

# DIIT: A Domain-Invariant Information Transfer Method for Industrial Cross-Domain Recommendation

Heyuan Huang\*

OPPO

Shenzhen, Guangdong, China

huangheyuan2@oppo.com

Xingyu Lou\*

OPPO

Shenzhen, Guangdong, China

louxingyu@oppo.com

Chaochao Chen

College of Computer Science and

Technology

Zhejiang University

Hangzhou, Zhejiang, China

zjuccc@zju.edu.cn

Pengxiang Cheng

OPPO

Shenzhen, Guangdong, China

chengpengxiang@oppo.com

Yue Xin

OPPO

Shenzhen, Guangdong, China

xinyue@oppo.com

Chengwei He

OPPO

Shenzhen, Guangdong, China

hechengwei@oppo.com

Xiang Liu

OPPO

Shenzhen, Guangdong, China

liuxiang10@oppo.com

Jun Wang†

OPPO

Shenzhen, Guangdong, China

junwang.lu@gmail.com

## KEYWORDS

Cross-Domain Recommendation, Incremental Learning, Adversarial Learning, Knowledge Distillation

## ACM Reference Format:

Heyuan Huang, Xingyu Lou, Chaochao Chen, Pengxiang Cheng, Yue Xin, Chengwei He, Xiang Liu, and Jun Wang. 2024. DIIT: A Domain-Invariant Information Transfer Method for Industrial Cross-Domain Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3627673.3679782>

## 1 INTRODUCTION

Large-scale commercial platforms typically contain multiple domains, and users are divided into many domains for different purposes, which leads to the data distribution shift over domains as shown in Figure 1 (a). Therefore, Cross-Domain Recommendation (CDR) has emerged to transfer information across domains [2, 35, 44]. Most existing CDR methods are based on an ideal static assumption that users’ interests do not change much in a short period [24, 37, 40]. However, as shown in Figure 1 (b), in the industrial RS environment, users’ immediate interests are constantly changing, which leads to the data distribution shift over time [36]. Therefore, it is indispensable to improve the efficiency of CDR methods to capture users’ interests immediately.

In this paper, we focus on the problem of how to maximize the transmission of beneficial information across domains in the industrial RS environment. Nevertheless, our work faces the following two challenges:

**CH1:** *How to improve the effectiveness of CDR methods in the industrial RS environment?* Since information in each domain can be divided into domain-specific and domain-invariant, where the former is only beneficial to the domain itself, the latter is beneficial to multiple domains [25, 43]. Therefore, indiscriminately transferring information from the source domain to the target domain

## ABSTRACT

Cross-Domain Recommendation (CDR) have received widespread attention due to their ability to utilize rich information across domains. However, most existing CDR methods assume an ideal static condition that is not practical in industrial recommendation systems (RS). Therefore, simply applying existing CDR methods in the industrial RS environment may lead to low effectiveness and efficiency. To fill this gap, we propose DIIT, an end-to-end Domain-Invariant Information Transfer method for industrial cross-domain recommendation. Specifically, We first simulate the industrial RS environment that maintains respective models in multiple domains, each of them is trained in the incremental mode. Then, for improving the effectiveness, we design two extractors to fully extract domain-invariant information from the latest source domain models at the domain level and the representation level respectively. Finally, for improving the efficiency, we design a migrator to transfer the extracted information to the latest target domain model, which only need the target domain model for inference. Experiments conducted on one production dataset and two public datasets verify the effectiveness and efficiency of DIIT.

## CCS CONCEPTS

• Information systems → Recommender systems.

\*Both authors contributed equally to this research.

†Corresponding author.

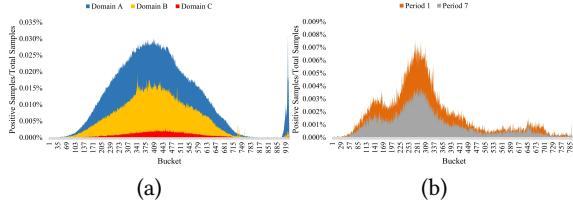
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CIKM '24, October 21–25, 2024, Boise, ID, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0436-9/24/10...\$15.00

<https://doi.org/10.1145/3627673.3679782>



**Figure 1: Visualization of the data distribution shift over domains (a) and time (b) using PCA. Area under curve represents CTR, and the data is collected from our production system.**

may be harmful. To tackle this challenge, most existing CDR methods assume the existence of overlapped samples between different domains, and regard these samples as "bridges" to transfer domain-invariant information across domains [3, 4, 42]. However, in the industrial RS environment, it is difficult to fully obtain overlapped samples due to reasons like the large number of source domains involved and privacy protection, which seriously harm the effectiveness of these CDR methods.

**CH2: How to improve the efficiency of CDR methods in the industrial RS environment?** Since large-scale commercial platforms contain multiple domains and each domain maintains its own model that trains in the incremental mode [1, 20, 34], directly applying CDR methods to the industrial RS environment is sub-optimal. Recently, several works were proposed to specifically optimize CDR methods in the industrial RS environments and have achieved good results. However, we argue that these works fail to solve **CH2** well because they need additional computation and storage resources [40] or need to keep the source domain model as external information to assist the inference in the target domain [24], which leads to low efficiency.

To address the above challenges, in this paper, we propose DIIT, an end-to-end **D**omain-**I**nvariant **I**nformation **T**ransfer method for industrial cross-domain recommendation. For **CH1**, we design two extractors to fully extract the domain-invariant information. The first one named the domain-invariant information extractor at the domain level, which is composed of a gating network and uses the target domain model to adaptively guide the aggregation of multiple source domain models. The other one named the domain-invariant information extractor at the domain level, which is composed of an adversarial network and uses adversarial learning to align the distribution of representations output by source domain models and the target domain model respectively. Through these two extractors, we can better extract domain-invariant information. For **CH2**, we design a domain-invariant information migrator, which is composed of a multi-spot knowledge distillation (KD) network to transfer the extracted information to the latest target domain model. It is worth noting that, KD have a characteristic that not only allows a flexible development of teacher and student models with different architectures, but also requires only the student model in the inference phase. Therefore, we can transfer information from multiple source domain models with different structures, and keep only the target domain model for inference, which is more practical in the industrial RS environment and execute an efficient inference.

In summary, our contributions are as follows:

- We first discuss an important but neglected research direction that how to perform efficient recommendations across domains in the industrial RS environment. As a potential solution, we propose DIIT, an end-to-end domain-invariant information transfer method for industrial cross-domain recommendation.
- We further analyze the challenges of how to improve the performance and efficiency of CDR methods in the industrial RS environment. For the former challenge, we design two extractors to extract domain-invariant information between the source domains and the target domain at two levels respectively. For the latter challenge, we design a migrator to transfer domain-invariant information from the source domain models to the target domain model, while only keeping the target domain model for inference.
- We finally conduct extensive experiments on three datasets of different magnitudes, including one production dataset and two public datasets, to verify the effectiveness and efficiency of DIIT.

## 2 RELATED WORKS

### 2.1 Cross-Domain Recommendation in Incremental Learning

Cross-Domain Recommendation (CDR) is widely used to transfer domain-invariant information across domains [2, 35, 44]. Specifically, CDR methods divide different domains into source and target domains [45], and on the one hand, obtain the domain-invariant information from the source domain to improve the effectiveness of the model in the target domain. On the other hand, preserves as much domain-specific information of the target domain as possible [6, 12, 27, 39].

However, most of the above methods cannot be well adapted to the industrial RS environment [1, 20, 34]. InMSR [37] is a recent multi-domain incremental method that combines information from domain, time, and time-domain dimensions respectively. KEEP [40] and CTNet [24] involve the cross-domain recommendation of incremental learning, which is closest to the research direction in this paper. Among them, KEEP is a two-stage industrial knowledge extraction and plugging framework, but still suffers from disadvantages such as the need for additional computation and storage resources, and the need for a synchronization strategy to ensure consistency between the knowledge extraction and inference. CTNet is a lightweight method that transfers information from the time-evolving source domain to the time-evolving target domain. CTNet first extends the target domain model to be a two-tower architecture, including a source tower and a target tower, which are consistent with the corresponding source/target domain model respectively. Then, design an adapter to transfer information from the source tower to the target tower both in the training and inference phases, which leads to lower efficiency. Furthermore, we argue that the methods above cannot fully extract the domain-invariant information through simple addition or mapping layers. As a potential solution, in our paper, we proposed DIIT, which can efficiently transfer domain-invariant information in the industrial RS environment.

## 2.2 Adversarial Learning

Adversarial learning is a research direction of transfer learning, which can extract domain-invariant information by aligning domains [9, 11, 17, 19]. Its core is to design a discriminator to distinguish which domain the sample comes from. By confusing the discriminator, it can obtain domain-invariant information across domains. Adversarial learning works because it is equivalent to minimizing the Jensen-Shannon divergence between different distributions [15]. DANN [13] is an early work that uses adversarial learning to align the labeled source domain and unlabeled target domain, thereby classifying target domain data with the help of the source domain. In recommendation, adversarial learning has also received widespread attention [16, 30, 31, 41]. For example, su et al.[30] proposed an adversarial learning-based framework to train the target model together with a pre-trained source model. In this paper, to transfer domain-invariant information across domains in a fine-grained manner, we design an adversarial learning-based extractor to learn domain-invariant information by aligning the distribution of representations output by the source domain models and the target domain model respectively.

## 2.3 Knowledge Distillation

Knowledge distillation (KD) is a model compression method based on the teacher-student learning framework [5, 26] that was first proposed by Hinton et al. [18]. In addition to model compression, KD can also be used to train a unified model to represent multiple models [46]. When these models are come from different domains, information is transferred across domains. Existing KD methods can be divided into one-spot and multi-spot distillation [29, 32], where one-spot KD methods only transfer information from one layer of the teacher model, usually the logit layer. However, different layers of the teacher model have different semantic information, so multi-spot distillation is proposed, which provides more supervision signals for the student model by transferring the multi-layer information from the teacher network. In this paper, we use multi-spot KD to transfer information from multiple source domain models to the target domain model.

# 3 METHODOLOGY

## 3.1 Problem Formulation

Most existing CDR methods assume an ideal static assumption that users' interests do not change much in a short period [24, 37, 40]. Therefore, simply applying them to the industrial RS environment will result in low effectiveness and efficiency. In this paper, we consider a cross-domain recommendation task in the industrial RS environment that contains multiple domains, each domain maintains its own model and is trained in the incremental mode. On this basis, our goal is to efficiently extract and transfer the domain-invariant information, thereby ultimately improve the recommendation effectiveness in the target domain. Formally, we take the samples from the target domains in the period  $t$  as  $D^{Tar,t} = (x^{Tar,t}, y)$ , and collect source domain samples and target domain samples at a ratio of 1:1 to construct a mixed dataset in the period  $t$  as  $D^{Mix,t} = (x^{Mix,t}, d)$ , where the size of  $D^{Mix,t}$  is consistent with the size of the target domain dataset  $D^{Tar,t}$ .  $x^{Tar,t}$  and  $x^{Mix,t}$  represent samples from

the above two datasets respectively.  $y \in \{0, 1\}$  is the label that indicates whether the sample was clicked.  $t$  represents the period when the sample was collected.  $d \in \{0, 1\}$  is the domain indicator that indicates whether the sample is from the target domain. Furthermore, we use  $S_n^t(\cdot)$  to represent the model in each source domain in the period  $t$  respectively, and use  $T^t(\cdot)$  to represent the model in the target domain in the period  $t$ , where  $n \in \{1, \dots, N\}$  and  $N$  represents the number of source domains. Simply speaking, we want to transfer as much domain-invariant information as possible from source domain models  $S_n^t(\cdot)$  to the target domain model  $T^t(\cdot)$  while preserving the domain-specific information of the target domain in the period  $t$  of incremental learning.

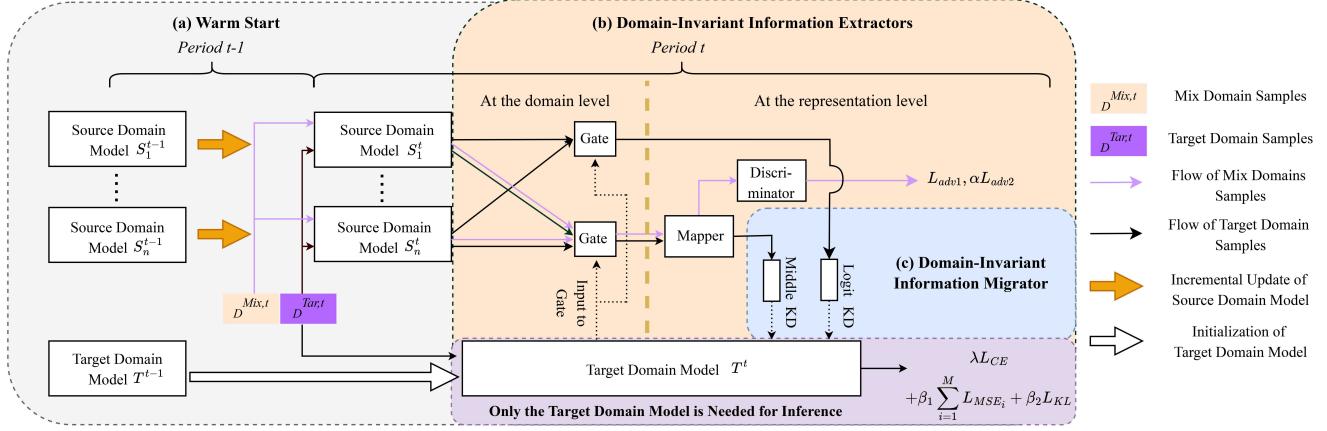
## 3.2 Overall Framework

The main framework of DIIT is shown in Figure 2, which mainly consists of three modules: a warm start module, a domain-invariant information extractor module and a domain-invariant information migrator module. The work process can be split into training and inference phases:

- **Training:** Firstly, we design a warm start module to simulate the industrial RS environment, which maintains a unique model in each domain and is trained in the incremental mode independently (see part (a) of Figure 2). Secondly, we design a domain-invariant information extractor module that consists of two extractors to extract the domain-invariant information at the domain level and the representation level respectively (see part (b) of Figure 2). Specifically, the first extractor uses the target domain model to guide the aggregation of multiple source domain models adaptively through a gating network. The second extractor uses an adversarial network to align the distributions of the aggregated source domain representations and the target domain representations. Thirdly, we design a domain-invariant information migrator, which is composed of a multi-spot knowledge distillation (KD) network to transfer the extracted domain-invariant information from the source domain models to the target domain model robustly (see part (c) of Figure 2). Finally, we optimize the above modules synchronously with the recommendation task in the target domain, thereby efficiently utilizing both the domain-specific information from the target domain and the domain-invariant information from the source domains to improve the recommendation effectiveness in the target domain.
- **Inference:** After training, the domain-invariant information from the source domains has actually been fully learned by the target domain model. And due to the characteristic of KD that requires only the student model for inference, we only need to keep the target domain model during the inference phase, which undoubtedly improves the efficiency of DIIT.

## 3.3 Warm Start

First of all, to simulate the industrial RS environment, each domain maintains an independent model that is trained in the incremental mode to capture domain-specific information. Taking the period  $t$  as an example, we first learn the period  $t$ 's model of each source domain independently using the latest incoming data, which is



**Figure 2: Model Architecture of the proposed DIIT.** DIIT consists of three parts: A warm start module, a domain-invariant information extractor module and a domain-invariant information migrator module. Note that only the target domain model in the bottom right region is used for inference. The solid arrows in violet and black represent the sample flow of the mix dataset and the target domain dataset respectively. The larger arrows in orange and white represent incremental update of the source domain models and initialization of the target domain model respectively.

shown by the orange arrows in part (a) of Figure 2. For the target domain model, we initialize the period  $t$ 's model using the period  $t-1$ 's model in a warm start manner, which as shown by the white arrow in part (a) of Figure 2. As mentioned in section 3.1, in the cross-domain recommendation task we consider, the structure of each source domain model can be different, and it is more practical in the industrial RS environment. In addition, since DIIT is plug-and-play, assuming that we plug the proposed DIIT for the first time in period  $t$ , there is no need to train the target domain model from scratch using all available data, which is undoubtedly efficient.

### 3.4 Domain-invariant Information Extractors

As mentioned above, most existing CDR methods require large amounts of source domain data, which is impractical in the industrial RS environment [3, 42]. To handle this problem and fully mine the domain-invariant information, as shown in part (b) of Figure 2, we design two extractors to extract the domain-invariant information from the source domain model at the domain level and the representation level respectively. Our method requires only the source domain models and a small number of source domain samples for training, we will introduce them in the rest of this section.

**3.4.1 at the domain level.** To extract the domain-invariant information at the domain level while ensuring the training time does not significantly increase as the number of source domains increases, we design the first extractor to aggregate the representations and the category probabilities (i.e. logits) output by multiple source domain models. Strategically, we input the period  $t$ 's target domain samples into the latest source and target domain models respectively. Next, through a gating network, the target domain model representations are used to adaptively guide the aggregation of representations and logits of each source domain model. Specifically, the output of the domain-invariant information extractor at the

domain level is formulated as:

$$\mathbf{e}_s^t = \sum_{n=1}^N g_{s_n}^t \mathbf{e}_{s_n}^t, \quad \mathbf{Z}_s^t = \sum_{n=1}^N g_{s_n}^t \mathbf{Z}_{s_n}^t, \quad (1)$$

where  $\mathbf{e}_{s_n}^t, \mathbf{Z}_{s_n}^t = S_n^t(x^{Tar,t})$  represents the representation and the logit output by the  $n$ -th source domain model in the period  $t$ .  $g_{s_n}^t$  represents the output of the gating network, which is composed of a two-layer multi-layer perceptron (MLP), and uses the softmax function as activation:

$$g_{s_n}^t = \text{softmax}(W_{s_n}^t(\mathbf{e}_T^t)), \quad (2)$$

where  $W_{s_n}^t$  represents the transformation matrix of the gating network,  $\mathbf{e}_T^t = T^t(x^{Tar,t})$  represents the representation output by the target domain model in the period  $t$ .

**3.4.2 at the representation level.** Despite obtaining  $\mathbf{e}_s^t$  that contains the domain-invariant information, we argue that it is still necessary to obtain more invariant information from a fine-grained perspective. Therefore, we design the second extractor to extract the domain-invariant information from multiple source domain models at the representation level. Strategically, we align the representation distributions of the source domains and the target domain through an adversarial network, which can separate the domain-invariant information and the domain-specific information.

As shown in part (b) of Figure 2, the second extractor consists of two parts, a mapper and a discriminator. As shown by the blue arrow in the figure, we input the representations of samples from different domains into the mapper and the discriminator successively. Note that the purpose of the discriminator is to determine whether the sample comes from the target domain, while the purpose of the discriminator is to confuse it. This is actually a min-max problem, and when the discriminator cannot correctly determine the source of the representations, it is considered that the distributions of the source domain representation and the target domain representation

have been aligned. Specifically, we feed samples from the mixed dataset  $D^{Mix,t}$  into each source domain model separately, and use the first extractor to obtain the aggregated representations  $\mathbf{e}_{adv}^t$ . Next, we feed  $\mathbf{e}_{adv}^t$  into the mapper, which consists of a simple linear transformation:

$$\mathbf{e}_{adv}^t = W_{mapper}^t(\mathbf{e}_{adv}^t), \quad (3)$$

where  $W_{mapper}^t$  represents the transformation matrix of the mapper. Next, we feed  $\mathbf{e}_{adv}^t$  into the discriminator to distinguish whether  $\mathbf{e}_{adv}^t$  comes from the source domains or the target domain. The discriminator consists of a two-layer MLP, and uses the sigmoid function as activation.

$$\hat{d} = \text{sigmoid}(W_{dis}^t(\mathbf{e}_{adv}^t)), \quad (4)$$

where  $W_{dis}^t$  represents the transformation matrix of the discriminator. After getting the predicted  $\hat{d}$ , Our goal is to minimize the cross-entropy loss as:

$$L_{adv1}(\Theta_{dis}) = -\frac{1}{|D^{Mix,t}|} \sum_{i=1}^{|D^{Mix,t}|} d_i \log \hat{d}_i - (1-d_i) \log(1-\hat{d}_i), \quad (5)$$

where  $\Theta_{dis}^t$  represents the parameters of the discriminator. Note that the discriminator and other parts of the model will be trained successively in one epoch to achieve an effect similar to the gradient inversion. That is, during the optimization of  $L_{adv1}$ , all parameters will be frozen except that of the discriminator. After optimizing the discriminator, what we have to do is to confuse it through the mapper. Specifically, we maximize a loss function  $L_{adv2}$  while freeze parameters of the discriminator during the optimization:

$$L_{adv2}(\Theta_{mapper}^t) = -\frac{1}{|D^{Mix,t}|} \sum_{i=1}^{|D^{Mix,t}|} d_i \log \hat{d}_i - (1-d_i) \log(1-\hat{d}_i), \quad (6)$$

where  $\Theta_{mapper}^t$  represents the parameters of the mapper. The optimization of  $L_{adv2}$  will be performed together with other losses of DIIT, we will introduce it in detail in section 3.6. Next, we pass  $\mathbf{e}_s^t$  obtained from Eq. (1) through the mapper, so that it can obtain the domain-invariant information at the representation level.

### 3.5 Domain-invariant Information Migrator

Once the representations with domain-invariant information are obtained, how to transfer them is the next challenge. In order to ensure efficient inferring and improve the generalization of the target domain model, we design a multi-spot KD network to transfer domain-invariant information efficiently from the source domain models to the target domain model, which is shown in part (c) of Figure 2. Strategically, to fully obtain multi-level information, we adopt the middle layer distillation and the logit layer distillation, which will be introduced in the rest of this section.

**3.5.1 middle layer distillation.** We first let the middle layer representations of the target domain model imitate the corresponding representations of the source domain models, thereby accepting information across domains robustly. Specifically, we use the Mean Square Error (MSE) loss to minimize the distribution gap between  $\mathbf{e}_s^t$  output by the domain-invariant information extractor at the representation level and the corresponding representation  $\mathbf{e}_T^t$  output

by the target domain model:

$$L_{MSE} = \frac{1}{|D^{Tar,t}|} \sum_{i=1}^{|D^{Tar,t}|} (W_{KD}^t \mathbf{e}_s^t - \mathbf{e}_T^t)^2 \quad (7)$$

where  $\mathbf{e}_T^t = f_T(x^{Tar,t})$  represents the representation output by the target model.  $W_{KD}^t$  represents a transformation matrix in case when  $\mathbf{e}_s^t$  and  $\mathbf{e}_T^t$  have different shapes. All in all,  $\mathbf{e}_s^t$  is actually similar to the supervision signal. By imitating  $\mathbf{e}_s^t$ , the target domain model can obtain cross-domain information. Note that as stated in [28], the effect of middle layer distillation on the target domain model can also be regarded as a regularization, so in order to avoid over-regularization, the number and position of middle layer distillation need to be carefully selected. In this paper, for convenience, we select only the representation output by the last hidden layer as the intermediate representation by default.

**3.5.2 logit layer distillation.** We further treat the logits output by the activation layer as soft labels, and then directly distill the extracted domain-invariant information through a high-temperature distillation method:

$$L_{KL} = \frac{1}{|D^{Tar,t}|} \sum_{i=1}^{|D^{Tar,t}|} \sigma\left(\frac{Z_{S_i}^t}{\tau}\right) \left( \log\sigma\left(\frac{Z_{S_i}^t}{\tau}\right) - \log\sigma\left(\frac{Z_{T_i}^t}{\tau}\right) \right), \quad (8)$$

where  $\tau$  represents the temperature coefficient, and it can control the discrepancy between different distributions and precisely determine the difficulty level of KD [23].  $\sigma(\cdot)$  is the softmax function.  $Z_{S_i}^t$  and  $Z_{T_i}^t$  are soft labels of source and target domain models respectively. Compared with the hard label that only has a value of 0 or 1, the soft label can provide a large amount of information contained in the negative label.

The overall loss of the KD in this paper is as follows:

$$L_{KD} = \beta_1 \sum_{i=1}^M L_{MSE_i} + \beta_2 L_{KL}, \quad (9)$$

where  $M$  represents the number of the middle layer distillation. It is worth noting that the domain-invariant information migrator module allows a variety of source domain information storage approaches. For example, when you have a source domain model with a small number of parameters, you can save the model and output the information in the form of gradient truncation during the training phase. And when you have a domain with a small amount of data, you can also pre-store the information output by the source domain model.

### 3.6 Optimization and Inference

In this section, we will first introduce the optimization of the recommendation task in the target domain, then introduce the overall optimization process and inference in detail.

**3.6.1 Train the Target domain Model.** Through the above operations, the target domain model has fully obtained the domain-invariant information. Next, we will mine the domain-specific information of the target domain through a model that is trained in the incremental mode. Specifically, we train the period  $t$ 's target domain model using only the latest incoming data. Our goal is to

minimize the cross-entropy loss  $L_{CE}$  as:

$$\hat{y} = f_T^t(x^{Tar,t}), \quad (10)$$

$$L_{CE} = -\frac{1}{|D^{Tar,t}|} \sum_{i=1}^{|D^{Tar,t}|} y_i \log \hat{y}_i - (1 - y_i) \log (1 - \hat{y}_i), \quad (11)$$

where  $\hat{y}$  represents the prediction of the target domain model.

**3.6.2 Overall Optimization.** Finally, we will perform the overall loss optimization:

$$L_{total} = \begin{cases} L_{adv} & \text{step 1} \\ \lambda L_{CE} + \alpha L_{adv2} + \beta_1 \sum_{i=1}^M L_{MSE_i} + \beta_2 L_{KL} & \text{step 2} \end{cases} \quad (12)$$

where  $\lambda$ ,  $\alpha$ ,  $\beta_1$  and  $\beta_2$  is the hyper-parameters used to control the importance of different losses of DIIT, thereby ensuring that the target domain model preserves domain-specific information while accepting domain-invariant information.

It is worth noting that, as described in Section 3.4, our optimization process is divided into two steps as shown in Eq. (12), which are performed in each epoch: first, update the discriminator parameter  $\Theta_{dis}^t$  by optimizing  $L_{adv1}$ . Next, update the remaining parameters by optimizing  $L_{CE}$ ,  $L_{adv2}$  and  $L_{KD}$  simultaneously. In addition, the back-propagation of  $L_{KD}$  cannot affect the parameters of the source domain models, thereby avoiding the loss of accuracy due to co-adaption of the source domain models and the target model [29].

**3.6.3 Inference.** In the inference phase, Since KD can transfer information from the teacher model to the student model, and only the student model is needed for inference. Therefore, unlike models proposed by [24, 40], we only need to use the target domain model to make predictions for inference according to Eq. (10), which significantly improves the efficiency.

## 4 EXPERIMENTS

In this section, we first introduce the experimental setup. Next, we design experiments to answer the following research questions:

- **RQ1:** How does the effectiveness and efficiency of DIIT compared with state-of-the-art single-domain/cross-domain methods?
- **RQ2:** As a pluggable unit, does DIIT perform well when used with different backbone models?
- **RQ3:** How about the impact of each part on the overall model?
- **RQ4:** What will be the impact when DIIT is introduced at different periods in incremental training?

### 4.1 Experimental Setup

**4.1.1 Datasets.** We conduct extensive experiments on one production dataset and two public datasets. These datasets have different magnitudes, ranging from hundreds of millions, tens of millions, and millions. The statistics of them are shown in Table 1.

- **Production.** Production is an advertising CTR prediction dataset that we collected in OPPO production system, which is one of the largest consumer electronics manufacturers in

**Table 1: Statistics of datasets**

	Dataset	Domain		
		A	B	C
Production	Instances	340M	170M	230M
	Percentage	45.8%	22.8%	31.4%
Taobao	Instances	8M	2M	8M
	Percentage	45.1%	10.3%	44.6%
KuaiRand	Instances	1M	3M	400K
	Percentage	22.4%	69.1%	8.5%

the world. Interaction logs from 2024-01-02 to 2024-01-08 are used for training and the next day for testing. Through discrete feature "domain ID", the dataset can be divided into 10 domains. We select three of these domains for training and select the one with the smallest number of samples as the target domain. The production dataset is a large-scale dataset, in order to reduce the data size, we execute 10% random negative sampling, and the ratio of positive and negative samples after it is about 1:5.

- **Taobao.**<sup>1</sup> Taobao is a display advertising CTR prediction dataset provided by Alibaba. Interaction logs from 2017-05-06 to 2017-05-12 are used for training and the next day for testing. Following [37] and [7], we divide the dataset into different domains based on the discrete feature "city\_level". We select three domains for training and select the one with the smallest number of samples as the target domain.
- **KuaiRand [14].** KuaiRand is a random recommendation dataset provided by Kuaishou. Interaction logs from 2022-04-16 to 2022-04-22 are used for training and the next day for testing. Through discrete feature "tab", the dataset can be divided into 14 domains. We select three of these domains for training and select the one with the smallest number of samples as the target domain.

**4.1.2 Evaluation Metrics and Implementation Details.** We apply AUC (Area Under Curve) and LogLoss (cross-entropy), which are commonly used in recommendation system as evaluation metrics. In general, a higher AUC or a lower LogLoss represents better performance. We use adam [21] for optimization, the number of source domains is 2, the number of middle layer distillation is 1, the number of epochs is 1, the learning rate is 0.001, and the batch size is 4096. We fine-tuned the temperature coefficient within {1,10,20,30,40,50}, the coefficients of different losses within {0,0.001,0.005,0.01,0.05,0.1,0.5,1,10,100}. Note that we choose DNN as the backbone by default.

**4.1.3 Baselines.** To evaluate the performance of DIIT, we compare it with several single-domain/cross-domain methods and train them in the incremental mode.

- **Single-domain:** DNN [38]: Deep Neural Network, a model of deep learning, which consists of MLPs. DCN [33]: Deep & Cross Network, a DNN-based model, it introduces a novel

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

**Table 2: Overall performance comparisons of DIIT with multiple single-domain and multi-domain models in the incremental mode on one production dataset and two public datasets. A higher AUC and a lower LogLoss represent a better performance.**

Model	Production			Taobao			KuaiRand			
	AUC	LogLoss	Impr(AUC)	AUC	LogLoss	Impr(AUC)	AUC	LogLoss	Impr(AUC)	
Single-domain	DNN	0.7455	0.0751	+0.00%	0.5969	0.1928	+0.00%	0.6648	0.6712	+0.00%
	DCN	0.7472	0.0756	+0.17%	0.5930	0.1930	-0.39%	0.6663	0.6747	+0.15%
	W&D	0.7333	0.0761	-1.22%	0.5834	0.1934	-1.35%	0.6751	0.6695	+1.03%
Cross-domain	DANN	0.7429	0.0750	-0.26%	0.5931	0.2011	-0.38%	0.6735	0.6690	+0.86%
	HAMUR	0.7454	0.0764	-0.01%	<b>0.6060</b>	0.2030	<b>+0.91%</b>	0.6807	<b>0.6632</b>	+1.59%
	CTNet	0.7496	0.0745	+0.41%	0.5991	0.1930	+0.22%	0.6821	0.6640	+1.73%
	<b>DIIT</b>	<b>0.7526</b>	<b>0.0743</b>	<b>+0.71%</b>	0.5994	<b>0.1923</b>	+0.25%	<b>0.6826</b>	0.6682	<b>+1.78%</b>

**Table 3: Comparative experiment results on reference efficiency between CTNet and DIIT on the production dataset, The best results are shown in bold.**

Model	Time-consuming(s)	RelaImpr
CTNet	191	+0.00%
<b>DIIT</b>	<b>159</b>	<b>-16.75%</b>

cross-network that can learn certain bounded-degree feature interactions more effectively without significantly increasing the complexity. W&D [8]: Wide & Deep Network, a model that combines the benefits of memorization and generalization by jointly training wide linear models and deep neural networks.

- **Cross-domain:** DANN [13]: Domain-Adversarial Neural Network, a model that uses adversarial learning to align the representation distributions between the source domain and the target domain. HAMUR [22]: a model that consists of a domain-specific adapter and a domain-shared hypernetwork, and can be seamlessly integrated with various existing backbones as a plug-and-play component. CTNet [24]: Continual Transfer Network, a model that transfers knowledge from a time-evolving source domain to a time-evolving target domain.

## 4.2 Overall Performance (RQ1)

In this section, we conduct extensive experiments to demonstrate the effectiveness and efficiency of DIIT. In addition to AUC and LogLoss, we also record the improvement of AUC.

**4.2.1 effectiveness.** To verify the effectiveness of DIIT in the industrial RS environment, we compare it with various advanced single-domain and cross-domain methods. The experimental results are shown in Table 2, and our observations are as follows.

- DIIT’s performance is advantageous in most cases, especially when compared with single-domain methods, which demonstrates the importance of transferring domain-invariant information across domains. It is worth noting that HAUAMR performs well on the Taobao dataset, its AUC is significantly

higher than various state-of-the-art models including DIIT. However, HAMUR is based on an ideal static assumption that users’ interests do not change much in a short period, which is impractical in the industrial RS environment.

- We particularly compare DIIT with CTNet, and find that CTNet’s performance is slightly worse than DIIT in most cases. The reasons for this are: 1) through two extractors, DIIT can extract domain-invariant information more efficiently. 2) CTNet is designed for the case where the source domain number is 1, which limits its ability to obtain more information across domains.

**4.2.2 efficiency.** To verify the efficiency of DIIT in the industrial RS environment, we compared it with KEEP [40] and CTNet [24], which have similar research directions to this paper. Among them, KEEP is a two-stage framework which need additional computation and storage resources. In contrast, DIIT directly extracts and transfers the domain-invariant information from multiple source domain models that train in the incremental mode independently.

CTNet is a more lightweight method that also does not need additional computation and storage resources. However, it needs to feed samples to both the source and the target domain model, whether in the training or inference phases. In contrast, DIIT only needs to keep the target domain model in the inference phase. As shown in Table 3, on the sampling test dataset in the period  $t$ , DIIT’s inference time-consuming is reduced by about 16.75%. As mentioned above, CTNet only considers the situation of one source domain, while DIIT can be easily applied to the situation of multiple source domains, and as the number of source domains increases, the gap of inference time-consuming will become larger.

## 4.3 Compatibility Experiment (RQ2)

As a pluggable component, it is necessary to verify the validity of DIIT with different backbones. In this section, we choose DNN, DCN, and W&D as the backbone respectively. The experimental results are shown in Table 4. Similar to Table 2, we also bold the best results and omit the improvement of LogLoss while marking the AUC improvement. We can found that no matter which network is used as the backbone, the proposed DIIT can bring improvements. It proves that DIIT can not only well help the target domain model

**Table 4: Compatibility experiment results of DIIT with three different models as the backbone on the production dataset**

Method	Production			Taobao			KuaiRand		
	AUC	LogLoss	Impr(AUC)	AUC	LogLoss	Impr(AUC)	AUC	LogLoss	Impr(AUC)
DNN	0.7455	0.0751	+0.00%	0.5969	0.1928	+0.00%	0.6648	0.6712	+0.00%
DNN+DIIT	0.7526	0.0743	+0.71%	0.5994	0.1923	+0.25%	0.6826	0.6682	+1.78%
DCN	0.7472	0.0756	+0.00%	0.5930	0.1930	+0.00%	0.6663	0.6747	+0.00%
DCN+DIIT	0.7516	0.0743	+0.44%	0.5993	0.1926	+0.63%	0.6699	0.6743	+0.36%
W&D	0.7333	0.0761	+0.00%	0.5834	0.1934	+0.00%	0.6751	0.6695	+0.00%
W&D+DIIT	0.7376	0.0758	+0.43%	0.5866	0.1931	+0.32%	0.6755	0.6697	+0.04%

**Table 5: Ablation experiment results of DIIT with DNN as the backbone on the production dataset, The best results are shown in bold.**

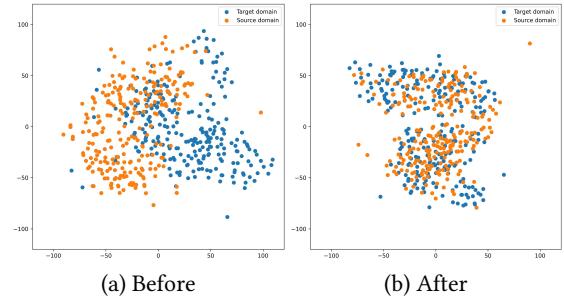
Model	AUC	Impr
Base (DNN)	0.7455	+0.00%
DIIT (Only A)	0.7508	+0.53%
DIIT (Only C)	0.7500	+0.45%
DIIT (w/o Gating)	0.7521	+0.66%
DIIT (w/o Adversarial)	0.7516	+0.61%
DIIT (w/o Middle)	0.7510	+0.55%
DIIT (w/o Logit)	0.7517	+0.62%
<b>DIIT</b>	<b>0.7526</b>	<b>+0.71%</b>

obtain the domain-invariant information across domains, but also has good practicability.

#### 4.4 Ablation Experiment (RQ3)

To verify the effectiveness of each module of DIIT, we design extensive ablation experiments by removing different modules. Specifically, we considered the following cases: 1) Use only one domain as the source domain (Only A or Only C); 2) Remove the domain-invariant information extractor at the domain level and use summation to aggregate multiple source domain model representations (w/o Gating); 3) Remove the domain-invariant information extractor at the representation level (w/o Adversarial); 4) Use only the middle distillation or the logit layer distillation for KD (w/o Middle & w/o Logit). Note that we also recorded the experimental results of DNN (Base) for intuitive comparison. The experimental results are shown in Table 3, best results are shown in bold, and in addition to recording AUC, we also marked the improvement of AUC. The improvement trend of LogLoss is similar to AUC, due to limited space, we omit it in Table 5. Our observations are as follows:

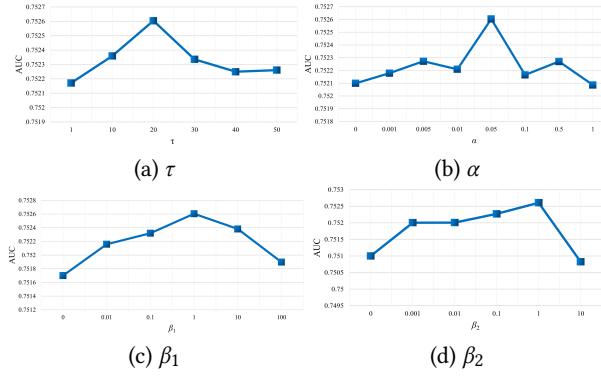
- Comparing DIIT, DIIT (Only A) and DIIT (Only C), we can find that using more source domains achieves better results, which shows that different source domains can complement each other, thereby providing richer information to the target domain model.
- Comparing DIIT and DIIT (w/o Gating), we can find that using the target domain model to guide the aggregation of

**Figure 3: The t-SNE visualization of representations before and after the domain-invariant information extractors on the production dataset.**

multiple source domain models through a gating network achieves better results, which proves the necessity of the domain-invariant information extractor at the domain level.

- Comparing DIIT and DIIT (w/o Adversarial), we can find that by using adversarial learning to align the representation distributions output by the source domain models and the target domain model, the domain-invariant information in the source domain is transferred to the target domain model in a fine-grained manner, and effectively improve the effectiveness of the target domain model.
- Comparing DIIT, DIIT (w/o Middle) and DIIT (w/o Logit), we can find that distilling information from different spots of the source domain models to the target domain model achieves better results, which proves the necessity of the multi-spot KD.
- Finally, no matter which version of DIIT is compared with Base, there is a significant improvement, which proves the importance of applying the CDR methods in the industrial RS environment.

**Visualization.** To provide a more comprehensive insight of DIIT, we visualize representations before and after the domain-invariant information extractors by t-SNE [10]. As shown in Figure 3, representations of the source and target domains become more inseparable, indicating that their distributions are aligned while domain-invariant information is extracted.



**Figure 4:** Hyper-parameter experiment results of DIIT with DNN as the backbone on the production dataset.

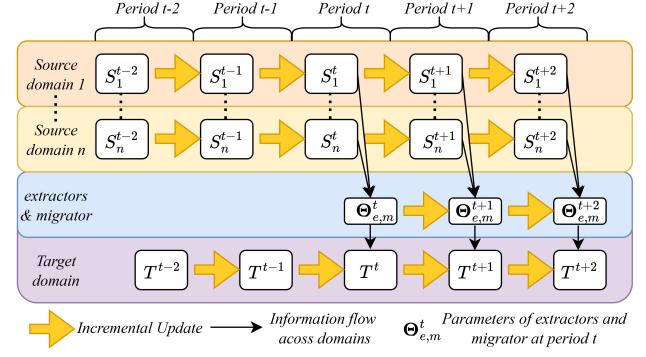
#### 4.5 Hyper-parameter Experiment (RQ3)

In this section, we further design hyper-parameter experiments to observe the impact of different modules in DIIT. As mentioned before, DIIT involves multiple hyper-parameters, most of which have been introduced in previous works. Therefore, we focus on the coefficient  $\tau$  of the distillation temperature, the coefficient  $\alpha$  of the adversarial loss  $L_{adv2}$ , the coefficient  $\beta_1$  of the middle layer distillation loss  $L_{MSE}$ , and the coefficient  $\beta_2$  of the logit layer distillation loss  $L_{KL}$ . The experimental results are shown in Figure 4, and our observations are as follows:

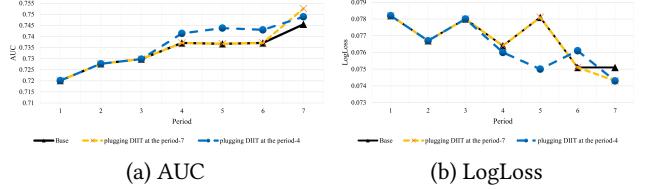
- $\tau$ : The temperature coefficient indicates how much the model pays attention to negative labels during the distillation. We find that a reasonable temperature can not only maximize the extraction of domain-invariant information from the source domain models, but also inhibit the transfer of source domains' domain-specific information.
- $\alpha, \beta_1$  and  $\beta_2$ : These coefficients represent the importance of different losses in DIIT respectively. We find that they have their own best performance ranges. By balancing them, the effectiveness of DIIT can be greatly improved. It is worth noting that when the coefficient of  $L_{adv2}$  is 0, the remaining losses is actually used as the other side of the adversarial network to align the source domains and the target domain. However, by observing Figure 4 and DIIT (w/o Adversarial) in Table 5 simultaneously, we can find that both  $L_{adv1}$  and  $L_{adv2}$  have positive effects, and retaining both is the best choice.

#### 4.6 Exploratory Experiment (RQ4)

As shown in Figure 5, we further explored more situations that may be faced in the industrial RS environment. As mentioned above, we consider a cross-domain recommendation task that contains multiple domains while each domain maintains its own model. In addition, we are curious about the impact of plugging the proposed DIIT in different periods under incremental learning (e.g. plugging DIIT in the period  $t$  as shown in Figure 5). Therefore, we conducted extensive experiments to compare the effects of base and DIIT that is plugged in period 4 and period 7 of a 7-period incremental training. The experimental results are shown in Figure 6, we find



**Figure 5:** An illustration of how DIIT works in the industrial RS environment.



**Figure 6:** Exploratory experiment results of DIIT with DNN as the backbone on the production dataset.

that plugging DIIT into the base model in different periods can bring positive effects in most cases. Moreover, comparing the curves of period 4 and period 7, it can be found that earlier plugging does not mean a better result. This may be caused by too much domain-invariant information from previous source domains flooding the domain-specific information of the latest target domain, which indicates that we can expect to observe the benefits after plugging DIIT in a short time.

## 5 CONCLUSION

In this paper, we propose DIIT, an end-to-end domain-invariant information transfer method for industrial cross-domain recommendation. We first simulated the industrial recommendation systems (RS) environment, where multiple domains maintain respective models and train them in the incremental mode. Next, in order to improve the effectiveness and efficiency of cross-domain recommendation in the industrial RS environment, we design two extractors to extract domain-invariant information at the domain level and the representation level respectively, and then design a migrator to transfer them to the target domain model. DIIT is plug-and-play and can be integrated with various methods. We further conduct extensive experiments on three datasets of different magnitudes, including one production dataset and two public datasets, to demonstrate the effectiveness and efficiency of DIIT. Future work will focus on the application of the cross-domain recommendation method in more complex environments, such as when the number of source or target domains is dynamically changing.

## REFERENCES

- [1] Jordan T. Ash and Ryan P. Adams. 2020. On warm-starting neural network training. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) (*NIPS* '20). Curran Associates Inc., Red Hook, NY, USA, Article 327, 11 pages.
- [2] Jiangxia Cao, Shaoshuai Li, Bowen Yu, Xiaobo Guo, Tingwen Liu, and Bin Wang. 2023. Towards Universal Cross-Domain Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (Singapore, Singapore) (*WSDM* '23). Association for Computing Machinery, New York, NY, USA, 78–86. <https://doi.org/10.1145/3539597.3570366>
- [3] Jiangxia Cao, Jiawei Sheng, Xin Cong, Tingwen Liu, and Bin Wang. 2022. Cross-Domain Recommendation to Cold-Start Users via Variational Information Bottleneck. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 2209–2223. <https://doi.org/10.1109/ICDE53745.2022.00211>
- [4] Chaochao Chen, Huiwen Wu, Jiajie Su, Lingjuan Lyu, Xiaolin Zheng, and Li Wang. 2022. Differential Private Knowledge Transfer for Privacy-Preserving Cross-Domain Recommendation. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) (*WWW* '22). Association for Computing Machinery, New York, NY, USA, 1455–1465. <https://doi.org/10.1145/3485447.3512192>
- [5] Gang Chen, Jiawei Chen, Fulí Feng, Sheng Zhou, and Xiangnan He. 2023. Unbiased Knowledge Distillation for Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (Singapore, Singapore) (*WSDM* '23). Association for Computing Machinery, New York, NY, USA, 976–984. <https://doi.org/10.1145/3539597.3570477>
- [6] Xu Chen, Zida Cheng, Jiangchao Yao, Chen Ju, Weilin Huang, Jinsong Lan, Xiaoyi Zeng, and Shuai Xiao. 2024. Enhancing Cross-Domain Click-Through Rate Prediction via Explicit Feature Augmentation. In *Companion Proceedings of the ACM on Web Conference 2024* (Singapore, Singapore) (*WWW* '24). Association for Computing Machinery, New York, NY, USA, 423–432. <https://doi.org/10.1145/3589335.3648341>
- [7] 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)*. 127–135. <https://doi.org/10.1109/ICDMW5131.2020.000027>
- [8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (Boston, MA, USA) (*DLRS 2016*). Association for Computing Machinery, New York, NY, USA, 7–10. <https://doi.org/10.1145/2988450.2988454>
- [9] Aveen Dayal, Vimal K B, Linga Reddy Cenkeramaddi, C Mohan, Abhinav Kumar, and Vineeth N Balasubramanian. 2023. MADG: Margin-based Adversarial Learning for Domain Generalization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) (*NIPS* '23). Curran Associates Inc., Red Hook, NY, USA, Article 2572, 15 pages.
- [10] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 2579–2605. <http://jmlr.org/papers/v9/vandermaaten08a.html>
- [11] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. 2017. CAN: Creative Adversarial Networks, Generating “Art” by Learning About Styles and Deviating from Style Norms. *ArXiv* abs/1706.07068. <https://api.semanticscholar.org/CorpusID:24986117>
- [12] Chunjing Gan, Bo Huang, Binbin Hu, Jian Ma, Zhiqiang Zhang, Jun Zhou, Guannan Zhang, and Wenliang Zhong. 2024. PEACE: Prototype LEarning Augmented transferable framework for Cross-domain rEcommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (Merida, Mexico) (*WSDM* '24). Association for Computing Machinery, New York, NY, USA, 228–237. <https://doi.org/10.1145/3616855.3635781>
- [13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* 17, 1, 2096–2030.
- [14] 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* (Atlanta, GA, USA) (*CIKM* '22). Association for Computing Machinery, New York, NY, USA, 3953–3957. <https://doi.org/10.1145/3511808.3557624>
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11, 139–144. <https://doi.org/10.1145/3422622>
- [16] 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* (Virtual Event, Singapore) (*KDD* '21). Association for Computing Machinery, New York, NY, USA, 2964–2973. <https://doi.org/10.1145/3447548>
- [17] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) (*SIGIR* '18). Association for Computing Machinery, New York, NY, USA, 355–364. <https://doi.org/10.1145/3209978.3209981>
- [18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *ArXiv* abs/1503.02531. <https://api.semanticscholar.org/CorpusID:7200347>
- [19] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. 2011. Adversarial machine learning. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence* (Chicago, Illinois, USA) (*AISec* '11). Association for Computing Machinery, New York, NY, USA, 43–58. <https://doi.org/10.1145/2046684.2046692>
- [20] Petros Katsikaris, Nikiforos Mandilaras, Dimitrios Mallis, Vassilis Pitsikalis, Stavros Theodorakis, and Gil Chamid. 2022. An Incremental Learning framework for Large-scale CTR Prediction. In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) (*RecSys* '22). Association for Computing Machinery, New York, NY, USA, 490–493. <https://doi.org/10.1145/3523227.3547390>
- [21] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980. <https://api.semanticscholar.org/CorpusID:6628106>
- [22] Xiaopeng Li, Fan Yan, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. HAMUR: Hyper Adapter for Multi-Domain Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) (*CIKM* '23). Association for Computing Machinery, New York, NY, USA, 1268–1277. <https://doi.org/10.1145/3583780.3615137>
- [23] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. 2023. Curriculum Temperature for Knowledge Distillation. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence (AAAI'23/IAAI'23/EAAI'23)*. AAAI Press, Article 167, 9 pages. <https://doi.org/10.1609/aaai.v37i2.25236>
- [24] Lixin Liu, Yanling Wang, Tianming Wang, Dong Guan, Jiawei Wu, Jingxu Chen, Rong Xiao, Wenxiang Zhu, and Fei Fang. 2023. Continual Transfer Learning for Cross-Domain Click-Through Rate Prediction at Taobao. In *Companion Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) (*WWW* '23 Companion). Association for Computing Machinery, New York, NY, USA, 346–350. <https://doi.org/10.1145/3543873.3584625>
- [25] Weiming Liu, Xiaolin Zheng, Jiajie Su, Mengling Hu, Yanchao Tan, and Chaochao Chen. 2022. Exploiting Variational Domain-Invariant User Embedding for Partially Overlapped Cross Domain Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) (*SIGIR* '22). Association for Computing Machinery, New York, NY, USA, 312–321. <https://doi.org/10.1145/3477495.3531975>
- [26] Utkarsh Ojha, Yuheng Li, Anirudh Sundara Rajan, Yingyu Liang, and Yong Jae Lee. 2023. What Knowledge Gets Distilled in Knowledge Distillation?. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) (*NIPS* '23). Curran Associates Inc., Red Hook, NY, USA, Article 487, 12 pages.
- [27] 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* (Virtual Event, Ireland) (*CIKM* '20). Association for Computing Machinery, New York, NY, USA, 2669–2676. <https://doi.org/10.1145/3340531.3412728>
- [28] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. FitNets: Hints for Thin Deep Nets. *CoRR* abs/1412.6550. <https://api.semanticscholar.org/CorpusID:2723173>
- [29] Jie Song, Ying Chen, Jingwen Ye, and Mingli Song. 2022. Spot-Adaptive Knowledge Distillation. *Trans. Img. Proc.* 31, 3359–3370. <https://doi.org/10.1109/TIP.2022.3170728>
- [30] Hongzu Su, Yifei Zhang, Xuejiao Yang, Hua Hu, Shuangyang Wang, and Jingjing Li. 2022. Cross-domain Recommendation via Adversarial Adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) (*CIKM* '22). Association for Computing Machinery, New York, NY, USA, 1808–1817. <https://doi.org/10.1145/3511808.3557277>
- [31] Juntao Tan, Shelby Heinecke, Zhiwei Liu, Yongjun Chen, Yongfeng Zhang, and Huan Wang. 2024. Towards More Robust and Accurate Sequential Recommendation with Cascade-guided Adversarial Training. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. 743–751.
- [32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive representation distillation. In *The International Conference on Learning Representations (ICLR 2020)*. 1–19.
- [33] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17* (Halifax, NS, Canada) (*ADKDD'17*). Association for Computing Machinery, New York, NY, USA, Article

- 12, 7 pages. <https://doi.org/10.1145/3124749.3124754>
- [34] Yichao Wang, Hufeng Guo, Ruiming Tang, Zhirong Liu, and Xiuqiang He. 2020. A Practical Incremental Method to Train Deep CTR Models. *ArXiv abs/2009.02147*. <https://api.semanticscholar.org/CorpusID:221507673>
- [35] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. 2023. A Survey on Cross-domain Recommendation: Taxonomies, Methods, and Future Directions. *ACM Trans. Inf. Syst.* 41, 2. <https://doi.org/10.1145/3548455>
- [36] Qiuhan Zeng, Changjian Shui, Long-Kai Huang, Peng Liu, Xi Chen, Charles X. Ling, and Boya Wang. 2024. Latent Trajectory Learning for Limited Times-tamps under Distribution Shift over Time. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*. <https://openreview.net/forum?id=btMMNT7IdW>
- [37] 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* (Merida, Mexico) (WSDM '24). Association for Computing Machinery, New York, NY, USA, 939–948. <https://doi.org/10.1145/3616855.3635828>
- [38] Weinan Zhang, Tiamming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *Advances in Information Retrieval*. Springer International Publishing, Cham, 45–57.
- [39] Wei Zhang, Pengye Zhang, Bo Zhang, Xingxing Wang, and Dong Wang. 2023. A Collaborative Transfer Learning Framework for Cross-domain Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) (KDD '23). Association for Computing Machinery, New York, NY, USA, 5576–5585. <https://doi.org/10.1145/3580305.3599758>
- [40] Yujing Zhang, Zhangming Chan, Shuhao Xu, Weijie Bian, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. KEEP: An Industrial Pre-Training Framework for Online Recommendation via Knowledge Extraction and Plugging. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) (CIKM '22). Association for Computing Machinery, New York, NY, USA, 3684–3693. <https://doi.org/10.1145/3511808.3557106>
- [41] Zeyu Zhang, Heyang Gao, Hao Yang, and Xu Chen. 2023. Hierarchical Invariant Learning for Domain Generalization Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) (KDD '23). Association for Computing Machinery, New York, NY, USA, 3470–3479. <https://doi.org/10.1145/3580305.3599377>
- [42] Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, and Jibing Gong. 2023. Beyond the Overlapping Users: Cross-Domain Recommendation via Adaptive Anchor Link Learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) (SIGIR '23). Association for Computing Machinery, New York, NY, USA, 1488–1497. <https://doi.org/10.1145/3539618.3591642>
- [43] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2023. Domain Generalization: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 4, 4396–4415. <https://doi.org/10.1109/TPAMI.2022.3195549>
- [44] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021)*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4721–4728. <https://doi.org/10.24963/ijcai.2021/639> Survey Track.
- [45] Feng Zhu, Yan Wang, Jun Zhou, Chaochao Chen, Longfei Li, and Guanfeng Liu. 2023. A Unified Framework for Cross-Domain and Cross-System Recommendations. *IEEE Transactions on Knowledge & Data Engineering* 35, 02, 1171–1184. <https://doi.org/10.1109/TKDE.2021.3104873>
- [46] Jieming Zhu, Jinyang Liu, Weiqi Li, Jincai Lai, Xiuqiang He, Liang Chen, and Zibin Zheng. 2020. Ensembled CTR Prediction via Knowledge Distillation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) (CIKM '20). Association for Computing Machinery, New York, NY, USA, 2941–2958. <https://doi.org/10.1145/3340531.3412704>

## A APPENDIX

### A.1 Training Process of DIIT

we summarize the detailed training process in the period  $t$  of DIIT in **Algorithm 1**.

**Algorithm 1:** Training Process in the period  $t$  of DIIT

---

```

input : Number of source domains  $N$ , the latest source
domain  $S_n^t$ , the latest target domain  $T^{t-1}$ , the latest
trainable parameters  $\Theta_{gate}^{t-1}, \Theta_{mapper}^{t-1}, \Theta_{dis}^{t-1}$ , the
target domain sample  $x^{Tar,t} \in D^{Tar,t}$ , the mix
domain sample  $x^{Mix,t} \in D^{Mix,t}$ .
output: The target domain model  $T^t$ .
// Warm Start
1 if plugging DIIT for the first time then
2   Initialize  $T^t$  based on  $T^{t-1}$ , Initialize Parameters  $\Theta_{gate}^t$ ,
 $\Theta_{mapper}^t, \Theta_{dis}^t$  randomly;
3 else
4   Initialize  $T^t, \Theta_{gate}^t, \Theta_{mapper}^t, \Theta_{dis}^t$  based on  $T^{t-1}, \Theta_{gate}^{t-1},$ 
 $\Theta_{mapper}^{t-1}, \Theta_{dis}^{t-1}$ ;
5 end
6 for each epoch do
7   for each mini-batch do
8     // Domain-invariant Information Extractors
9     Get  $e_{s_n}^t, Z_{s_n}^t \leftarrow S_n^t(x^{Tar,t})$ ,  $e_{adv_{s_n}}^t \leftarrow S_n^t(x^{Mix,t})$ ;
10    Get  $e_S^t, e_{adv}^t$  and  $Z_S^t$  by Eq. (1);
11    Get  $e_{adv}^t$  by Eq. (3);
12    Calculate  $L_{adv1}$  by Eq. (5);
13    Update  $\Theta_{dis}^t$  by using Adam;
14    Calculate  $L_{adv2}$  by Eq. (6);
15    Get  $e_S^t$  by Eq. (3);
16    // Domain-invariant Information Migrator
17    Calculate  $L_{KD}$  by Eq. (9);
18    // Train the Target Model
19    Calculate  $L_{CE}$  by Eq. (11);
20    // Overall Optimization
21    Calculate the total loss  $L_{total}$  by Eq. (12);
22    Update  $\Theta_{target}^t, \Theta_{gate}^t$ , and  $\Theta_{mapper}^t$  by using Adam
23 end
24 end
25 return  $T^t$ 

```

---