Task-Adversarial Adaptation for Multi-modal Recommendation

Hongzu Su
University of Electronic Science and Technology of China,
Chengdu, China
hongzus@std.uestc.edu.cn

Jingjing Li*
University of Electronic Science and Technology of China,
Chengdu, China
Institute of Electronic and Information Engineering of UESTC in Guangdong,
Dongguan, China
lijin117@yeah.net

Fengling Li
University of Technology Sydney
Sydney, Australia
fenglingli2023@gmail.com

Lei Zhu
Shandong Normal University
Jinan, China
leizhu0608@gmail.com

Ke Lu
University of Electronic Science and Technology of China
Chengdu, China
kel@uestc.edu.cn

Yang Yang
University of Electronic Science and Technology of China
Chengdu, China
yang.yang@uestc.edu.cn

ABSTRACT

An ideal multi-modal recommendation system is supposed to be timely updated with the latest modality information and interaction data because the distribution discrepancy between new data and historical data will lead to severe recommendation performance deterioration. However, upgrading a recommendation system with numerous new data consumes much time and computing resources. To mitigate this problem, we propose a Task-Adversarial Adaptation (TAA) framework, which is able to align data distributions and reduce resource consumption at the same time. This framework is specifically designed to align distributions of embedded features for different recommendation tasks between the source domain (i.e., historical data) and the target domain (i.e., new data). Technically, we design a domain feature discriminator for each task to distinguish which domain a feature comes from. By the two-player min-max game between the feature discriminator and the feature embedding network, the feature embedding network is able to align the source and target data distributions. With the ability to align source and target distributions, we are able to reduce the number of training samples by random sampling. In addition, we formulate the proposed approach as a plug-and-play module to accelerate the model training and improve the performance of mainstream multi-modal multi-task recommendation systems. We evaluate our method by predicting the Click-Through Rate (CTR) in e-commerce scenarios. Extensive experiments verify that our method is able to significantly improve prediction performance and accelerate model training on the target domain. For instance, our method is able to surpass the previous state-of-the-art method by 2.45% in terms of Area Under Curve (AUC) on AliExpress_US dataset while only utilizing one percent of the target data in training. Code: https://github.com/TL-UESTC/TAA.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Computational advertising**.

KEYWORDS

adversarial adaptation, multi-modal multi-task learning, cross-domain recommendation, model training acceleration

ACM Reference Format:
Hongzu Su, Jingjing Li, Fengling Li, Lei Zhu, Ke Lu, and Yang Yang. 2023. Task-Adversarial Adaptation for Multi-modal Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612391

1 INTRODUCTION

Online services have played a vital role in people's life with the rapid development of e-commerce. Online service providers deploy various recommendation systems to keep consumers engaged with their applications. Among these recommendation systems, multi-task learning methods [17, 18, 25] are widely studied to provide Click-Through Rate (CTR) predictions and Conversion Rate (CVR) predictions. Mainstream multi-task learning methods comprise several shared feature embedding networks and task-specific towers. With such network structures, multi-task methods are able to simultaneously exploit the shared information and task-specific information from historical user-item interaction data. To further improve the prediction performance, researchers propose to utilize both the interaction data and multi-modal information (e.g., images or videos) and construct multi-modal recommendation methods [9, 15, 29, 32, 34]. These multi-modal recommendation methods mainly add independent multi-modal embedding layers on the basis of the mainstream recommendation methods and combine the

*Corresponding author.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612391

embedded multi-modal features and interactive features into joint features. The joint features are then utilized to provide predictions for the CTR task or the CVR task.

In real-world applications, multi-modal multi-task learning methods are supposed to be trained over long periods of time using large amounts of historical multi-modal data and interaction data. As users increase or promotions (e.g., Alibaba Singles Day) begin, the recommendation systems are not able to timely learn from numerous newly logged data, which leads to severe performance degradation. To utilize the newly logged data, a natural idea is to fine-tune the prediction model on them. Unfortunately, the fine-tuning process will make the model biased toward newly logged data because of the absence of historical data. A feasible solution for this problem is learning a model with historical data and newly logged data. However, training a recommendation model on the mixed data requires vast computing resources and time because of the large amount of multi-modal data and interaction data.

Different from previous methods, we propose to leverage the newly logged data from the perspective of domain adaptation which is a widely studied transfer learning technology. We formulate the aforementioned problem as a cross-domain recommendation problem. The historical multi-modal data and interaction data are treated as the source domain data, and the newly logged multi-modal data and interaction data are treated as the target domain data. It is worth noting that the data distribution of the source domain and target domain are different because the distribution of multi-modal data and interaction data vary over time[1, 2, 21, 22, 31]. We are supposed to train a recommendation model with both the source domain data (i.e., historical data) and the sampled target domain data (i.e., newly logged data) and verify it with target domain data (i.e., newly logged data). To provide a clear comprehension, we illustrate the differences in data usage between our method and other methods in Figure 1. As illustrated in Figure 1, single-domain methods are not able to learn the data distribution of newly logged data. Fine-tuning methods mainly learn from the distribution of newly logged data. Mixed-training methods learn from the hybrid distribution of both historical data and new data. Unfortunately, these three types of methods are not able to tackle different data distributions and require vast computing resources. However, the proposed method in this paper is able to learn from the global data distribution of both historical data and new data and requires less time and computing resource consumption.

Before introducing our method, we identify two main challenges in our work as follows: (1) The distribution shift problem, which is caused by the different data distributions between the source domain data and the target domain data. (2) The vast time and computing resource consumption is caused by the numerous multi-modal data and user-item interaction data. To challenge these issues, we propose a task-adversarial adaptation framework for multi-modal multi-task learning methods. With the proposed task-adversarial adaptation framework, the target recommendation model is adversarially trained to exploit the domain-shared information. The adversarial training process aligns source and target data distributions and thus mitigates the distribution shift problem. Based on the ability to align source and target distributions, we randomly sample the target domain data to reduce the consumption of computing

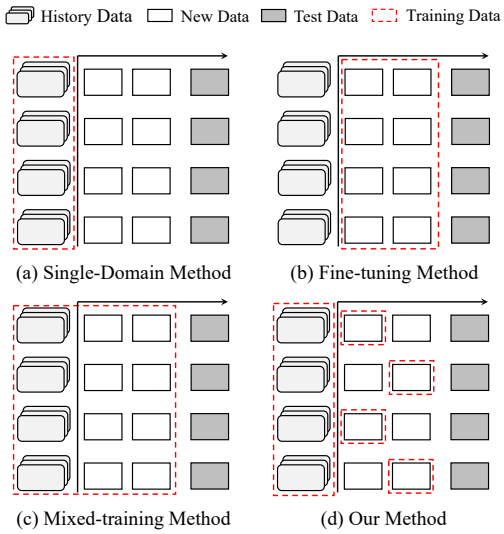


Figure 1: A comparison of training data between the proposed method and other solutions. Single-domain methods utilize historical data to train the model. Fine-tuning methods utilize all the new data to train the model. Mixed-training methods utilize both historical data and new data. The proposed method first utilizes the historical data to train a source model and then utilizes the sampled new data to conduct domain adaptation.

resources. Since the target data is independent and identically distributed (the i.i.d. assumption), the randomly sampled data is also supposed to be independent and identically distributed. Thus, the sampled data is able to represent the target distribution in the adversarial training process. Utilizing less data in training inevitably reduces the consumption of time and computing resources.

Technically, We design a feature discriminator on the joint features for each task of the multi-modal multi-task learning model in the proposed framework. The feature discriminators identify whether the joint features come from the source domain or the target domain. We construct the adversarial training paradigm with all task feature discriminators and the feature embedding network. By the two-player min-max game between feature discriminators and the feature embedding network, the feature embedding network is guided to encode domain-shared joint features for each task. We then train different task towers on the joint features to provide recommendation predictions.

To summarize, we list the contributions of this paper as follows:

(1) We identify two main challenges of utilizing numerous newly logged data for multi-modal multi-task recommendation methods. We also propose a task-adversarial adaptation framework to mitigate the distribution shift problem and reduce the consumption of time and computing resources.
(2) We formulate the proposed framework as a plug-and-play module to improve the performance of mainstream recommendation systems. In addition, we conduct extensive experiments in four scenarios of a large-scale public dataset. Experimental results show that the proposed method is able to significantly outperform vanilla mainstream multi-task learning models.

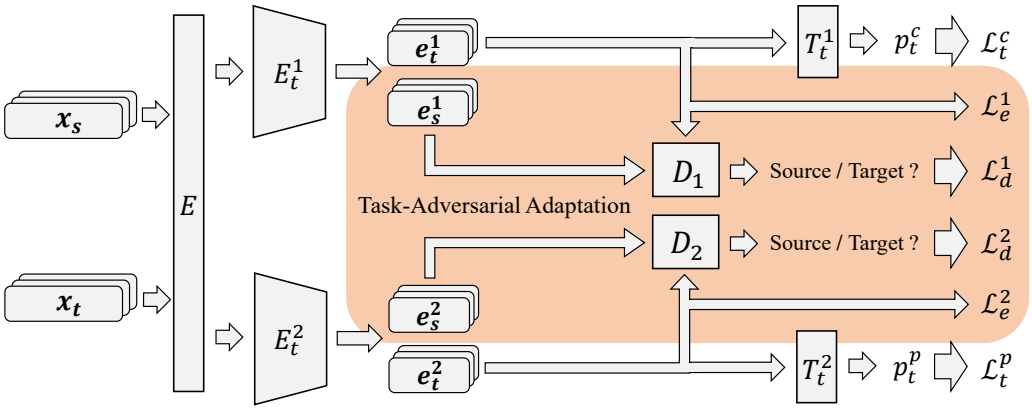


Figure 2: Illustration of the task-adversarial adaptation framework. In this framework, task feature embedding models E_t^1 and E_t^2 firstly encode source/target sample x_s/x_t into joint features e_s^1/e_t^1 and e_s^2/e_t^2. Then, feature discriminators D_1 and D_2 distinguish which domain the joint features come from. Meanwhile, task towers T_t^1 and T_t^2 leverage joint features to predict the CTR value p_t^c and CTCVR value p_t^p, respectively. The notation E denotes the shared feature embedding layer for E_t^1 and E_t^2. Notations $\mathcal{L}_t^c$, $\mathcal{L}_t^p$, $\mathcal{L}_d^1$, $\mathcal{L}_d^2$, $\mathcal{L}_e^1$ and $\mathcal{L}_e^2$ denote the loss function of CTR prediction task, CTCVR prediction task, discriminator D_1, discriminator D_2, CTR feature embedding model and CTCVR feature embedding model, respectively.

2 RELATED WORK

Multi-modal recommendation methods utilize multi-modality information to improve the recommendation performance. Among them, BM3 [34] is a representative method that utilizes visual information and textual information to match user-item pairs. Different from the matching task, we aim to provide CTR and CVR predictions with our recommendation system. Previous researches such as HyperCTR [9], FIM-IM [32], MARN [15] and MMCTR [29] provide some feasible solutions. HyperCTR [9] first learns joint user-item features with a hypergraph neural network and then predicts CTR based on the joint user-item features. FIM-IM [32] adds a local impression modeling and a global impression modeling on the vanilla FIM [28] to provide CTR prediction. MARN [15] utilizes the adversarial learning strategy to exploit the modality-specific and modality-invariant information. MMCTR [29] is a multi-task method that learns joint features from category features, long-term preference features, video features, and audio features.

Multi-task learning [3] is a technology that jointly optimizes multiple related tasks to improve the performance of each task. Recently, various solutions have been proposed to improve the recommendation performance with the multi-task learning paradigm. Among them, Shared-Bottom [17] is a method that deploys two task towers upon the same feature embedding network. OMoE [17] and MMoE [17] consist of several feature embedding experts and two task towers. ESMM [18] is proposed to handle the sample selection bias problem by utilizing the entire space data. PLE [25] is proposed to tackle the negative effects of jointly optimizing different tasks. AITM [30] introduces an adaptive information transfer module for multi-step conversions. MetaHeac [36] is a two-stage framework that contains an offline task-driven gate and an online task-driven gate. CrossDistil [33] is proposed to transfer the task-specific information across tasks. Researchers design auxiliary tasks for each main task to exploit cross-task information. AESM^2 [37] is a method that combines the multi-task learning paradigm and multi-scenario learning paradigm.

Domain adaptation is a transfer learning technology that trains a model in the source domain and transfers it to the target domain [13, 16, 24]. Early domain adaptation methods mainly focus on minimizing the source and target distribution with a specific metric criterion. For instance, Maximum Mean Discrepancy (MMD) [7] is a widely used metric that measures the distribution discrepancy in the reproducing kernel Hilbert space. The Maximum Density Divergence (MDD) [12] is a metric that simultaneously minimizes the inter-domain divergence and maximizes the intra-class density. The Kullback-Leibler (KL) divergence is also widely adopted to measure the distribution discrepancy. Recently, most of the domain adaptation methods are adversarial learning-based. Among them, DANN [5] and ADDA [26] are GAN-based [6] methods that deploy domain classifiers or domain discriminators to distinguish the domain of an input. CyCADA [10] is a CycleGAN-based [35] method that transfers information in both generative space and latent space. CDAN [16] contains two specifically designed conditioning strategies to better align the source and target data distribution.

3 PROPOSED METHOD

3.1 Problem Formulation

As aforementioned, we formulate the challenge of utilizing newly logged data as the cross-domain recommendation problem. In this setting, we are supposed to train the recommendation model with sufficient source domain data and limited target domain data. In our work, we treat the historical multi-modal data and interaction data as the source domain $\mathcal{D}_s$ and treat the newly logged multi-modal data and interaction data as the target domain $\mathcal{D}_t$. For clarity, we describe the source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$ with corresponding feature set $\mathcal{X}$ and label set $\mathcal{Y}$. The source domain is formulated as $\mathcal{D}_s = \{x_s, y_s^c, y_s^p\}$, where $x_s \in \mathcal{X}_s$ denotes the source domain feature, $y_s^c \in \mathcal{Y}_s$ and $y_s^p \in \mathcal{Y}_s$ refer to the label of click-through rate (CTR) prediction and click-through & conversion rate (CTCVR) prediction corresponding to the source feature x_s.

Similarly, the target domain is formulated as $\mathcal{D}_t = \{x_t, y_t^c, y_t^p\}$, where $x_t \in \mathcal{X}_t$ denotes the target domain feature, $y_t^c \in \mathcal{Y}_t$ and $y_t^p \in \mathcal{Y}_t$ refer to the label of CTR prediction and CTCVR prediction corresponding to the target feature x_t.

In our work, we first pre-train a mainstream multi-task learning model with sufficient source domain data. The source recommendation model first embeds the input features x_s into joint features by the feature embedding networks E_s^i, where i refers to the i-th task. Then, the task towers T_s^i provide recommendation predictions (p_s^c, p_s^p) based on the embedded joint features. Then, we initialize the target model with the well-trained source model and construct the task-adversarial structure according to Figure 2. In this structure, we deploy feature discriminator D_i for each task to distinguish which domain the joint features come from and align the source and target data distributions. With the ability of distribution alignment, we are able to utilize the randomly sampled target data rather than all the target data to train the target recommendation model. As a result, our method is able to improve recommendation performance while reducing the consumption of time and computing resources.

3.2 The Source Multi-modal Multi-task Model

As aforementioned, we are not trying to design a dedicated multi-modal multi-task learning model. Instead, we aim to design an adaptation framework that can be combined with the mainstream multi-task learning methods. Thus, we directly adopt the mainstream model as the source model to conduct our research. Given a sample (x_s, y_s^c, y_s^p), the source model predicts CTR and CTCVR according to $p_s^c = T_s^1(E_s^1(x_s))$ and $p_s^p = T_s^2(E_s^2(x_s))$, respectively. The source model is optimized by minimizing the following loss:

$$\begin{aligned}\mathcal{L}_s = \beta_c(-y_s^c \cdot \log(p_s^c) + (1 - y_s^c) \cdot \log(1 - p_s^c)) \\ + \beta_p(-y_s^p \cdot \log(p_s^p) + (1 - y_s^p) \cdot \log(1 - p_s^p)),\end{aligned} \tag{1}$$

where hyper-parameters β_c and β_p are used to balance the contribution of two prediction tasks.

In our work, the source model is pre-trained with sufficient source domain data. It is used to accelerate the task-adversarial adaptation in the following aspects: (1) Initializing the target model. The target model initialized with the source model is able to achieve good recommendation performance with less training epochs. (2) Learning the source feature distribution. The well-trained source model can be directly adopted to encode joint features that represent the source data distribution instead of learning the source distribution from scratch.

3.3 Task-Adversarial Structure

Since the source model is able to accelerate the task-adversarial adaptation, we directly initialize the target model with parameters of the source model. In our work, the target model is supposed to learn from numerous target domain data. To this end, we propose to specifically align the source data distribution and the target data distribution. We design a domain discriminator upon each feature embedding model to construct the task-adversarial structure. As illustrated in Figure 2, discriminator D_1 distinguishes source and target joint features of the CTR task, and discriminator D_2 distinguishes source and target joint features of the CTCVR task.

In our task-adversarial structure, the target model takes both source and target features as input and embeds them into task-specific features as follows:

$$\begin{aligned} e_s^1 = E_t^1(x_s), \quad e_s^2 = E_t^2(x_s), \\ e_t^1 = E_t^1(x_t), \quad e_t^2 = E_t^2(x_t). \end{aligned} \tag{2}$$

Two feature discriminators are able to perform adversarial learning based on the encoded features. Discriminator D_1 takes e_s^1 and e_t^1 as inputs and distinguishes the domain of them according to the following loss function:

$$\begin{aligned}\mathcal{L}_d^1 = \mathbb{E}_{e_t^1 \sim \mathbb{P}_t^1}[D_1(e_t^1)] - \mathbb{E}_{e_s^1 \sim \mathbb{P}_s^1}[D_1(e_s^1)] \\ + \lambda_{gp}\mathbb{E}\left[\left(\left\|\nabla_{\hat{e_t^1}} D_1(\hat{e_t^1}))\right\|_2 - 1\right)^2\right],\end{aligned} \tag{3}$$

where $\hat{e_t^1} = \mu e_s^1 + (1-\mu)e_t^1$ with $\mu \sim U(0,1)$, $\mathbb{P}_s^1$ and $\mathbb{P}_t^1$ denote the data distribution of source CTR features and target CTR features, respectively. The term in the second line refers to the calculation of gradient penalty which is used to enforce the Lipschitz constraint [8]. $\lambda_{gp} > 0$ is used to adjust the gradient penalty.

By minimizing the Eq. (3), the discriminator D_1 is able to distinguish which domain an input comes from. To confuse this discriminator, the feature embedding model E_t^1 is supposed to encode target features e_t^1 similar to the source feature e_s^1. Thus, the feature embedding model E_t^1 is guided to encode e_t^1 by the following loss:

$$\mathcal{L}_e^1 = -\mathbb{E}_{e_t^1 \sim \mathbb{P}_t^1}\left[D_1(e_t^1)\right], \tag{4}$$

where $\mathbb{P}_t^1$ refers to the data distribution of target CTR features.

Similar to discriminator D_1, discriminator D_2 takes e_s^2 and e_t^2 as inputs and distinguishes the domain of them conditioned on the following loss function:

$$\begin{aligned}\mathcal{L}_d^2 = \mathbb{E}_{e_t^2 \sim \mathbb{P}_t^2}[D_2(e_t^2)] - \mathbb{E}_{e_s^2 \sim \mathbb{P}_s^2}[D_2(e_s^2)] \\ + \lambda_{gp}\mathbb{E}\left[\left(\left\|\nabla_{\hat{e_t^2}} D_2(\hat{e_t^2}))\right\|_2 - 1\right)^2\right],\end{aligned} \tag{5}$$

where $\hat{e_t^2} = \mu e_s^2 + (1-\mu)e_t^2$ with $\mu \sim U(0,1)$, $\mathbb{P}_s^2$ and $\mathbb{P}_t^2$ denote data distribution of source CTCVR features and target CTCVR features.

The feature embedding model E_t^2 is also supposed to encode target features e_t^2 similar to the source feature e_s^2. It is optimized by minimizing the following loss function:

$$\mathcal{L}_e^2 = -\mathbb{E}_{e_t^2 \sim \mathbb{P}_t^2}\left[D_2(e_t^2)\right], \tag{6}$$

where $\mathbb{P}_t^2$ denotes the data distribution of target CTCVR features.

By optimizing the task-adversarial adaptation structure, the feature embedding models E_t^1 and E_t^2 are able to encode similar target features and source features, i.e., the data distribution of source and target domain are aligned.

3.4 Data Sampling and Task Learning

In our work, the task-adversarial structure is able to align the source and target data distribution. Since the alignment is based on the overall data distribution rather than specific data samples, we can reduce the number of samples without changing the data distribution. We propose to reduce the number of target samples by randomly sampling from the target domain. As aforementioned,

Table 1: Statistics of the dataset under different settings. Symbol # denotes the number of corresponding entries.

Datasets (Settings)	#Train impression	#Train click	#Train conversion	#Test impression	#Test click	#Test conversion
RU (source)	95,355,689	2,568,767	43,102	34,564,064	1,049,725	18,796
NL (Basic)	12,157,894	244,971	8,904	5,559,301	136,107	4,911
NL (Ours 1%)	121,087	2,442	94	5,559,301	136,107	4,911
ES (Basic)	22,326,719	574,544	12,934	9,342,708	267,511	6,162
ES (Ours 1%)	222,385	5,695	151	9,342,708	267,511	6,162
FR (Basic)	18,212,800	344,605	9,051	8,822,801	198,148	5,379
FR (Ours 1%)	182,137	3,390	79	8,822,801	198,148	5,379
US (Basic)	19,932,049	289,063	6,963	7,460,564	160,545	3,867
US (Ours 1%)	198,859	2,828	66	7,460,564	160,545	3,867

the target data is independent identically distributed. Thus, the randomly sampled data is also supposed to be independent identically distributed, and the sampled data is able to represent the target distribution. We set a random number *rand* for each target sample and select the data for training according to the following function:

$$\hat{\mathcal{D}}_t = \left\{ \begin{array}{ll} (x_t, y_t^c, y_t^p)\ select, & rand < \gamma \\ (x_t, y_t^c, y_t^p)\ drop, & rand \geq \gamma \end{array} \right\}, \tag{7}$$

where γ refers to the threshold to select a sample.

Once the target samples are selected, target feature embedding models E_t^1 and E_t^2 embed them into task features e_t^1 and e_t^2. Then, we can train target CTR task tower T_t^1 with CTR feature e_t^1 and corresponding label according to the following loss function:

$$\mathcal{L}_t^c = (-y_t^c \cdot \log(p_t^c) + (1 - y_t^c) \cdot \log(1 - p_t^c)), \tag{8}$$

where $p_t^c = T_t^1(e_t^1)$. Similarly, the target CTCVR task tower T_t^2 is trained by minimizing the following loss function:

$$\mathcal{L}_t^p = (-y_t^p \cdot \log(p_t^p) + (1 - y_t^p) \cdot \log(1 - p_t^p)), \tag{9}$$

where $p_t^p = T_t^2(e_t^2)$. At last, we optimize all the components of target model according to the following loss:

$$\mathcal{L}_t = \lambda_{e1}\mathcal{L}_e^1 + \lambda_{e2}\mathcal{L}_e^2 + \lambda_c\mathcal{L}_t^c + \lambda_p\mathcal{L}_t^p, \tag{10}$$

where λ_{e1}, λ_{e2}, λ_c and λ_p are used to balance the contribution of the four components.

4 EXPERIMENTS

In this section, we apply the proposed task-adversarial adaptation framework on five mainstream multi-task learning models and conduct extensive experiments in four scenarios of a public e-commerce dataset to study the following research questions: **RQ1:** Does the proposed method achieves better performance than the model of other experiment settings? **RQ2:** Does the proposed method reduces the consumption of time and computing resources? **RQ3:** Does the proposed method achieves better performance than other multi-task learning methods? **RQ4:** What is the effect of sampling threshold γ (in Eq.(7)) on the model performance? **RQ5:** What is the effect of different hyper-parameters on the model performance?

4.1 Datasets

We evaluate our method on the real-world dataset AliExpress [20] that is not specifically designed for the multi-modal recommendation. We chose this dataset for the following reasons: (1) There are no publicly available multi-modal datasets in the community that provide CTR and CTCVR predictions. (2) Since our method performs adversarial adaptation on the joint features of mainstream multi-task methods, the validation on common CTR and CTCVR prediction datasets is able to reveal the effect of our method. (3) The AliExpress dataset is a real-world large-scale dataset with five individual sub-datasets and over 100 million pieces of samples. Thus, this dataset is able to validate our approach adequately.

The AliExpress dataset consists of five domains with data collected from five countries: Netherlands (NL), French (FR), Spain (ES), America (US), and Russia (RU). Since the data is collected from five different countries, there is inevitable data distribution discrepancy between different domains. Thus, the aforementioned source domain can be represented by selecting one domain from the AliExpress dataset, while another domain can be chosen to represent the target domain. In our work, RU is treated as the source domain, and the other four datasets are treated as four target domains. In the training period, we randomly sample the target data according to Eq. (7). It is worth noting that the test data is not modified and consistent with the test samples in the original dataset. We detail the statistics of the dataset under different settings in Table 1.

4.2 Experimental Protocols

Evaluation Metric. In this paper, we evaluate the recommendation performance with the metric of Area Under the ROC Curve (AUC) [4] over the target domain. The AUC is able to reflect the prediction accuracy of a recommendation model and is widely used in both CTR prediction task and CTCVR prediction task [18, 33, 37]. High quantitative results in terms of AUC indicate the outstanding predictive ability of a recommendation model.

Implementation Details. As aforementioned, our approach is formulated as a plug-and-play module for the mainstream multi-task learning methods. In our work, the mainstream models are implemented exactly the same as reported in corresponding papers. We implement the domain discriminators with three fully-connected layers. We use Adam [11] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ to optimize all the models. In our work, we select all the hyper-parameters through cross validation. Hyper-parameters λ_{e1}, λ_{e2}, λ_c and λ_p are setting to 1.0. All results are reported with Pytorch 1.10.2 and trained on NVIDIA RTX 3090 GPUs. We also release a MindSpore implementation of our work.

Experiment Setting. In our work, we conduct experiments under different settings to evaluate the proposed method. The designed experiment settings are listed as follows:

Basic: We train the mainstream multi-task learning models with all the target training data and test the models with target test data. Notice that the **Basic** model is different from the single domain method in Figure 1, which is trained on the source domain and not

Table 2: The performance comparison of different models. In this table, CTR and CTCVR refer to the quantitative results of the CTR prediction task and CTCVR prediction task in terms of AUC. The best results are marked in Bold.

Methods		NL		FR		ES		US	
		CTR	CTCVR	CTR	CTCVR	CTR	CTCVR	CTR	CTCVR
Shared-Bottom [17]	Basic	72.23	85.75	71.99	85.88	72.77	88.61	70.74	86.87
	Fine-tuning	72.45	85.93	72.79	87.09	72.83	88.96	70.59	87.48
	Mixed-training	72.59	86.02	71.62	88.35	72.68	88.43	69.69	87.35
	TAA (Ours)	**74.42**	**87.12**	**74.31**	**88.92**	**74.20**	**89.41**	**73.25**	**88.74**
MMoE [17]	Basic	71.60	85.93	73.15	87.58	72.87	88.94	70.67	87.83
	Fine-tuning	72.33	86.16	72.48	87.18	72.77	89.03	70.62	88.15
	Mixed-training	72.14	86.64	72.04	88.37	72.75	88.98	69.96	87.42
	TAA (Ours)	**74.58**	**87.56**	**74.46**	**89.14**	**74.42**	**89.72**	**73.33**	**89.11**
OMoE [17]	Basic	71.69	85.16	72.13	87.29	73.01	88.50	71.17	87.22
	Fine-tuning	72.47	86.09	72.48	87.17	72.85	89.06	70.47	87.56
	Mixed-training	72.40	86.62	72.07	88.14	72.63	88.53	70.79	87.12
	TAA (Ours)	**74.38**	**87.37**	**74.28**	**88.89**	**74.16**	**89.42**	**73.11**	**88.64**
PLE [25]	Basic	72.32	85.66	72.53	87.59	72.46	88.34	70.37	87.18
	Fine-tuning	72.27	86.25	72.57	86.65	72.91	89.03	70.73	87.99
	Mixed-training	72.67	86.71	71.26	88.14	72.80	89.29	69.90	87.56
	TAA (Ours)	**74.30**	**87.08**	**74.04**	**88.65**	**74.12**	**89.01**	**72.80**	**88.11**
AITM [30]	Basic	72.45	86.47	72.55	86.71	72.71	88.53	71.34	86.86
	Fine-tuning	72.27	86.27	72.48	87.60	72.75	89.08	70.59	87.98
	Mixed-training	72.82	85.52	71.92	88.25	72.80	88.70	70.28	87.63
	TAA (Ours)	**74.12**	**86.94**	**73.87**	**88.84**	**73.90**	**89.41**	**72.82**	**88.54**

Table 3: The time consumption of different models. In this table, notation m refers to minutes.

Methods		NL	FR	ES	US
Shared-Bottom	Fine-tuning	16m	31m	35m	46m
	Ours	**6**m	**6**m	**6**m	**6**m
MMoE	Fine-tuning	18m	36m	28m	39m
	Ours	**5**m	**5**m	**6**m	**5**m
OMoE	Fine-tuning	15m	37m	46m	25m
	Ours	**3**m	**3**m	**4**m	**3**m
PLE	Fine-tuning	30m	34m	48m	33m
	Ours	**3**m	**3**m	**3**m	**4**m
AITM	Fine-tuning	17m	35m	20m	38m
	Ours	**2**m	**3**m	**3**m	**3**m

able to effectively provide predictions on the target domain. The setting of the Basic model is designed to verify that our method is able to improve the prediction performance on the target domain.

Fine-tuning: We train the mainstream multi-task learning models on source domain and fine-tune them with all the target training data. We test these fine-tuned models with target test data.

Mixed-training: We train the mainstream multi-task learning models on the mixed dataset of source and target training data. We test these models with target test data.

TAA: We first train the mainstream multi-task learning models on source domain and then learn the proposed model with source data and selected target training data. The proposed method is also tested on target test data.

4.3 Experimental Results

RQ1: The ability to outperform other experiment settings. We report the results of the basic model, the fine-tuned model, the mixed model and our method in Table 2. It is worth noting that we set the same learning rate for the fine-tuned model and our method.

According to Table 2, our method is able to significantly outperforms the basic model. From the results, our method is able to achieve the average improvement of 2.11%, 2.13%, 1.98%, 1.89% and 1.42% when compared with the basic Shared-Bottom, MMoE, OMoE, PLE and AITM on the CTR prediction task. Among these methods, our method is able to achieve the best CTR prediction performance with the MMoE method. The task-adversarial adaptation is able to surpass the basic MMoE by 2.98%, 1.31%, 1.55%, 2.66% on NL, FR, ES and US, respectively. We can also observe that our method is able to achieve the average improvement of 1.77%, 1.31%, 1.54%, 1.02% and 1.29% when compared with the basic Shared-Bottom, MMoE, OMoE, PLE and AITM on the CTCVR prediction task. The combination of our method and Shared-Bottom is able to achieve the best CTCVR prediction performance.

According to Table 2, our method is able to surpass the fine-tuned model by a wide margin. From the results, our method is able to achieve the average improvement of 1.88%, 2.14%, 1.91%, 1.70% and 1.65% when compared with the fine-tuned Shared-Bottom, MMoE,

Table 4: The CTCVR prediction performance with different data sampling thresholds. In this table, Shared refers to Shared-Bottom. A smaller threshold γ indicates fewer samples. The best results are shown in Bold.

Methods		NL	FR	ES	US
Shared + TAA	$\gamma = 0.005$	86.95	**88.99**	89.31	88.73
	$\gamma = 0.01$	87.12	88.92	89.41	88.74
	$\gamma = 0.05$	**87.19**	88.94	**89.44**	**88.75**
MMoE + TAA	$\gamma = 0.005$	87.54	89.09	89.72	**89.13**
	$\gamma = 0.01$	87.56	**89.14**	89.72	89.11
	$\gamma = 0.05$	**87.61**	**89.14**	**89.75**	88.72
OMoE + TAA	$\gamma = 0.005$	87.36	88.92	89.34	**88.64**
	$\gamma = 0.01$	**87.37**	88.89	89.42	**88.64**
	$\gamma = 0.05$	87.28	**88.98**	**89.44**	88.59
PLE + TAA	$\gamma = 0.005$	**87.11**	88.59	88.98	88.07
	$\gamma = 0.01$	87.08	**88.65**	89.01	**88.11**
	$\gamma = 0.05$	86.89	88.54	**89.13**	87.71
AITM + TAA	$\gamma = 0.005$	86.87	88.77	**89.45**	88.55
	$\gamma = 0.01$	86.94	88.84	89.41	88.54
	$\gamma = 0.05$	**87.08**	**88.90**	89.42	**88.59**

Table 5: Performance comparison with other multi-task learning models. Methods marked with † and ∗ refer to the results reported in previous work and the results achieved by our implementation, respectively. The best results are shown in Bold.

Methods	NL		FR		ES		US	
	CTR	CTCVR	CTR	CTCVR	CTR	CTCVR	CTR	CTCVR
Cross-Stitch† (2016) [19]	72.04	84.92	71.59	86.50	71.77	87.56	69.66	83.02
MMOE† (2018) [17]	71.77	84.87	71.45	86.18	72.16	87.75	69.83	84.87
HMoE† (2020) [14]	71.87	85.47	71.56	86.76	72.06	88.01	69.81	84.71
PLE† (2020) [25]	71.91	85.60	71.60	86.95	72.31	88.53	69.90	86.18
STAR† (2021) [23]	71.45	84.76	71.29	86.36	71.94	88.21	70.03	85.63
AESM^2† (2022) [37]	72.60	86.38	72.41	88.08	72.95	89.49	70.88	87.74
PLE∗	72.32	85.66	72.53	87.59	72.46	88.34	70.37	87.18
PLE + TAA (Ours)	74.30	87.08	74.04	88.65	74.12	89.01	72.80	88.11
MMoE∗	71.60	85.93	73.15	87.58	72.87	88.94	70.67	87.83
MMoE + TAA (Ours)	**74.58**	**87.56**	**74.46**	**89.14**	**74.42**	**89.72**	**73.33**	**89.11**

Table 6: The CTCVR prediction performance between task-level adaptation and non-task-level adaptation. Shared refers to Shared-Bottom. The best results are shown in Bold.

Methods		NL	FR	ES	US
Shared + TAA	non-task	86.87	88.63	89.08	88.18
	task (Ours)	**87.12**	**88.92**	**89.41**	**88.74**
MMoE + TAA	non-task	87.35	88.95	89.60	88.62
	task (Ours)	**87.56**	**89.14**	**89.72**	**89.11**
OMoE + TAA	non-task	86.96	88.57	89.18	88.13
	task (Ours)	**87.37**	**88.89**	**89.42**	**88.64**
PLE + TAA	non-task	84.88	87.24	88.77	87.67
	task (Ours)	**87.08**	**88.65**	**89.01**	**88.11**
AITM + TAA	non-task	86.51	88.19	89.06	87.75
	task (Ours)	**86.94**	**88.84**	**89.41**	**88.54**

OMoE, PLE and AITM on the CTR prediction task. According to the results of the CTCVR prediction task, our method is able to surpass the fine-tuned Shared-Bottom, MMoE, OMoE, PLE and AITM by 1.18%, 1.25%, 1.11%, 0.73% and 0.70%. We can also observe that the performance of the fine-tuned model may be worse than the basic model. For instance, the fine-tuned model achieves 70.47 in the CTR prediction task on the US dataset, yet the basic model achieves 71.17. This phenomenon is mainly due to the inability of the fine-tuned method to efficiently process different source and target data distributions. This observation verifies that the ability to align source and target data distributions is vital.

According to Table 2, our method is able to significantly outperforms the mixed-training method. From the results, our method achieves the average improvement of 2.40%, 2.47%, 2.01%, 2.16% and 1.72% when compared with the mixed-training Shared-Bottom, MMoE, OMoE, PLE and AITM on the CTR prediction task. On the CTCVR prediction task, our method surpasses the mixed-training Shared-Bottom, MMoE, OMoE, PLE and AITM by 1.01%, 1.03%, 0.98%, 0.29% and 0.91%. We can also observe that the mixed-training method may perform worse than the fine-tuned method. For instance, the fine-tuned model surpasses the mixed-training Shared-Bottom, MMoE, OMoE, PLE and AITM by 0.52%, 0.33%, 0.10%, 0.46% and 0.07% on the CTR prediction task. This phenomenon is mainly because of the distribution gap between the source data and the target data. This observation also verifies that aligning source and target data distributions is necessary.

RQ2: The ability to reduce the time consumption and computing resource consumption. To study the time consumption of different models, we conduct experiments and record the training time of the fine-tuning method and our method. It is worth noting that we record the time consumption under the condition of **the same batch size and machine**. We report the time consumption in Table 3. As aforementioned in Figure 1, the fine-tuning method and our method are all pre-trained on the historical data (i.e., the source data in our experiments). Thus, the time consumption of training the source model is not included in Table 3. According to Table 3, the average training time of the fine-tuned method is eight times larger than our method. This observation verifies that our method is able to successfully reduce the consumption of time. According to the number of training samples in Table 1, our method only leverages one percent of the target samples and inevitably reduces the consumption of computing resources. We can conclude from the experiment results that our method is able to reduce the consumption of time and computing resources.

RQ3: The ability to outperform other multi-task learning methods. We compare the performance of our method with other multi-task learning methods and report the results in Table 5. From the results, our method is able to surpass the previous state-of-the-art method by 1.98%, 2.05%, 1.47% and 2.45% in CTR prediction tasks on NL, FR, ES and US, respectively. According to the results of CTCVR prediction task, our method is able to achieve the improvement of 1.18%, 1.06%, 0.23% and 1.37% on NL, FR, ES and US, respectively. This observation verifies that our method is able to outperform previous methods on all of the tested datasets. The results in Table 5 also reveal that aligning source and target data distributions is able to improve the recommendation performance.

4.4 Model Analysis

RQ4: Effect of the data sampling threshold γ. According to Eq. (7), the target data is selected with threshold γ. In our work, smaller sampling threshold indicates selecting fewer samples from the target domain. To study the effect of the number of samples on model performance, we conduct experiments with different sampling threshold and report the experimental results in Table 4. From the results, our method is prone to achieve better performance with greater sampling threshold, yet the improvement is very slight. For instance, the performance with threshold $\gamma = 0.05$ only achieves the improvement of 0.07%, 0.02%, 0.03% and 0.01% on NL, FR, ES and US when compared with the performance with threshold $\gamma = 0.01$. The results in Table 4 verifies that our method is relatively not sensitive to the number of target data because our method is able to align data distributions with few data samples.

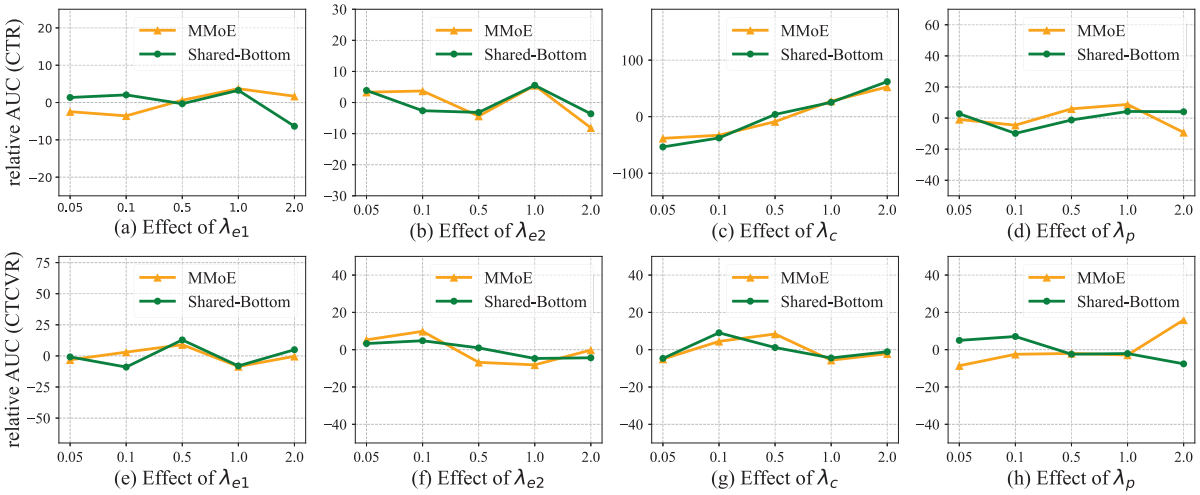
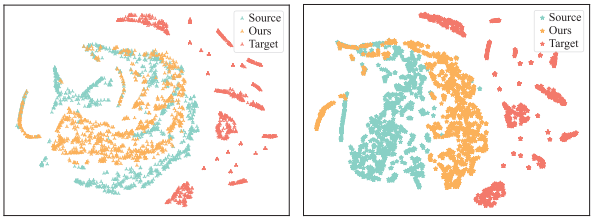


Figure 3: Parameter sensitivity analysis. The results are reported with metric *relative AUC*.



(a) Results of the CTR task (b) Results of the CTCVR task

Figure 4: Visualization results of the joint features.

RQ5: Effect of hyper-parameters λ_{e1}, λ_{e2}, λ_c and λ_p. To clearly demonstrate the effect of different hyper-parameters, we replace the results of AUC with relative AUC as: $relative\ AUC = (AUC - mean) \times 1000$, where *mean* is the mean value of all the real AUC results. Hyper-parameters λ_{e1} and λ_{e2} control the contribution of Eq. (4) and Eq. (6). According to Figure 3 (a), 3 (b), 3 (e), 3 (f), our method is relatively not sensitive to λ_{e1} and λ_{e2}. The results of both CTR prediction and CTCVR prediction fluctuate as λ_{e1} and λ_{e2} increase. The performance of the CTR prediction task tends to decline as λ_{e1} and λ_{e2} increase. The performance of the CTCVR prediction task first grows up and then shows decline slowly as λ_{e1} and λ_{e2} increase. Hyper-parameters λ_c and λ_p control the contribution of the CTR prediction task and CTCVR prediction task. According to Figure 3 (c) and 3 (g), the performance of the CTR prediction increases steadily as the λ_c grows up, yet the performance of the CTCVR prediction first grows up and then shows a slight decline as the λ_c increases. According to Figure 3 (d) and 3 (h), our method is relatively not sensitive to λ_p. The performance of the CTR task first grows up slowly and then tends to decline as λ_p increases.

Effect of the task-level adaptation. In this work, our method separately aligns source and target features for each task. To verify the effect of the task-level adaptation, we design experiments of non-task-level adaptation by directly concatenating the features of different tasks and report the performance comparison in Table 6. From the results, our method is able to surpass the non-task-level Share-Bottom, MMoE, OMoE, PLE and AITM by 0.36%, 0.25%, 0.37%, 1.07% and 0.55% on the CTCVR prediction task. This observation verifies that the task-level adaptation is able to achieve better recommendation performance than the non-task-level adaptation.

The ability to align distributions. We apply the proposed method on MMoE and visualize the joint features for each task by t-SNE [27]. We report the visualization results in Figure 4. From the results of CTR task, joint features embedded by our method locate closer to source domain features and show a clear boundary with original target domain features. The visualization results of CTCVR task show the same observation as the results of CTR task. These observations verify that joint features encoded by our method share similar data distributions with source domain features. We can conclude that our method is able to successfully align the distribution of encoded target domain features with source domain distribution. With the ability to align two data distributions, we are able to improve the recommendation performance on the target domain with abundant source domain data.

5 CONCLUSION

In this work, we propose a task-adversarial adaptation framework for the mainstream multi-task learning methods to tackle the challenge of utilizing newly logged data. In this framework, we design a feature discriminator for each task in the multi-task learning model. By adversarially training the feature discriminators and the feature embedding networks, our method is able to align the data distribution of historical data and newly logged data. With the ability of distribution alignment, we are able to reduce the consumption of time and computing resources by randomly sampling the newly logged data in the training process. We apply our framework to mainstream multi-task learning methods and conduct extensive experiments. Experimental results verify that our method is able to simultaneously improve the recommendation performance and reduce the time and computing resources consumption.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62250061, 62176042, and in part by Sichuan Science and Technology Program under Grant 2023NSFSC0483, and in part Sponsored by CAAI-Huawei MindSpore Open Fund, and in part by Guangdong Basic and Applied Basic Research Foundation (No. 2021B1515140013).

REFERENCES

[1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proceedings of the 3rd international web science conference*. 1–8.

[2] Pedro G Campos, Fernando Díez, and Iván Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction* 24 (2014), 67–119.

[3] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[4] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[7] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13, 1 (2012), 723–773.

[8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems* 30 (2017).

[9] Li He, Hongxu Chen, Dingxian Wang, Shoaib Jameel, Philip Yu, and Guandong Xu. 2021. Click-through rate prediction with multi-modal hypergraphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 690–699.

[10] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*. Pmlr, 1989–1998.

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[12] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. 2020. Maximum density divergence for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence* 43, 11 (2020), 3918–3930.

[13] Jingjing Li, Mengmeng Jing, Hongzu Su, Ke Lu, Lei Zhu, and Heng Tao Shen. 2021. Faster domain adaptation networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2021), 5770–5783.

[14] 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.

[15] Xiang Li, Chao Wang, Jiwei Tan, Xiaoyi Zeng, Dan Ou, Dan Ou, and Bo Zheng. 2020. Adversarial multimodal representation learning for click-through rate prediction. In *Proceedings of The Web Conference 2020*. 827–836.

[16] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. *Advances in neural information processing systems* 31 (2018).

[17] 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.

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[19] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3994–4003.

[20] 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 28th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19- 23,2019*. ACM, New York,NY,USA.

[21] Dimitrios Rafailidis and Alexandros Nanopoulos. 2015. Modeling users preference dynamics and side information in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46, 6 (2015), 782–792.

[22] Diego Sánchez-Moreno, Yong Zheng, and María N Moreno-García. 2020. Time-aware music recommender systems: Modeling the evolution of implicit user preferences and user listening habits in a collaborative filtering approach. *Applied Sciences* 10, 15 (2020), 5324.

[23] 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.

[24] Hongzu Su, Yifei Zhang, Xuejiao Yang, Hua Hua, Shuangyang Wang, and Jingjing Li. 2022. Cross-domain Recommendation via Adversarial Adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1808–1817.

[25] 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.

[26] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7167–7176.

[27] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[28] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 836–845.

[29] Jinshan Wang, Qianfang Xu, Qiang Wang, Zhongjian Lyu, Jiaxin Chen, and Wenchao Xu. 2019. MMCTR: A Multi-Task Model for Short Video CTR Prediction with Multi-Modal Video Content Features. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 679–682.

[30] Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3745–3755.

[31] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*. SIAM, 211–222.

[32] Jiahao Xun, Shengyu Zhang, Zhou Zhao, Jieming Zhu, Qi Zhang, Jingjie Li, Xiuqiang He, Xiaofei He, Tat-Seng Chua, and Fei Wu. 2021. Why do we click: visual impression-aware news recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3881–3890.

[33] Chenxiao Yang, Junwei Pan, Xiaofeng Gao, Tingyu Jiang, Dapeng Liu, and Guihai Chen. 2022. Cross-Task Knowledge Distillation in Multi-Task Recommendation. *arXiv preprint arXiv:2202.09852* (2022).

[34] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2022. Bootstrap latent representations for multi-modal recommendation. *arXiv preprint arXiv:2207.05969* (2022).

[35] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

[36] Yongchun Zhu, Yudan Liu, Ruobing Xie, Fuzhen Zhuang, Xiaobo Hao, Kaikai Ge, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Expand Audience via Meta Hybrid Experts and Critics for Recommendation and Advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4005–4013.

[37] Xinyu Zou, Zhi Hu, Yiming Zhao, Xuchu Ding, Zhongyi Liu, Chenliang Li, and Aixin Sun. 2022. Automatic Expert Selection for Multi-Scenario and Multi-Task Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1535–1544.