/o S** denotes IncMSR without transfer over **scenario** dimension while still incorporating transfers over time and time-scenario dimensions.
- **IncMSR w/o T** denotes IncMSR without transfer over **time** dimension while still incorporating transfers over scenario and time-scenario dimensions.
- **IncMSR w/o TS** denotes IncMSR without transfer over **time-scenario** dimension while still incorporating transfers over scenario and time dimensions.

5.4.1 Transfer over Scenario Dimension. By comparing IncMSR with IncMSR-S (which does not involve scenario dimension transfer), we can observe that leveraging scenario dimension transfer contributes to the overall performance enhancement of IncMSR. The performance improvement is particularly significant for small scenarios such as Scenario #0. With scenario dimension transfer, IncMSR can effectively capture and utilize the shared information across scenarios. Therefore, it can improve overall model performance and mitigate the effects of data sparsity in small scenarios.

5.4.2 Transfer over Time Dimension. By comparing IncMSR with IncMSR-T (which does not involve time dimension transfer), the findings demonstrate performance enhancements during incremental updates. This suggests that the inclusion of time dimension

Table 4: Ablation study of the constraints on three dimensions of information transfer in IncMSR with STAR as the backbone on the KuaiRand dataset.

Scenarios	Models	Incre-1		Incre-2		Incre-3		Incre-4		Incre-5		Incre-6	
		Gains	AUC	Gains	AUC	Gains	AUC	Gains	AUC	Gains	AUC	Gains	AUC
#0	IncCTR	-	0.6847	-	0.6664	-	0.6406	-	0.6839	-	0.6485	-	0.6613
	IncMSR w/o S	0.25%	0.6872	0.22%	0.6686	-0.20%	0.6386	-0.83%	0.6756	-0.75%	0.6410	-0.34%	0.6579
	IncMSR w/o T	0.18%	0.6865	0.24%	0.6688	0.00%	0.6406	0.08%	0.6847	-0.05%	0.6480	0.15%	0.6628
	IncMSR w/o ST	0.38%	0.6885	0.11%	0.6675	0.08%	0.6414	0.16%	0.6855	0.10%	0.6495	-3.11%	0.6302
	IncMSR	0.23%	0.6870	0.21%	0.6685	-0.06%	0.6400	0.28%	0.6867	0.34%	0.6519	0.18%	0.6631
#1	IncCTR	-	0.6976	-	0.6915	-	0.6945	-	0.6906	-	0.6941	-	0.6909
	IncMSR w/o S	0.08%	0.6984	0.05%	0.6920	-0.02%	0.6943	0.01%	0.6907	0.08%	0.6949	0.09%	0.6918
	IncMSR w/o T	0.09%	0.6985	0.06%	0.6921	0.06%	0.6951	0.01%	0.6907	-0.03%	0.6938	0.01%	0.6910
	IncMSR w/o ST	0.08%	0.6984	0.12%	0.6927	0.09%	0.6954	0.17%	0.6923	0.01%	0.6942	-0.03%	0.6906
	IncMSR	0.12%	0.6988	0.07%	0.6922	0.03%	0.6948	0.06%	0.6912	0.09%	0.6950	0.07%	0.6916
#2	IncCTR	-	0.7152	-	0.7143	-	0.7435	-	0.7264	-	0.6806	-	0.7110
	IncMSR w/o S	0.03%	0.7155	0.23%	0.7166	-0.18%	0.7417	-0.28%	0.7236	-0.60%	0.6746	-0.07%	0.7103
	IncMSR w/o T	0.08%	0.7160	0.21%	0.7164	-0.15%	0.7420	-0.18%	0.7246	-0.07%	0.6799	0.06%	0.7116
	IncMSR w/o ST	-0.26%	0.7126	0.05%	0.7148	-0.40%	0.7395	-0.40%	0.7224	0.24%	0.6830	0.49%	0.7159
	IncMSR	0.02%	0.7154	0.32%	0.7175	-0.04%	0.7431	0.00%	0.7264	0.55%	0.6861	0.17%	0.7127
#4	IncCTR	-	0.6662	-	0.6787	-	0.6599	-	0.6530	-	0.6760	-	0.6532
	IncMSR w/o S	0.25%	0.6687	0.44%	0.6831	0.28%	0.6627	-0.08%	0.6522	0.13%	0.6773	0.13%	0.6545
	IncMSR w/o T	0.23%	0.6685	0.44%	0.6831	0.16%	0.6615	1.35%	0.6665	0.35%	0.6795	0.05%	0.6537
	IncMSR w/o ST	0.61%	0.6723	0.71%	0.6858	-0.31%	0.6568	-0.30%	0.6500	-0.12%	0.6748	-0.28%	0.6504
	IncMSR	0.33%	0.6695	0.54%	0.6841	0.27%	0.6626	0.26%	0.6556	0.43%	0.6803	0.33%	0.6565

transfer ensures consistent transferability of knowledge from previous steps to the current step stably. With the help of time dimension transfer, IncMSR can better adapt to the changing data distribution over time. This capability allows IncMSR to leverage previously acquired knowledge to better make predictions in subsequent steps.

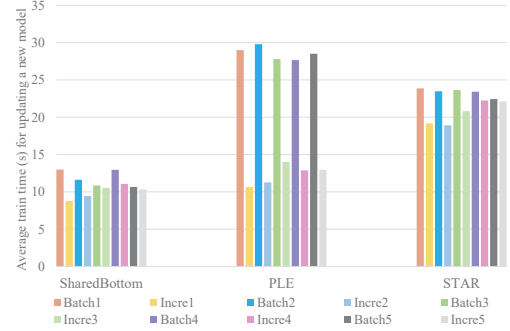
5.4.3 Transfer over Time-Scenario Dimension. By comparing IncMSR with IncMSR-TS (which does not involve time-scenario dimension transfer), we find that IncMSR achieves better performance and stability. With time-scenario dimension transfer, IncMSR exhibits the capability to distinguish and capture the characteristics and distinctions between scenarios at different time steps, resulting in improved sensitivity in the distribution across both time and scenario dimensions. The time-scenario dimension transfer in IncMSR enables the capture and utilization of transferable information, taking into consideration the impact of the coexistence of two-dimensional transfer.

Overall, the findings of the ablation studies underscore the importance of all three dimensions of information transfer in IncMSR.

5.5 Efficiency Analysis

To evaluate the efficiency of IncMSR, we conduct efficiency analysis by comparing the time consumption between IncMSR and the corresponding Batch-multi models on the Taobao dataset. All experiments are performed on an NVIDIA Tesla V100 GPU with 16G memory. The results are reported in Figure 4, which demonstrate that IncMSR achieves shorter training time compared to the Batch-multi models.

The efficiency of IncMSR can be attributed to its design and the use of metric learning to introduce a supplementary penalty term to the original MSR loss function. Importantly, this modification does not change the structure of the MSR model, nor does it increase its number of parameters or complexity. As a result, IncMSR retains its efficiency while ensuring that the training process remains efficient without compromising performance.

**Figure 4: Efficiency comparison between Batch-multi and IncMSR over consecutive days on the Taobao dataset.**

6 CONCLUSION

In this paper, we address the challenges of multi-scenario recommendation and propose an incremental learning approach to tackle them. Our proposed IncMSR approach involves three dimensions of information transfer: time, scenario, and time-scenario dimensions respectively. By quantifying the distances between representations in these dimensions and applying fine-grained constraints on scenario-shared and scenario-specific representations, IncMSR achieves better improvement in handling changing distribution and effectively leveraging information transfer. All constraints are elegantly fused based on metric learning framework. Comprehensive experiments are conducted to demonstrate the effectiveness and efficiency of the proposed approach. Besides, IncMSR is lightweight, model-agnostic, and can be easily extended to other MSR models, without increasing the number of parameters or complexity.

ACKNOWLEDGMENT

This work was partly supported by the Joint Funds of Guangdong Basic and Applied Basic Research Foundation (Grant No. 2019A1515110261) and Key Technology Projects in Shenzhen (Grant No. JSGG20220831110203007). And we thank MindSpore[1] for the partial support of this work, which is a new deep learning computing framework.

ETHICAL CONSIDERATION

In conducting our research and proposing the IncMSR approach, we have adhered to ethical considerations to ensure the integrity and social responsibility of our work. Our research primarily focuses on advancing the technical aspects of recommendation systems and does not involve direct interactions with human subjects or the collection of personal data. As such, potential ethical concerns related to informed consent, privacy, and data handling are minimized.

Our work aims to contribute to the field of recommendation systems by addressing technical challenges associated with multi-scenario recommendation in incremental mode. Throughout our research process, we have followed established research ethics guidelines and practices to ensure the accuracy, transparency, and rigor of our methods and results. We have also taken care to properly attribute prior works and provide appropriate citations to relevant sources to maintain academic integrity.

We acknowledge that, while our research primarily concerns technical advancements, the deployment and application of recommendation systems in real-world scenarios may raise broader ethical considerations related to user privacy, fairness, and potential algorithmic biases. We recognize their significance and encourage researchers and practitioners to approach the deployment of recommendation systems with careful consideration of these ethical implications.

In summary, our research on IncMSR has been conducted with a commitment to upholding ethical standards within the scope of our technical contributions.

REFERENCES

- [1] 2020. MindSpore. <https://www.mindspore.cn>.
- [2] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* 22, 14 (2006), e49–e57.
- [3] Jianxin Chang, Chenbin Zhang, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. 2023. Pepnet: Parameter and embedding personalized network for infusing with personalized prior information. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3795–3804.
- [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] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3953–3957.
- [6] Prachi Garg, Rohit Saluja, Vineeth N Balasubramanian, Chetan Arora, Anbumani Subramanian, and CV Jawahar. 2022. Multi-Domain Incremental Learning for Semantic Segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 761–771.
- [7] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2493–2500.
- [8] Renchu Guan, Haoyu Pang, Fausto Giunchiglia, Ximing Li, Xuefeng Yang, and Xiaoyue Feng. 2022. Deployable and Continuable Meta-learning-Based Recommender System with Fast User-Incremental Updates. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1423–1433.
- [9] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming session-based recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1569–1577.
- [10] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*. 193–201.
- [11] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.
- [12] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 169–177.
- [13] Minsoo Kang, Jaeyoo Park, and Bohyung Han. 2022. Class-Incremental Learning by Knowledge Distillation with Adaptive Feature Consolidation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16071–16080.
- [14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [15] 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.
- [16] Pan Li and Alexander Tuzhilin. 2021. Dual metric learning for effective and efficient cross-domain recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [17] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.
- [18] Zhuoran Liu, Leqi Zou, Xuan Zou, Caihua Wang, Biao Zhang, Da Tang, Bolin Zhu, Yijie Zhu, Peng Wu, Ke Wang, et al. 2022. Monolith: Real Time Recommendation System With Collisionless Embedding Table. *arXiv preprint arXiv:2209.07663* (2022).
- [19] 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.
- [20] Fei Mi, Xiaoyu Lin, and Boi Faltings. 2020. Ader: Adaptively distilled exemplar replay towards continual learning for session-based recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 408–413.
- [21] Danni Peng, Sinno Jialin Pan, Jie Zhang, and Anxiang Zeng. 2021. Learning an Adaptive Meta Model-Generator for Incrementally Updating Recommender Systems. In *Fifteenth ACM Conference on Recommender Systems*. 411–421.
- [22] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 2020. Gag: Global attributed graph neural network for streaming session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 669–678.
- [23] Caruana Rich. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
- [24] 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.
- [25] 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.
- [26] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems* 29 (2016).
- [27] 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.
- [28] Xing Tang, Yang Qiao, Yuwen Fu, Fuyuan Lyu, Dugang Liu, and Xiuqiang He. 2023. OptMSM: Optimizing Multi-Scenario Modeling for Click-Through Rate Prediction. *arXiv preprint arXiv:2306.13382* (2023).
- [29] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 world wide web conference*. 729–739.
- [30] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*. 1977–1987.
- [31] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
- [32] Weiqing Wang, Hongzhi Yin, Zi Huang, Qinyong Wang, Xingzhong Du, and Quoc Viet Hung Nguyen. 2018. Streaming ranking based recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 525–534.
- [33] Yichao Wang, Huifeng Guo, Bo Chen, Weiwen Liu, Zhirong Liu, Qi Zhang, Zhicheng He, Hongkun Zheng, Weiwei Yao, Muyu Zhang, et al. 2022. CausalInt: Causal Inspired Intervention for Multi-Scenario Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4090–4099.
- [34] Yichao Wang, Huifeng Guo, Ruiming Tang, Zhirong Liu, and Xiuqiang He. 2020. A Practical Incremental Method to Train Deep CTR Models. *arXiv preprint arXiv:2009.02147* (2020).

- [35] Yuzhe Yang, Hao Wang, and Dina Katabi. 2022. On Multi-Domain Long-Tailed Recognition, Generalization and Beyond. *arXiv preprint arXiv:2203.09513* (2022).
- [36] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to retrain recommender system? A sequential meta-learning method. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1479–1488.
- [37] 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. *arXiv preprint arXiv:2208.11457* (2022).
- [38] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2021. Cold-start sequential recommendation via meta learner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4706–4713.
- [39] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.
- [40] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696* (2021).