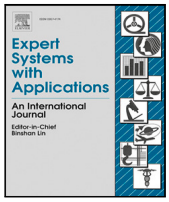
Expert Systems With Applications 243 (2024) 122880

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



DADIN: Domain Adversarial Deep Interest Network for cross domain recommender systems

Menglin Kong [a], Muzhou Hou [a], Shaojie Zhao [b], Feng Liu [c], Ri Su [a], Yinghao Chen [a,d,e,*]

[a] School of Mathematics and Statistics, Central South University, Changsha, 410083, Hunan, China
[b] School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, Shanghai, 201620, Shanghai, China
[c] School of Mathematics and Statistics, The University of Melbourne, Parkville, 3010, Melbourne, Australia
[d] College of Engineering, Eastern Institute for Advanced Study, Ningbo, 315201, Zhejiang, China
[e] College of Design and Engineering, National University of Singapore, 117575, Singapore

ARTICLE INFO

Dataset link: https://github.com/KongMLin/C2DR

Keywords:
Cross-domain recommendation
Click-through rate
Cold-start recommender systems
Transfer learning

ABSTRACT

The cross-domain recommendation (CDR) model addresses challenges such as data sparsity, the long tail distribution of user–item interactions, and the cold start of items or users. However, solely transferring domain-shared knowledge based on the co-occurrence patterns, without considering user preferences, leads to negative transfer in CDR. To overcome these limitations, we propose an advanced deep learning CDR model called the Domain Adversarial Deep Interest Network (DADIN) aims to facilitate smooth knowledge transfer from the source domain to the target domain and effectively alleviate negative transfer. Firstly, the joint distribution alignment of user preference in DADIN is realized by introducing a skip-connection-based domain agnostic layer, and then the domain classifier is artificially designed to distinguish between the information coming from the source domain or the target domain. Additionally, DADIN combines prediction loss, global domain confusion loss, and intra-class domain confusion losses through the Min-Max game and gradient reverse layer to achieve collaborative optimization. Two real-world experiments show the area under curve (AUC) of DADIN is 0.78 on the Huawei dataset, and it outperforms its competitors by 0.71% on the Amazon dataset, showcasing its state-of-the-art performance. Moreover, our ablation studies further demonstrate that domain adversarial technique increases the AUC by 2.34% on the Huawei dataset and 16.67% on the Amazon dataset, respectively.

1. Introduction

Recommendation systems have gained significant attention in both industry and academia as a means to enhance the utilization of information (Mai, Fan, & Shen, 2009; Richardson, Dominowska, & Ragno, 2007; Yang, Guo, Liu, & Steck, 2014) as well as deal with the challenges of big data (Wu, Guo, Huang, Liu, & Xiang, 2018; Wu, Guo, Li, & Zeng, 2016) in the era of information overload. Classical recommender systems often face challenges such as sparse data (Pan, Xiang, Liu, & Yang, 2010; Song, Huang, Zhang, & Lu, 2021), long-tail distribution of user–item interactions (Zhang et al., 2021), and the cold start problem for new items or users (Kang, Hwang, Lee, & Yu, 2019), all of which arise from real-world complexities. To address these challenges, researchers have recently introduced cross-domain recommendation (CDR) models. In a typical CDR scenario, illustrated in Fig. 1(a), the domain with a high number of user–item interactions is referred to as the source domain. On the other hand, the domain with fewer interactions is considered the target domain, where the users are a subset of those in the source domain. Additionally, there is no overlap between the items in the source and target domains (Zhu, Wang, et al., 2021). The CDR model operates under the assumption that the preferences of the users in the target domain can be partly inferred from their interactions in the source domain (Cao et al., 2023; Cao, Lin, et al., 2022; Cao, Sheng, et al., 2022). This assumption allows for the resolution of data sparsity and cold-start issues in the target domain by identifying domain-shared knowledge (i.e., similarities between user preferences) in overlapping domains or scenarios and transferring that knowledge to the target domain (Li et al., 2023; Min, Luo, Lin, Huang, & Liu, 2023).

The existing deep learning (DL) methods to achieve CDR can be classified into two primary categories. The first category includes self-supervised or semi-supervised pre-training function mapping methods, such as EMCDR (Man, Shen, Jin, & Cheng, 2017), TMCDR (Zhu, Ge, et al., 2021), and PTUPCDR (Zhu et al., 2022). The second category

* Corresponding author at: School of Mathematics and Statistics, Central South University, Changsha, 410083, Hunan, China.
E-mail address: chenyinghao1@csu.edu.cn (Y. Chen).

https://doi.org/10.1016/j.eswa.2023.122880
Received 2 July 2023; Received in revised form 30 November 2023; Accepted 6 December 2023
Available online 10 December 2023

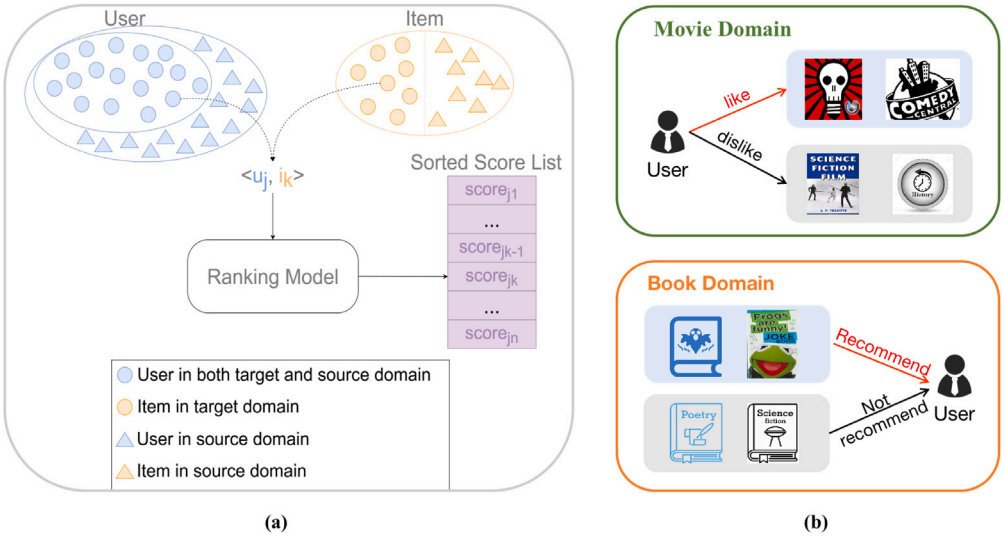


Fig. 1. (a) A diagram of the main scenarios of interest for this research. (b) A toy example in which the CDR is conducted based on the label of interactions in source domain, not only the co-occurrence.

consists of double-tower structure models that leverage cross-mapping based on multi-layer linear transformations and non-linear mappings, such as CoNet (Hu, Zhang, & Yang, 2018), CATN (Zhao, Li, Xiao, Deng, & Sun, 2020), and MiNet (Ouyang et al., 2020). It is widely believed that compared to the two-stage training model (pre-training function mapping methods), the end-to-end model (cross-mapping methods) is more effective at optimizing the objective function and incorporating prior knowledge into the model through parameter regulations. blue-Most mainstream cross-mapping methods primarily focus on the co-occurrence of users and items in the source domain when transferring domain-shared knowledge. However, these methods often overlook the importance of the user preferences towards items in the source domain. This oversight is significant because user–item interactions in the source domain can include both positive and negative samples. Relying solely on co-occurrence patterns while disregarding user preferences (i.e., interaction labels) can introduce undesired bias to the target domain, resulting in negative transfer in CDR. From a Transfer Learning (TL) perspective, this limitation can be seen as a failure to achieve the joint distribution alignment of user preferences. To illustrate the significance of considering user preferences in the source domain for CDR, we present a hypothetical scenario in Fig. 1(b). In this scenario, our goal is to utilize the user preferences observed in the movie viewing records from the source domain to recommend books that align with their interests. For example, if a user has watched horror, comedy, and science fiction movies, a co-occurrence-based cross-mapping method may recommend horror, humor, and science fiction books. However, if the user has previously expressed a dislike for science fiction in their browsing history, including science fiction in the recommendation list may lead to negative transfer, ultimately reducing user satisfaction.

In this paper, we propose the Domain Adversarial Deep Interest Network (DADIN) as a solution for mitigating negative transfer and enhancing recommendation performance in CDR. Drawing inspiration from Ganin et al. (2016), DADIN innovatively introduces the domain adversarial technique, by explicitly considering the label signals of the source domain samples to and aligning the marginal and conditional distributions at the same time, ultimately realizing the joint distribution alignment of user preferences in the source and target domains. Specifically, DADIN incorporates domain classifiers fitted with a *Domain Agnostic Layer* to obtain domain-agnostic representations through a Min-Max game, thereby achieving joint distribution alignment in the shared representation space. To the best of our knowledge, this is the first instance of domain adversarial methods being applied to recommender systems to address negative transfer in CDR. Additionally, DADIN includes a skip-connection between the input and output of the *Domain Agnostic Layer* before making final predictions, considering both domain-agnostic and domain-specific aspects of user preferences. Moreover, DADIN adopts a hierarchical attention mechanism at the bottom of the model to dynamically aggregate information from diverse behaviors and different domains.

The main contributions of our work are summarized as follows:

- DADIN leverages domain adversarial techniques to effectively address negative transfer from the source domain to the target domain in CDR. The Joint distribution alignment of user preferences is achieved through the domain-agnostic layer with a skip connection and domain classifiers.
- DADIN enhances the extraction of more detailed features by incorporating an item-level attention mechanism, which strengthens the feature representation of the user's historical behavior. Furthermore, an interest-level attention mechanism is employed to perform weighted concatenation of embeddings from different fields and domains.
- To validate the efficacy of DADIN and the domain adversarial learning method, thorough experiments are conducted on both artificial datasets and real datasets through ablation studies.
- The effectiveness of DADIN is evaluated by comparing its performance with that of both single-domain and cross-domain models on two real datasets, resulting in the achievement of state-of-the-art (SOTA) outcomes.

The rest of this paper is organized as follows: Section 2 introduces the single-domain and cross-domain recommendation models and deep network domain adaption. Section 3 describes the proposed DADIN and introduces its components. Next, Section 4 provides conceptual experiment results to verify the effectiveness of the domain adversarial method and the comparison results of DADIN with both single-domain and cross-domain baseline models on two real datasets. Finally, Section 5 is the conclusion and prospects.

2. Related work

2.1. Cross-domain recommendation

CDR addresses the issue of data sparsity by incorporating overlapping user or item information from source domains (Zhu, Wang, et al., 2021). This incorporation of information improves the performance

on the target domain even when limited data is available (Cheng et al., 2016; Juan, Zhuang, Chin, & Lin, 2016; Shan et al., 2016). The cross-mapping CDR methods can be broadly classified into three categories: Collaborative methods (Hu et al., 2018; Singh & Gordon, 2008) Content-based methods (Elkahky, Song, & He, 2015; Zhang, Yuan, Lian, Xie, & Ma, 2016) Hybrid methods (Lian, Zhang, Xie, & Sun, 2017) methods. The collaborative method is inspired by the cross-stitch network (CSN) in the field of computer vision (CV) (Misra, Shrivastava, Gupta, & Hebert, 2016). CSN extracts image features independently in two domains and achieves bidirectional knowledge transfer by linear combination of the feature maps in a high-dimensional space. Collaborative cross network (CoNet) (Hu et al., 2018) represents the use of collaborative methods in CDR. By introducing cross connection units, linear combination is transformed into linear transformation, enabling more detailed and sparse knowledge transfer. However, collaborative methods primarily focus on the mutual improvement of classification tasks in the two domains, and the shared transfer matrix may result in negative transfer. The content-based method maps information from the user side and the item side into a common embedding space through a shared embedding layer and explicitly or implicitly models their interaction effects. Multi-view deep neural network (MV-DNN) (Elkahky et al., 2015) extends the content-based method in CDR. MV-DNN considers user/item information from multiple domains as expressions in multiple views and uses multiple mapping layers to independently map their features into a high-dimensional feature space for information fusion. Introducing multi-modal item-side information on the basis of MV-DNN, as done in Zhang, Yuan, et al. (2016), facilitates more comprehensive and detailed feature extraction. However, content-based methods require the introduction of an independent DNN for each view feature, resulting in a complex model structure and redundant parameters. The hybrid method combines the above two approaches. Cccfnet (Lian et al., 2017) uses a double-tower neural network in which multilayer perceptron (MLP) extract user and item vectors based on matrix factorization (MF) algorithms, while DNN encoder rich user and item side features. The mixed interest network (MiNet) (Ouyang et al., 2020) jointly models three types of user interest in a hierarchical attention-based manner. Our method refers to the hierarchical attentions of MiNet when modeling the user's historical behavior, but introduces the *Domain Agnostic Layer* and domain classifiers based on domain adversarial learning to alleviate negative transfer. Recently applying theory and techniques from causal inference to recommendation tasks such as CDR, debiasing, and uplifting model has become a hot topic in the recommender systems community (Bi et al., 2020; Cao et al., 2023; Cao, Lin, et al., 2022; Cao, Sheng, et al., 2022). A recent work, MADD (Zhang, Li, Su, Zhu, & Shen, 2023), proposes a domain disentanglement framework that uses the attention mechanism to separate the original user embedding into domain-invariant and domain-specific features. However, MADD is highly sensitive to the training strategy and notoriously difficult to tune. Adversarial Cross Domain Recommendation (ACDR) proposed in Li, Brost, and Tuzhilin (2022) also employs adversarial training in CDR. However, ACDR focuses on modeling the heterogeneity of user behavior in different domains, while our emphasis is on extracting domain-agnostic information from participated domains and alleviating negative transfer.

2.2. Deep domain adaptation

DA is a fundamental concept in TL, aiming to map samples from source and target domains to the same feature space, thereby achieving a similar probability distribution of samples in both domains (Ghifary, Kleijn, & Zhang, 2014; Krizhevsky, Sutskever, & Hinton, 2017; Long, Cao, Wang, & Jordan, 2015; Tzeng, Hoffman, Zhang, Saenko, & Darrell, 2014). To overcome the limitation of limited training samples, a classification model trained on the source domain can be directly applied to the target domain. An early DL-based DA method called DaNN (Ghifary et al., 2014) consists of a feature mapping layer and a prediction layer. In the loss function, the Maximum Mean Discrepancy (MMD) measure between samples from the source and target domains is incorporated as a regularization term to align the two domains in the feature space. Moreover, Ganin et al. (2016) draws inspiration from the Min-Max game concept used in Generative Adversarial Nets (GAN). Ganin compares the feature mapping module to the generator in GAN and introduces a domain classifier to distinguish representations from different domains. If the domain discriminator fails to differentiate the domains, it indicates that the extracted features no longer contain domain information, which implies an implicit domain adaptation process. In recent years, the concept of dynamic distribution adaptation in classical TL has been extended to deep adversarial networks (Zhao, Wang, Guo, & Wang, 2022), highlighting the existence of a discrepancy between marginal and conditional distribution in adversarial networks. In this study, DADIN adaptively aligns the distribution between the source and target domains. By incorporating both global and intra-class domain confusion losses as regularization terms, the alignment of the marginal and conditional distributions of user preferences across the source and target domains is achieved in the same representation space. Furthermore, by utilizing the Gradient Reverse Layer (GRL) introduced in Ganin et al. (2016), a universal label classifier capable of operating in both domains is learned within the shared embedding space, taking advantage of data from both the source and target domains. This method facilitates the joint training of the domain classifiers, label classifier, and feature extractor.

3. Method

In this section, we introduce the proposed DADIN model in the order of overview, interest extraction module, domain adaptation module, and training strategy.

3.1. Overview

3.1.1. Problem formulation

In standard CDR tasks, the data from the source domain are represented as $\mathcal{D}^S = (\mathcal{I}^S, \mathcal{U}^S, \mathcal{Y}^S)$, while the data from the target domain are represented as $\mathcal{D}^T = (\mathcal{I}^T, \mathcal{U}^T, \mathcal{Y}^T)$. Here, $\mathcal{I}$, $\mathcal{U}$, and $\mathcal{Y}$ represent the item set, user set, and interaction set for each domain, respectively. Specifically, we have $\mathcal{U}^T \subseteq \mathcal{U}^S$ and $\mathcal{I}^S \cap \mathcal{I}^T = \emptyset$. The user input features for the target and source domains are denoted as $\mathbf{f}_u^T \in \mathcal{X}$ and $\mathbf{f}_u^S \in \mathcal{X}$, respectively. The item input features for their respective domains are denoted as $\mathbf{f}_i^T \in \mathcal{X}$ and $\mathbf{f}_i^S \in \mathcal{X}$, where $\mathcal{X}$ denotes the feature space. Given the observed interaction matrices $\mathcal{Y}^S \in \{0,1\}^{|\mathcal{I}^S| \times |\mathcal{U}^S|}$ and $\mathcal{Y}^T \in \{0,1\}^{|\mathcal{I}^T| \times |\mathcal{U}^T|}$ in both domains, where each element y_{ui}^T, y_{ui}^S indicates whether user $u \in \mathcal{U}^*$ is interested in item $i^* \in \mathcal{I}^*$ or not, with $* \in \{T, S\}$. In real-world scenarios, the size of user–item interactions in the target domain is often much smaller than that in the source domain, i.e., $\|\mathcal{Y}^S\|_F \geq \|\mathcal{Y}^T\|_F$. It can be assumed that a user u has a domain-agnostic consistent preference tendency towards clicking on items from different domains. Hence, leveraging the knowledge in $\mathcal{Y}^S$ with respect to user u can help to predict unknown elements in $\mathcal{Y}^T$ of the target domain, particularly in the common user cold-start scenario.

3.1.2. Forward progress

A bird's-eye view of DADIN is provided in Fig. 2. Inputs consist of features from the target domain item field $\mathbf{f}_i^T$, features from the source domain item field $\mathbf{f}_i^S$, and features from the user field $\mathbf{f}_u$. Since DADIN utilizes an asymmetric double-tower structure, each input instance contains only the user–item information from one of the domains above, denoted as $\mathbf{Input} = [\mathbf{f}_u || \mathbf{f}_i^*]$, where $* \in \{T, S\}$. We use an instance from the target domain as an example to simplify the explanation. The inputs are projected to a dense but low-dimensional space by the *Embedding Layer* (corresponding to Input Embed & Aggregation in Fig. 2(a)). DADIN incorporates a *Sequence Aggregation*

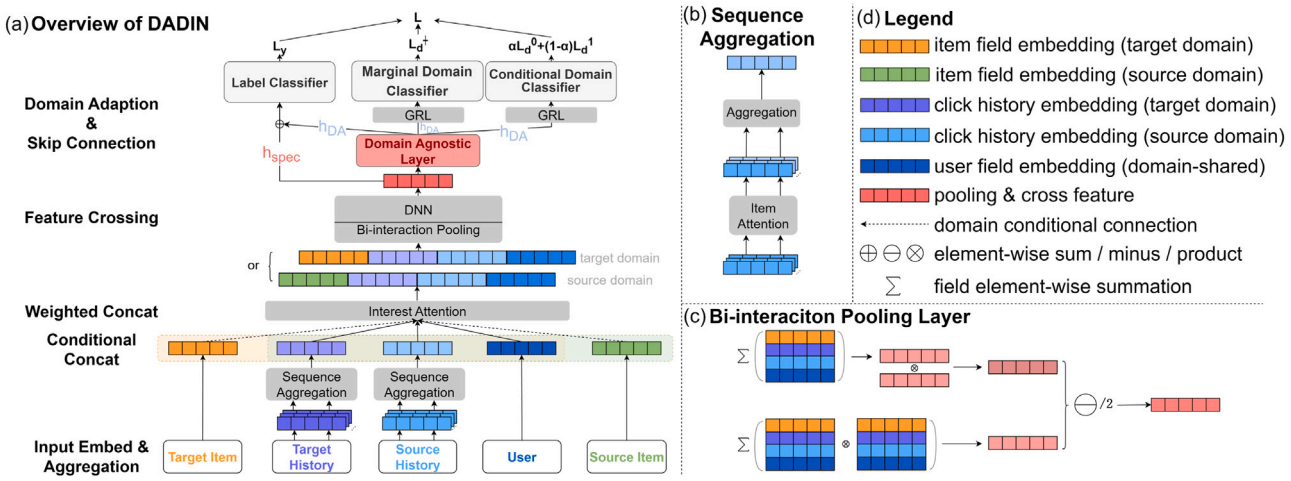


Fig. 2. Structure of Domain Adversarial Deep Interest Network (DADIN). (a) is an overview of DADIN. (b) and (c) illustrate two special operations, that is, *Sequence Aggregation* and *Bi-interaction Pooling*. (d) is an explanation of those graphics that appear previously.

operation to reinforce the embedding of user history behavior features from the target and source domains. Subsequently, an *Interest Attention Layer* is employed to achieve the weighted concatenation of embeddings from the user field and target domain item field. The concatenated embeddings are then fed into the *Feature Crossing Module*, which comprises a *Bi-interaction Pooling Layer* to perform explicit feature crossing and a *DNN* to perform implicit feature crossing. In contrast to regular CDR models, DADIN utilizes a *Domain Agnostic Layer* to extract domain-agnostic features $\mathbf{h}_{DA}$ from both domains. It then performs element-wise summation on the input and output of the *Domain Agnostic Layer*, which corresponds to a skip layer connection. This aggregation allows the incorporation of both domain-agnostic information $\mathbf{h}_{DA}$ and domain-specific information $\mathbf{h}_{spec}$. Consequently, DADIN enhances the feature representation of the target domain by utilizing source domain information. A general label classifier for both domains utilizes the final feature representation $\mathbf{h} = \mathbf{h}_{DA} + \mathbf{h}_{spec}$ to make predictions. Additionally, the *Marginal Domain Classifier* and two *Conditional Domain Classifiers* use the domain-agnostic information $\mathbf{h}_{DA}$ to calculate the global domain confusion loss $L_d^\dagger$ and intra-class domain confusion losses L_d^0 and L_d^1 respectively. This approach enables both the alignment of marginal distribution and the alignment of conditional distribution of user preferences across domains.

3.2. Interest extraction

3.2.1. Feature embedding

In DADIN, an instance represents a user interaction with an item and consists of features from either the target domain item field or the source domain item field. For a user u, it is supposed that he has features "user_ID=100009, age=5, residence=13, city=191", there are four embedding vectors $\mathbf{f}_{u,id=100009}$, $\mathbf{f}_{u,age=5}$, $\mathbf{f}_{u,city=191}$, $\mathbf{f}_{u,residence=13} \in \mathbb{R}^d$. To obtain the aggregated embedding $\mathbf{f}_u \in \mathbb{R}^d$ for user u, field pooling is performed through element-wise summation on the four vectors.

$$\mathbf{f}_u = \mathbf{f}_{u,id=100009} + \mathbf{f}_{u,age=5} + \mathbf{f}_{u,city=191} + \mathbf{f}_{u,residence=13}. \tag{1}$$

It is important to note that users may have a varying number of attributes in different datasets. Therefore, the embedding for the user profile field is the aggregation of a different number of embedding vectors $\mathbf{f}_{u,attribute} \in \mathbb{R}^d$. Similarly, embeddings from the target domain item field $\mathbf{f}_i^T \in \mathbb{R}^d$ and the source domain item field $\mathbf{f}_i^S \in \mathbb{R}^d$ can be obtained.

3.2.2. Sequence aggregation

DADIN employs sequential aggregation based on item-level attention to model the historical behavior of users in both the target and source domains. The set of embedding vectors of recently clicked items by user u in the target domain is denoted as $\{\mathbf{r}_t\}_u$, and the aggregated representation of the candidate item i^T from the target domain is denoted as $\mathbf{a}_i^T$.

$$\mathbf{a}_i^T = \sum_t \beta_t \mathbf{r}_t, \beta_t = \frac{\exp\left(\tilde{\beta}_t\right)}{\sum_{t'} \exp\left(\tilde{\beta}_{t'}\right)}, \tag{2}$$

$$\tilde{\beta}_t = \mathbf{h}_1^T \operatorname{ReLU}\left(\mathbf{W}_1 \left[\mathbf{r}_t \,||\, \mathbf{f}_i^T \,||\, \mathbf{f}_u \,||\, \mathbf{r}_t \odot \mathbf{f}_i^T\right]\right), \tag{3}$$

here $\mathbf{W}_1 \in \mathbb{R}^{d\times 4d}$ and $\mathbf{h}_1 \in \mathbb{R}^d$ are trainable parameters. In Eq. (2), $\mathbf{r}_t$ represents the embedding vector of clicked item t in the target domain, $\mathbf{f}_i^T$ represents the embedding vector of the candidate item i^T in the target domain, $\mathbf{f}_u$ represents the embedding vector of user u, and $\mathbf{r}_t \odot \mathbf{f}_i^T$ reflects the similarity between the history item t and the candidate item i^T from the target domain. Additionally, $\odot$ denotes the element-wise product operator. After the item-level attention-based sequence aggregation operation, the feature from the target history field $\{\mathbf{r}_t\}_u \in \mathbb{R}^{d\times seqlen}$ is converted into a compact feature representation $\mathbf{a}_i^T$, where $\mathbf{a}_i^T \in \mathbb{R}^d$. Through the same processes, a compact feature representation $\mathbf{b}_i^T$ for a user u and a candidate item i^T is obtained, based on the item-level attention and sequence aggregation on the interacted items of u in the source domain, denoted as $\{\mathbf{r}_s\}_u$.

3.2.3. Interest attention

For a given instance $\mathbf{Input} = \left[\mathbf{f}_u \,||\, \mathbf{f}_i^T\right]$ from the target domain, where $\mathbf{f}_i^T \in \mathbb{R}^d$ represents the vector from the target domain item field, and $\mathbf{f}_u \in \mathbb{R}^d$ represents the vector from the user profile field. In Section 3.2.2, the operation of *Sequence Aggregation* produces the vector $\mathbf{b}_i^T \in \mathbb{R}$ from the source history field, and the vector $\mathbf{a}_i^T \in \mathbb{R}$ from the target history field of user u. These four types of embedding vectors can be concatenated to form a more informative feature representation $\mathbf{m} \in \mathbb{R}^{4d}$ for the final prediction, as follows:

$$\mathbf{m} \triangleq \left[\mathbf{f}_i^T \,||\, \mathbf{f}_u \,||\, \mathbf{a}_i^T \,||\, \mathbf{b}_i^T\right]. \tag{4}$$

However, this approach of combining features incorrectly assumes that the three different types of information from the user field are equally important for the candidate item embedding vector $\mathbf{f}_i^T$. This assumption restricts the representation capacity of the model. Instead, DADIN employs *Interest Attention* to concatenate these four types of

embeddings and assign different weights. Specifically, based on the correlation between embeddings $\mathbf{f}_u, \mathbf{a}_i^T, \mathbf{b}_i^T$ and embedding $\mathbf{f}_i^T$, the weighted concatenated feature $\mathbf{m}_{i^T,u}$ can be calculated as:

$$\mathbf{m}_{i^T,u} \triangleq \left[\mathbf{f}_i^T \,||\, v_u\mathbf{f}_u \,||\, v_t\mathbf{a}_i^T \,||\, v_s\mathbf{b}_i^T\right], \tag{5}$$

and weights are calculated as:

$$\begin{aligned} v_u &= \exp\left(\mathbf{g}_u^T \,\mathrm{ReLU}\left(\mathbf{V}_u\left[\mathbf{f}_i^T \,||\, \mathbf{f}_u \,||\, \mathbf{a}_i^T \,||\, \mathbf{b}_i^T\right]\right) + b_u\right), \\ v_s &= \exp\left(\mathbf{g}_s^T \,\mathrm{ReLU}\left(\mathbf{V}_s\left[\mathbf{f}_i^T \,||\, \mathbf{f}_u \,||\, \mathbf{a}_i^T \,||\, \mathbf{b}_i^T\right]\right) + b_s\right), \\ v_t &= \exp\left(\mathbf{g}_t^T \,\mathrm{ReLU}\left(\mathbf{V}_t\left[\mathbf{f}_i^T \,||\, \mathbf{f}_u \,||\, \mathbf{a}_i^T \,||\, \mathbf{b}_i^T\right]\right) + b_t\right), \end{aligned} \tag{6}$$

where $\mathbf{V}_* \in \mathbb{R}^{d\times 4d}$ is a matrix parameter, $\mathbf{g}_* \in \mathbb{R}^d$ is a vector parameter, and b_* is a scalar parameter, where $* \in \{u, s, t\}$. In Eq. (5), the dynamic weights v_u, v_t, v_s are calculated using all available information related to user u and candidate item i, considering all three types of user information (user profile features, user history behaviors in the source domain, user history behaviors in the target domain). These weights measure the importance of the embedding vectors reflecting the user interest from three different perspectives to the embedding of candidate item i^T. Experiments show that the weights obtained through matrix multiplication and ReLU activation function may be small, resulting in the weighted concatenated vector $\mathbf{m}_{i^T,u}$ being dominated by $\mathbf{f}_i^T$. To address this issue, an exponential function is applied in Eq. (6), mapping the results to $\mathbb{R}^+$.

3.2.4. Feature crossing module

The *Interest Attention* mechanism considers the candidate item as a query and the user information from different aspects as keys to measure similarity and perform weighted concatenation. This allows us to separately represent item-related and user-related features as $\mathbf{m}_{i^T,u}$. However, in recommendation system models, feature crossing is often of great importance for modeling user–item interaction effects. To address this, the *Bi-Interaction Pooling Layer* (Zhang, Yuan, et al., 2016) is introduced to explicitly cross the feature embeddings of these four fields in a second-order manner:

$$f_{BI}(\mathbf{m}_{i^T,u}; C_\mathbf{m}) = \sum_{p=1}^{4}\sum_{q=i+1}^{4} \mathbf{m}^p_{i^T,u} c_p \odot \mathbf{m}^q_{i^T,u} c_q. \tag{7}$$

Here, $f_{BI}(\cdot)$ represents the Bi-interaction Pooling Layer, $C_\mathbf{m}$ is its parameter, $\mathbf{m}^p_{i^T,u}$ represents the embedding vector of the pth field of the weighted concatenated feature representation $\mathbf{m}_{i^T,u}$, and c_p denotes the pth element of $C_\mathbf{m}$. To provide a more intuitive interpretation, this operation can be represented as follows:

$$f_{BI}(\mathbf{m}_{i^T,u}; C_\mathbf{m}) = \frac{1}{2}\left|(\sum_{p=1}^{4}\mathbf{m}^p_{i^T,u}c_p)^2 - \sum_{p=1}^{4}(\mathbf{m}^p_{i^T,u}c_p)^2\right|. \tag{8}$$

Fig. 2(c) provides a visual explanation of Bi-interaction Pooling, which transforms the concatenated feature representation of the instance from $\mathbf{m}_{i^T,u} \in \mathbb{R}^{4d}$ to a compact representation $\mathbf{z} \in \mathbb{R}^d$.

The output obtained from $f_{BI}(\cdot)$, denoted as $\mathbf{z} \in \mathbb{R}^d$, is then fed into a DNN module with two fully-connected (FC) layers for further implicit feature crossing. The resulting output of the DNN, named $\mathbf{h}_{spec} \in \mathbb{R}^d$, represents the combination of features from different fields and carries more fine-grained feature crossing signals between user u and candidate item i^T.

3.3. Domain adaptation

3.3.1. Domain agnostic layer

The combined features $\mathbf{h}_{spec} \in \mathbb{R}^d$ are obtained in the feature crossing module through field pooling and more fine-grained feature crossing operations. However, each instance only contains the user–item pair and their interaction information from either the target or source domain item. In other words, the combined feature $\mathbf{h}_{spec} \in \mathbb{R}^d$ is derived from either the target or source domain. Considering that DADIN uses a general label classifier to output predictions $\hat{y}$ for both domains, direct knowledge transfer from the source domain to the target domain can have a negative impact on the model's performance if the two domains are dissimilar. This situation can be seen as one type of negative transfer in CDR when the marginal distribution of user preferences is not aligned, as shown by the left state in Fig. 3. Existing cross-mapping-based approaches implicitly address the alignment of user preference distributions in the source and target domains during the forward process of the model. However, due to the inadequate utilization of label information from source domain samples during knowledge transfer, irrelevant content from the source domain may be introduced into the target domain, resulting in another type of negative transfer in CDR scenarios. This process is depicted in the middle part of Fig. 3, where the marginal distributions of user preferences across the source and target domains are aligned by the model, indicating that the distribution of sample points in blue and orange becomes closer in the representation space. However, the conditional distribution of user preferences is not aligned, indicating that the classification boundary in the source domain cannot effectively distinguish samples from the target domain.

To further align the conditional distribution of user preference in CDR, DADIN introduces the *Domain Agnostic Layer*, which extracts domain-agnostic information $\mathbf{h}_{DA}$ from $\mathbf{h}_{spec}$ in a domain adversarial manner. This approach achieves alignment not only of marginal distributions but also of conditional distributions of user preferences. The process is illustrated in the right part of Fig. 3. When making predictions, DADIN combines domain-agnostic and domain-specific information. Specifically, $\mathbf{h}$ represents the intermediate state before the prediction modules, composed of domain-agnostic information $\mathbf{h}_{DA}$ and domain-related information $\mathbf{h}_{spec}$. Their relationship can be described as $\mathbf{h} = \mathbf{h}_{DA} + \mathbf{h}_{spec}$. In domain adversarial learning, $\mathbf{h}_{DA} \in \mathbb{R}^d$ is expected to confuse the domain classifier, preventing it from accurately determining if an instance belongs to the source or target domain. This inductive bias is introduced through the domain confusion loss $L_d^\dagger, L_d^0, L_d^1$ (which are described in the next section), calculated with the domain labels and the outputs of the *Marginal Domain Classifier* and *Conditional Domain Classifier*. Finally, $\mathbf{h} = \mathbf{h}_{DA} + \mathbf{h}_{spec}$ is gained through the skip connection.

3.3.2. Label classifier & domain classifier

The feature representation derived by adding domain-agnostic information to domain-specific information is first fed to the common label classifier of the target and the source domain to obtain the recommendation prediction, namely $G(\mathbf{h}) = \hat{y}$. Specifically, our label classifier contains two FC layers and adds a Dropout layer between the two FC layers to alleviate the over-fitting of the model. Formally, the label classifier is defined as follows:

$$\mathbf{h}_1 = \mathrm{Dropout}(W_2\mathbf{h} + \mathbf{b}_2), \tag{9}$$

$$\hat{y} = \sigma(W_3\mathbf{h}_1 + \mathbf{b}_3), \tag{10}$$

where $W_2 \in \mathbb{R}^{100\times d}$, $\mathbf{b_2} \in \mathbb{R}^{100}$, $W_3 \in \mathbb{R}^{1\times 100}$, and $\mathbf{b_3} \in \mathbb{R}$ are trainable parameters. The intermediate state, $\mathbf{h}_1 \in \mathbb{R}^{100}$, is used, and the Dropout layer denoted by $\mathrm{Dropout}(\cdot)$ and the Sigmoid function denoted by $\sigma(\cdot)$ are applied. On the other hand, the domain-agnostic feature representation, $\mathbf{h}_{DA}$, serves as input to the domain classifiers for calculating the domain prediction $\hat{d}$. As the purpose of $\mathbf{h}_{DA}$ is to confuse the domain classifiers, the predicted domain $\hat{d}$ should significantly differ from the true domain label, leading to considerable domain confusion loss. To achieve this, GRL (Ganin et al., 2016) is employed, which allows for the simultaneous minimization of the standard prediction loss and the domain confusion loss. The mathematical formulation of GRL, referred to as Rev(x), is used in the forward and backpropagation process.

$$\begin{aligned} &\mathrm{Rev}(\mathbf{x}) = \mathbf{x}, \\ &\frac{\mathrm{dRev}}{\mathrm{d}\mathbf{x}} = -I, \end{aligned} \tag{11}$$

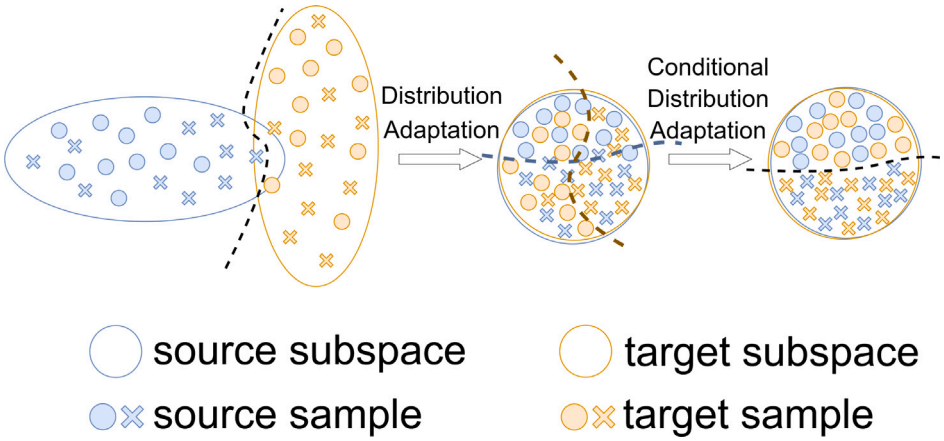


Fig. 3. Achievement of the marginal and conditional distribution alignment of user preferences across both domains, which is also the core idea of DADIN.

where I is an identity matrix. The implementation of GRL serves to accomplish the minimization of prediction loss in both domains and the maximization of domain confusion during the training stage. Further elaboration on the technical aspects will be provided in Section 3.4.

Marginal Domain Classifier $D^{\dagger}(\cdot)$ is obtained by the domain classifiers, which takes $\mathbf{h}_{DA}$ through GRL as input and outputs the domain prediction $\hat{d}$.

$$\hat{d} = \sigma(W_4 \mathrm{Rev}(\mathbf{h}_{DA}) + b_4), \tag{12}$$

where $W_4 \in \mathbb{R}^{1\times d}, b_4 \in \mathbb{R}$ are trainable parameters of the $D^{\dagger}$.

Inspired by Yu, Wang, Chen, and Huang (2019), we introduce two *Conditional Domain Classifier*s, namely D^0 and D^1 to DADIN to enhance the alignment of conditional distribution of user preference. Both classifiers assume the identical structure as $D^{\dagger}$. Different from $D^{\dagger}$, D^1 only takes h_{DA} through GRL, which is predicted to be from class 1 by the label classifier $G(\cdot)$. Especially, once the threshold T is set, the instance is believed to come from class 1 if $\hat{y}$ from label classifier outnumbers T. Subsequently, $\hat{d}^1$ is computed by feeding to D^1 for domain prediction. Analogously, the procedure is also replicated for D^0.

$$\hat{d}^1 = \sigma(W_5 \mathrm{Rev}(\hat{y} \times \mathbf{h}_{DA}) + \mathbf{h}_5), \tag{13}$$

$$\hat{d}^0 = \sigma(W_6 \mathrm{Rev}((1-\hat{y}) \times \mathbf{h}_{DA}) + \mathbf{h}_6). \tag{14}$$

By introducing the *Marginal Domain Classifier* $D^{\dagger}$ and global domain confusion loss $L_d^{\dagger}$, the marginal distribution alignment of the target and source domain is conducted. The conditional distribution alignment is realized by using two *Conditional Domain Classifier*s D^0, D^1 and intra-class domain confusion losses L_d^1, L_d^0. The joint distribution alignment of features from both domains based on domain adversarial learning greatly enhances knowledge transfer from the source domain to the target domain, and the competitive results of the domain adversarial method will be discussed in Section 4.

3.4. Training strategy

The cross-entropy loss is adopted as the loss function. For the standard prediction loss L_y, instances from both target and source domains are calculated as:

$$L_y = \frac{1}{n}\sum_{i=1}^{n}[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)], \tag{15}$$

where y_i is the ground truth of the instance i, and $\hat{y}_i$ is corresponding prediction by label classifier G, and n is the number of instances from both domains in the training set.

For the global domain confusion loss $L_d^{\dagger}$ aligning marginal distribution, the same instances are used to calculate the loss, but instances from target domain are assigned with domain label $d_i = 1$, and those from source domain with domain label $d_i = 0$, then there is:

$$L_d^{\dagger} = \frac{1}{n}\sum_{i=1}^{n}[d_i \log \hat{d}_i + (1-d_i)\log(1-\hat{d}_i)], \tag{16}$$

where d_i is the domain label of the instance i, and $\hat{d}_i$ is the corresponding domain prediction by $D^{\dagger}$.

The intra-class domain confusion losses L_d^1 and L_d^0 are introduced to align conditional distribution of user preference. Different from L_y and $L_d^{\dagger}$, both L_d^1 and L_d^0 only incorporate a subset of the total instances to calculate the loss. Specifically, a threshold T is set, if $\hat{y}_i \geq T$, then instance i will be assigned to be the positive example regardless of its domain, the feature representation of instance $\mathbf{h}_{DA}$ will be fed-forward to domain classifier D^1, and intra-class domain confusion losses L_d^1 can be calculated as:

$$L_d^1 = \frac{1}{\rho}\sum_{i=1}^{\rho}[d_i \log \hat{d}_i^1 + (1-d_i)\log(1-\hat{d}_i^1)]. \tag{17}$$

On the contrary, if $\hat{y}_i < T$, the situation is exactly the similar:

$$L_d^0 = \frac{1}{\tau}\sum_{i=1}^{\tau}[d_i \log \hat{d}_i^0 + (1-d_i)\log(1-\hat{d}_i^0)], \tag{18}$$

where ρ denotes the number of instances whose label prediction y_i outnumbers T and τ is the number of instances whose label prediction y_i is less than T, $n = \rho + \tau$, $\hat{d}_i^1$ and $\hat{d}_i^0$ are the domain prediction of D^1 and D^0 respectively.

All trainable parameters in DADIN are optimized by minimizing the total loss L, which is defined in Fig. 2(a) as:

$$L = \lambda_1 L_y + \lambda_2 L_d^{\dagger} + \lambda_3(\alpha L_d^0 + (1-\alpha)L_d^1), \tag{19}$$

where α is a hyperparameter that balances the importance of two intra-class domain confusion losses. For our experiment, the default value is $\lambda_1 = \lambda_2 = \lambda_3 = 1$, unless otherwise specified. It should be noted that for recommendation tasks that can be considered binary classification tasks (such as click-through rate (CTR) prediction), the importance of domain confusion loss in both categories should be the same, biasing towards either category will increase the prediction loss. Thus, the total loss L can be reduced to:

$$\begin{aligned} L(\theta_f, \theta_y, \theta_d^{\dagger}, \theta_d^1, \theta_d^0) = &\frac{1}{n}\sum_{i=1}^{n} L_y^i(\theta_f, \theta_y) + \frac{1}{n}\sum_{i=1}^{n} L_d^{\dagger i}(\theta_f, \theta_d^{\dagger}) \\ &+ \alpha(\frac{1}{\tau}\sum_{i=1}^{\tau} L_d^{0i}(\theta_f, \theta_d^0)) \\ &+ (1-\alpha)(\frac{1}{\rho}\sum_{i=\tau+1}^{n} L_d^{1i}(\theta_f, \theta_d^1)), \end{aligned} \tag{20}$$

where, $\theta_f, \theta_y, \theta_d^{\dagger}, \theta_d^1, \theta_d^0$ are parameters of feature extractor of the model (*Domain Agnostic Layer and its former components*), label classifier is demonstrated as G, *Marginal Domain Classifier* is demonstrated as $D^{\dagger}$, and two *Conditional Domain Classifier*s is demonstrated as D^1, D^0 respectively.

The training process of DADIN can be regarded as finding the saddle point $\theta_f, \theta_y, \theta_d^{\dagger}, \theta_d^1, \theta_d^0$ such that

$$(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d^{\dagger}, \hat{\theta}_d^1, \hat{\theta}_d^0) = \arg\min_{\theta_f, \theta_y, \theta_d^{\dagger}, \theta_d^1, \theta_d^0} L \ . \tag{21}$$

Theoretically, the optimal solution defined by Eq. (21) can be found as a stationary point of the following gradient updates:

$$\begin{aligned}
\theta_f^{(t+1)} &\longleftarrow \theta_f^{(t)} - \mu\left(\frac{\partial L_y^i}{\partial \theta_f} + \frac{\partial L_d^{\dagger i}}{\partial \theta_f} + \alpha\frac{\partial L_d^{0i}}{\partial \theta_f} + (1-\alpha)\frac{\partial L_d^{1i}}{\partial \theta_f}\right), \\
\theta_y^{(t+1)} &\longleftarrow \theta_y^{(t)} - \mu\frac{\partial L_y^i}{\partial \theta_y}, \\
\theta_d^{\dagger(t+1)} &\longleftarrow \theta_d^{\dagger(t)} - \mu\frac{\partial L_d^{\dagger i}}{\partial \theta_d^{\dagger}}, \\
\theta_d^{0(t+1)} &\longleftarrow \theta_d^{0(t)} - \mu \times \alpha\frac{\partial L_d^{0i}}{\partial \theta_d^0}, \\
\theta_d^{1(t+1)} &\longleftarrow \theta_d^{1(t)} - \mu \times (1-\alpha)\frac{\partial L_d^{1i}}{\partial \theta_d^1},
\end{aligned} \tag{22}$$

where μ is the learning rate. As two optimizers encapsulated in PyTorch, stochastic gradient descent (SGD) algorithm and Adam algorithm (Kingma & Ba, 2014) can used to training our model. Moreover, from Eq. (22), it can be seen that due to the differences in model structure and feature scales, the gradient values of parameters in different components of the model may vary significantly during the training process. This is reflected in the trend of loss curves in joint training of multiple tasks (L_y to L_d^0, L_d^1), which will be further discussed in Section 4.

Algorithm 1 Algorithm of DADIN

Require: source domain data, target domain data and hyper-parameters
Ensure: parameters F of DADIN
1: Initialize F;
2: **for** each epoch during the training **do**
3: Resample instances with feature from target domain and source domain;
4: Calculate the embedding vector of user field: $\mathbf{f}_u$ and the embedding vector of item field: $\mathbf{f}_i^S$ or $\mathbf{f}_i^T$;
5: Employ *Sequence Aggregation* to calculate the compact feature representation of target history field: $\mathbf{a}_i^*$ by Equation (2) and the compact feature representation of source history field: $\mathbf{b}_i^*$;
6: Employ *Interest Attention* to weighted concatenate four field vector ($\mathbf{f}_u, \mathbf{f}_i^S, \mathbf{a}_i^S, \mathbf{b}_i^S$ or $\mathbf{f}_u, \mathbf{f}_i^T, \mathbf{a}_i^T, \mathbf{b}_i^T$) as Equation (5) and (6): $\mathbf{m}_{i^*,u}$;
7: Employ *Feature Crossing Layer* f_{BI} and DNN to obtain the combination of features from different fields: $\mathbf{h}_{spec}$;
8: Employ *Domain Agnostic Layer* to compute domain-agnostic information: $\mathbf{h}_{DA}$ and employ GRL on $\mathbf{h}_{DA}$;
9: Employ *Marginal Domain Classifier* $D^{\dagger}$ and two *Conditional Domain Classifiers* D^1, D^0 on $\mathbf{h}_{DA}$ to obtain the domain predictions: $\hat{d}, \hat{d}^1, \hat{d}^2$ by Equations (12)–(14);
10: Calculate the compose of domain-agnostic information $\mathbf{h}_{DA}$ and domain-specific information $\mathbf{h}_{spec}$: $\mathbf{h} = \mathbf{h}_{DA} + \mathbf{h}_{spec}$;
11: Employ the label classifier G on $\mathbf{h}$ to calculate the label prediction: $\hat{y}$ by Equation (10);
12: Calculate the total loss L by Equation (19) and update F.
13: **end for**

4. Experiments

We provide three sets of conceptual experiments, which are conducted on artificial datasets. By visualizing the intermediate states of inputs under different model variants, the superiority and interpretability of DADIN are intuitively verified. Then, two real world datasets are used to performance the state-of-the-art result than existing single-domain and cross-domain recommendation technology. Ablation experiments are also provided to illustrate that each component of DADIN is indispensable. Finally, the trend of each components loss function are analyzed to help understanding the joint training strategy in our work.

4.1. Toy experiment

We conduct a Toy experiment imitating the design of *Domain Agnostic Layer* to verify the effectiveness of domain adversarial learning achieved by adding domain agnostic layers and domain classifiers. Similarly, a variant of *inter-twinning moons* 2D problem is investigated. Among them, samples of target domain data are obtained by rotating the source domain samples 35 degrees around the center. According to above operation, 300 source domain data and 300 target domain data are generated. To this end, DADIN toy model for this problem is built separately, whose architecture only adds the designed domain adversarial module to the double-tower three-layer DNN. The results of the DADIN toy model on generated two-dimensional data compared with the naive DNN with the same double-tower structure and the same number of parameters verifies: **(1)** Domain adversarial method can provide more accurate classification boundary for the target domain. **(2)** Domain confusion loss can confuse domain information to get an invalid classification boundary of the domain classifiers. **(3)** *Domain Agnostic Layer* can extract domain-agnostic features of data from both domains to enhance knowledge transfer from the source domain to the target domain.

Firstly, we try to explain whether the domain adversarial method can enhance target domain discrimination. Here, 300 source domain data and 150 target domain data (half of source domain data) are used to train classifiers of DADIN and DNN, to simulate a real cold start scenarios. The trained models are evaluated on all samples in the target domain, and their results are shown in Fig. 4.

Fig. 4 reveals the classification boundary of DADIN as superior, demonstrating consistency with the data distribution of target domain. In contrast, the DNN boundary, while seemingly accurate, overfits the training data, lacking full understanding data distribution of the target domain. These results from a lack of use of source domain data in training the target domain classifier. Consequently, the cross-domain information remains underutilized and only the marginal distribution is aligned in the embedding space. The conditional distribution between target and source domains remains unlearned. This underscores the efficacy of domain adversarial method for enhancing target domain classification via conditional distribution alignment.

Next, we evaluate whether the domain confusion loss can sufficiently confuse domain information, leading to an invalid boundary for domain classification. For this experiment, we train DADIN and DNN with all samples for the domain classification task, comparing the respective domain classifiers. The visualization results are depicted in Fig. 5.

As shown in Fig. 5(b), while the DNN does learn the pattern of source and target domains (with the target domain distribution obtained by a 35-degree rotation of the source domain distribution), its classification boundary still falls short. This owes to DNN's lack of a structure that fully segregates domain-specific and domain-agnostic information. In contrast, DADIN employs a domain separation method, adding a skip-connection-based Domain Agnostic Layer and domain confusion loss to segregate domain information. As demonstrated in Fig. 5(b), the domain-agnostic information $\mathbf{h}_{DA}$ completely confounds domain classifiers of DADIN.

Further, to probe the efficacy of the *Domain Agnostic Layer* in extracting domain-agnostic features from both domains' data, we train DADIN-toy using samples from both domains. We deploy dimensional

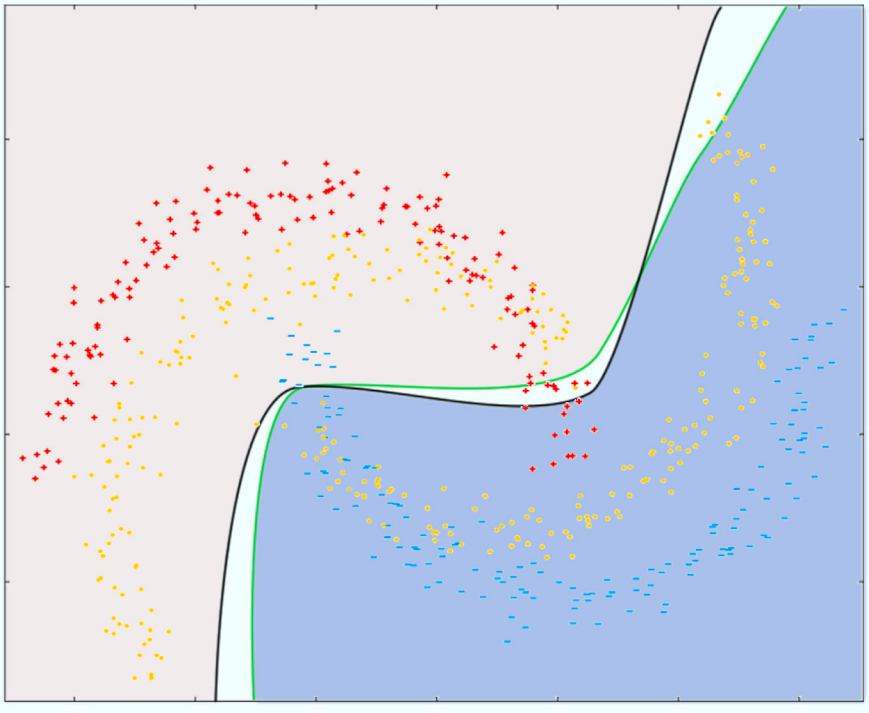
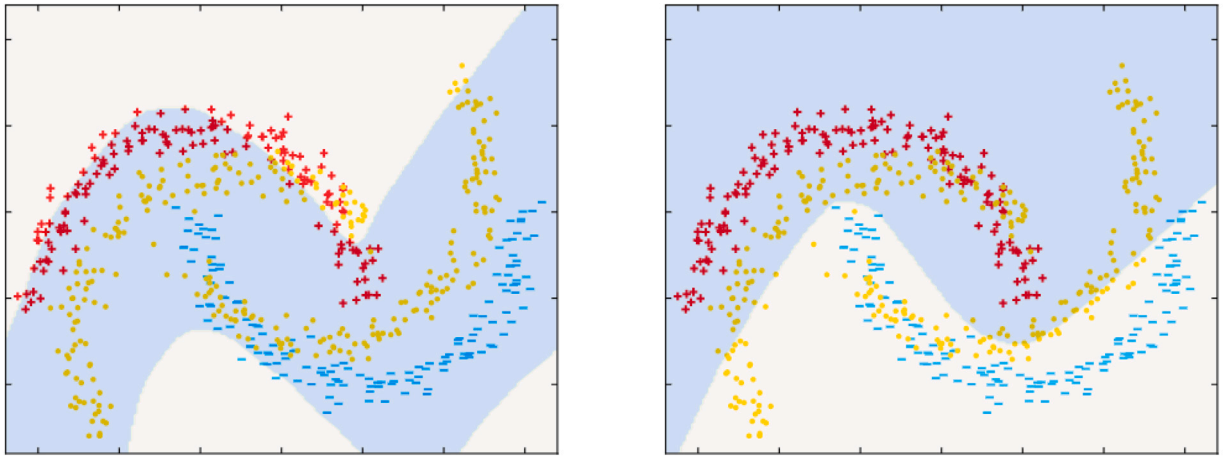


Fig. 4. Test results of DADIN-toy and DNN on artificial datasets, in which red point "+" is the positive sample of the source domain, light blue point "-" is the negative sample of the source domain, the yellow point is the target domain sample, the light red part is the area predicted as the positive example, the blue part is the area predicted as the negative example, the black curve is the classification boundary of DADIN, and the green curve is the classification boundary of DNN.



(a) The domain discrimination classification boundary of DADIN, almost all samples are classified as negative cases.

(b) The domain discrimination classification boundary of DNN, and about half of the samples can be correctly classified.

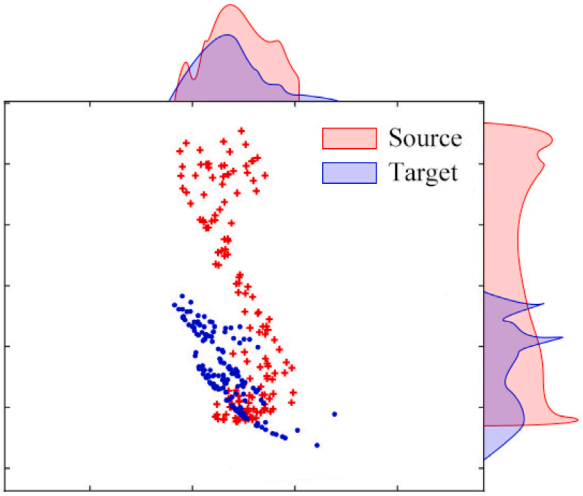
Fig. 5. The domain discrimination classification boundaries of DADIN and DNN.

reduction visualization technology to illustrate the distribution of the intermediate states $\mathbf{h}_{spec}$ and $\mathbf{h}_{DA}$, pre and post *Domain Agnostic Layer* transformation, respectively. The intermediate states $\mathbf{h}_{DA}, \mathbf{h}_{spec} \in \mathbb{R}^d$ are transformed into a two-dimensional vector via principal component analysis (PCA). This representation allows for an intuitive observation of their distribution, with positive and negative samples from different domains depicted as distinct color points.

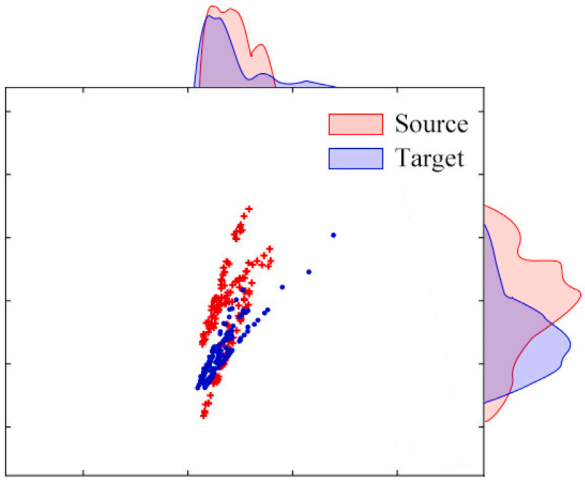
Fig. 6 demonstrates that post *Domain Agnostic Layer* transformation, the positive samples of $\mathbf{h}_{DA}$ of source and target domain align more closely in the two-dimensional space than $\mathbf{h}_{spec}$. This implies that the *Domain Agnostic Layer* effectively extracts domain-agnostic features from source and target domain samples, aligning marginal and conditional distributions in the feature space. This validation underscores the utility of our *Domain Agnostic Layer*. The feasibility of the proposed domain adversarial approach has been demonstrated using artificial datasets for all three purposes.

4.2. Experiments setup

Before exploring real dataset experiments, it is vital to outline the key details of experiments, specifically: **(1)** dataset selection, **(2)** construction of the cold start scenario (via the resampling method), and **(3)** independent repeated experiment settings.



(a) The intermediate state $\mathbf{h}_{spec}$ of the input data before the domain agnostic layer transformation (after PCA reduction). The red point "+" is a positive example from the source domain, and the dark blue dot is a positive example from the target domain.



(b) The intermediate state $\mathbf{h}_{DA}$ of the input data after the domain agnostic layer transformation (after PCA reduction). The red point "+" is a positive example from the source domain, and the dark blue dot is a positive example from the target domain.

Fig. 6. The visualization of the distribution of intermediate state $\mathbf{h}_{spec}$ and $\mathbf{h}_{DA}$. The blue and red probability density function curves around Fig. 6(a) and (b) represent the distribution of target domain samples and source domain samples after dimensionality reduction respectively.

Dataset The performance of DADIN is evaluated and compared with other models using two public real-world datasets. First is the dataset from Huawei's 2022 cross-domain CTR prediction competition,[1] notable for its substantial size and comprehensive features. Here, the source domain constitutes user click news data, and the target domain comprises user click advertisement data. Advantageously, the target domain user set of original dataset is encompassed within the source domain user set, facilitating a cold start scenario construction. The second dataset comes from Amazon,[2] using the 'music' and 'movie' fields and capturing the overlapping user subset. The source domain includes 'movie' user behavior data, whereas the target domain houses 'music' user behavior data. We also statistically count the historical behavior in both domains to generate historical behavior features.

Resampling To emulate a genuine cold start scenario, we apply a specialized resampling algorithm. The main objective is to ensure minimal overlap between the user set of the training set of the target domain and the user set of the validation and test sets, noting that the source domain contains all the users of the target domain. The entire dataset is divided according to the user set, yielding the training, validation, and testing sets.

Evaluation Metrics

(1) AUC:

$$AUC = \frac{\sum pred_{pos} > pred_{neg}}{posistiveNum \times negativeNum},$$

where the denominator $posistiveNum \times negativeNum$ is the total count of positive and negative sample combinations and the numerator $\sum pred_{pos} > pred_{neg}$ is the count of combinations where positive samples exceed negative samples;

(2) Logloss: The binary cross-entropy loss consistent with Eq. (15).

Independent Repeat Experiments To enhance the reliability of the experimental results, ten unique sets of data are generated for each dataset using different random seeds. These datasets are employed to compare the models, maintaining consistent random seeds for the models and the experiment itself. Ultimately, we record the mean and standard deviation of the evaluation results to ensure the outcome's reliability.

Algorithm 2 Algorithm of resampling

Require: A_{train}, A_{cross}
Ensure: **train_data**, **valid_data**, **test_data**

1: Take the ads field users (Huawei has too much data and has resampled an 50% of all data additionally) as all the users to be sampled and denote them as **user_list**;
2: A_{train} of the users are taken as all the users in the training set and recorded as **user_train_list**. The users contained in **user_train_list** are used to obtain data on the ads field and feeds field respectively, and are recorded as **train_data**;
3: Sample A_{cross} in **user_train_list** as users that overlap the validation set and test set. And take out the last day records of the overlap users from the ads data in the **train_data** as the validation set (50%), namely **valid_data** and test set (50%) namely **test_data**, and delete this part of data in the training set;
4: Divide the remaining $1 - A_{train}$ of the users in **user_list** into test set and validation set, and take out the last day data of ads data according to these users and add them to the **test_data** and **validation_data** obtained in step 3: respectively;
5: Finish the sampling. Return **train_data**, **valid_data**, **test_data**.

4.3. Comparing different approaches

We perform a comparative analysis between single-domain and cross-domain methods. In single-domain model configuration, instances from the two domains are directly concatenated using the "user_Id" attribute as the key, and missing features are filled with the special token "na_value", treated as a category within categorical variables. The concatenated instance is input into the single-domain model to provide the corresponding CTR prediction, calculating the loss using true labels within its domain. As for cross-domain methods, input organization

[1] https://www.huawei.com/
[2] http://jmcauley.ucsd.edu/data/amazon/index_2014.html

Table 1
Baseline models.

	Abbr	Full name	Description
Single-domain	LR	Logistic Regression	A linear model with Sigmoid function as the activation function.
	FM	Factorization Machine	Advanced LR model that learns the weights of second-order feature interactions in a way of latent vector learning.
	Wide&Deep	Wide&Deep	Classic deep learning framework for CTR prediction consisting of a linear model as wide part and a DNN as deep part.
	xDeepFM	eXtreme Deep Factorization Machine	Powerful CTR prediction model based on Wide&Deep framework, using Compressed Interaction Network (CIN) to explicitly model higher-order crossing features in a vector-wise manner.
	DIN	Deep Interest Network	DIN assigns activation weights to the historical behavior vectors according to the candidate items and combine them as users' interest given the candidate item by an attention-based pooling method.
	NFM	Neural Factorization Machine	NFM introduces the Bi-Interaction Pooling layer to replace the concatenation layer of classic DNN to add enough feature interaction information to the bottom layer.
Cross-domain	MV-DNN	Multi-View DNN	MV-DNN maps the information from the user side and the item side into a same embedding space through a shared embedding layer, then models their interactions.
	MLP++	MLP++	The naive version of CoNet modeling representation of users and items separately with two independent MLP.
	CSN	Cross-Stitch Network	CSN extracts the features of images in the two fields independently and achieves the goal of bidirectional knowledge transfer by linear combination of feature maps.
	CoNet	Collaborative cross Network	CoNet introduces cross connection units that changes linear combination to linear transformation realizing more fine-grained and sparse knowledge transfer.
	MiNet	Mixed Interest Network	MiNet jointly models user interest from multiple domains, weighting them in a hierarchical attention-based manner.
	ACDR	Adversarial Learning for Cross Domain Recommendation	ACDR incorporates adversarial learning to capture both global user preferences and domain-specific user preferences across different domains.
	MADD	Multiple-Level Attention-Based Domain Disentanglement	MADD utilizes the attention mechanism to construct personalized preferences by disentangling raw user behavior into domain-shared and domain-specific features.
	DADIN	Domain Adversarial Deep Interest Network	Domain Adversarial Deep Interest Network proposed in this paper.

for different cross-domain models follows the same procedures as in their original papers. Certain two-stage models, like MV-DNN, retain the feature extraction part and modify the second stage by adjusting the model's loss function to suit the cross-domain CTR prediction task. Abbreviations, full names, and descriptions of the comparison methods are shown in Table 1.

4.3.1. Baseline

Baseline hyperparameter settings Considering the equipment limitations and model performance, the different hyperparameters selected for the model in the comparison experiment are shown in the Table 2. Adam (Kingma & Ba, 2014) is used as the optimizer for all models, the embedding dimension of all models is fixed as 64, and early stopping is used in training to obtain the optimal model. Moreover, cross-entropy loss is used as the loss function for both single-domain models and cross-domain models (see Table 2).

Table 2
Baseline hyperparameter settings.

Model		Batch size	Hidden units	Learning rate
Single-domain	LR	1000	–	1.00E-03
	FM			
	Wide&Deep			
	xDeepFM	1000	[512, 512, 512]	1.00E-03
	DIN			
	NFM			
Cross-domain	MV-DNN	2000	[512, 128, 64]	1.00E-04
	MLP++			
	CSN			
	CoNet			
	MiNet			
	ACDR			
	MADD			
	DADIN8			
	DADIN++			

4.3.2. Results

Table 3 displays the results of different models on the two datasets, indicating the relative improvement of DADIN in terms of AUC compared to other baselines over other baselines with respect to AUC. Specifically, ten independent repeat experiments on two datasets are conducted for the proposed model and all competitors. The mean and standard deviation of the two evaluation metrics are listed in Table 3. For the fair comparison, the number of parameters for all methods are also provided, DADIN surpasses several baseline models on two datasets without augmenting the size of network model.

Single-domain Methods Single-domain models generally does not perform as well as cross-domain models on two datasets, except for MV-DNN. In Huawei dataset, the AUC of predict value of certain model (e.g., Wide&Deep) surpasses that of MV-DNN, and even approaches MLP++ and MiNet. This is due to these single-domain models possessing superior feature extraction modules (Wide&Deep, xDeepFM, NFM) or leveraging user's historical behavior information (DIN), allowing them to better learn the distribution of target domain on larger datasets. However, on the smaller Amazon dataset, the single-domain model's performance is inferior to the cross-domain models. This is an expected result, owing to the limited information on the target domain's distribution.

Table 3
Cross-domain CTR prediction performance on the two real datasets of the proposed method and baselines.

Dataset	Model		AUC	Improve of AUC(%)	Logloss	#Parameters
Amazon	Single-domain	LR	0.50862(± 0.008)	27.07%	0.57614	**0.22** M
		FM	0.51930(± 0.002)	25.54%	1.19791	14.43 M
		Wide&Deep	0.49617(± 0.004)	28.85%	1.47038	15.18 M
		xDeepFM	0.50537(± 0.008)	27.53%	1.51702	14.47 M
		DIN	0.50138(± 0.004)	28.11%	2.51903	14.98 M
		NFM	0.57002(± 0.005)	18.26%	1.94616	14.99 M
	Cross-domain	MV-DNN	0.62652(± 0.035)	10.16%	0.62100	30.43 M
		MLP++	0.66092(± 0.037)	5.23%	0.53870	4.58 M
		CSN	0.66714(± 0.032)	4.34%	0.52994	4.58 M
		CoNet	0.66674(± 0.029)	4.39%	**0.52932**	4.58 M
		MiNet	0.69240(± 0.046)	0.71%	0.55484	4.49 M
		ACDR	0.69237(± 0.044)	0.72%	0.55454	4.49 M
		MADD	0.66694(± 0.026)	4.56%	0.53132	4.58 M
		DADIN++	**0.69738(±0.008)**	–	0.57720	4.61 M
Huawei	Single-domain	LR	0.72240(± 0.008)	7.29%	0.10116	**0.92**M
		FM	0.69917(± 0.027)	10.27%	0.12173	59.46 M
		Wide&Deep	0.76328(± 0.015)	2.04%	0.08665	61.19 M
		xDeepFM	0.74465(± 0.009)	4.44%	0.10059	59.59 M
		DIN	0.76226(± 0.016)	2.18%	0.08670	60.23 M
		NFM	0.73720(± 0.009)	5.39%	0.10241	59.95 M
	Cross-domain	MV-DNN	0.71864(± 0.001)	7.77%	0.07829	40.24 M
		MLP++	0.77510(± 0.003)	0.53%	0.07326	14.39 M
		CSN	0.77859(± 0.006)	0.08%	0.07276	14.39 M
		CoNet	0.77186(± 0.007)	0.94%	0.07345	14.39 M
		MiNet	0.76363(± 0.004)	2.00%	0.07443	14.30 M
		ACDR	0.76361(± 0.004)	2.00%	0.07413	14.30 M
		MADD	0.77161(± 0.006)	0.98%	0.07313	14.39 M
		DADIN++	**0.77921(±0.003)**	–	**0.07242**	14.42 M

Cross-domain Methods Among various cross-domain CTR prediction models, MV-DNN with two-stage modeling performs the poorest on the two real datasets, implying an advantage for one-stage end-to-end CTR prediction models in cross-domain CTR prediction tasks. The performance of CoNet, CSN, MLP++, and MADD show no significant variation on Huawei and Amazon datasets, indicating that the shared cross connection coefficient of the intermediate state of the double-tower base network, whether represented by matrix $\mathbf{H}_D$, or scalar α_D, or even an attention-based dynamic weight, has no substantial impact. Notably, MiNet and ACDR perform slightly worse on the Huawei dataset than the CoNet series models and MADD, but they outperform them on the Amazon dataset. Considering the challenge of knowledge transfer between movie and music rating data in the Amazon dataset, we suggest that more granular operations of different domain-aware embeddings can enhance the transfer of knowledge between two distantly distributed domains, as demonstrated by MiNet and ACDR. Compared to the most competitive baselines in both datasets, our proposed model, DADIN, shows significant improvement, outperforming CSN by 0.08% on the Huawei dataset and MiNet by 0.71% on the Amazon dataset.

In summary, single-domain methods such as LR and Wide&Deep show an acceptable performance on Huawei datasets due to the strong correlation between source and target domains. However, the performance of the single-domain approach deteriorates significantly on the Amazon dataset due to the weak correlation between the source and target domains. This discrepancy in performance highlights the representativeness and diversity of the two datasets we have selected. On the contrary, DADIN achieves the best performance on both datasets and maintains a high level of accuracy even on the Amazon dataset, which has a weak correlation. This showcases the remarkable generalization ability of DADIN.

4.4. Ablation study

To quantitatively validate the efficacy of each component of our proposed model, we examine ten variants of the DADIN model and conduct experiments on two real datasets. To maintain comparability, all model variants follow the same hyperparameter settings. The variants are illustrated in Table 4.

Table 4
Illustration of variants.

Variant	Meaning
DADIN1	Replacing item-level attention-based sequence aggregation with average pooling.
DADIN2	Replacing interest-level attention-based concatenation with equal weight concatenation.
DADIN3	Removing the *Bi-Interaction Pooling layer*.
DADIN4	Removing *DNN* layers.
DADIN5	Removing both the *Bi-Interaction Pooling Layer* and *DNN* layers.
DADIN6	Removing the *Domain Agnostic Layer*.
DADIN7	Removing the global domain confusion loss $L_d^{\dagger}$.
DADIN8	Removing the intra-class domain confusion losses L_d^0 and L_d^1.
DADIN9	Removing both global domain confusion loss and intra-class domain confusion losses.
DADIN++	The complete model version.

Table 5 presents comparative results of our model and all its variants, using evaluation metrics including AUC and Logloss as previously discussed in Section 4.2. Evidently, the complete DADIN version yields higher AUC values on both datasets than its variants. The relative improvement of DADIN++ is calculated using the AUC indicator compared to other variants in Table 5.

On the Huawei dataset, DADIN++ exhibits the most remarkable improvement versus DADIN6 (36.57%), where the *Domain Agnostic Layer* is removed. DADIN++ attains an AUC of 0.788, contrasting with DADIN6's AUC of 0.5, indicating model collapse after the removal of the *Domain Agnostic Layer*. In DADIN6, with the *Domain Agnostic Layer* eliminated, domain confusion loss constraint ensures that the intermediate state of the label classifier, *Marginal Domain Classifier*, and *Conditional Domain Classifier* inputs are essentially the domain-agnostic information $\mathbf{h}_{DA}$. Yet, experiments show that employing only $\mathbf{h}_{DA}$ to predict the CTR on the target domain results in model collapse, implying ineffective output from the label classifier. From a multi-task learning perspective, domain confusion loss optimization detrimentally

Table 5
The comparative results of DADIN++ and all its variants.

Model	Amazon			Huawei		
	AUC	improve of AUC(%)	LogLoss	AUC	improve of AUC(%)	LogLoss
DADIN_1	0.67336	3.23%	0.63213	0.78174	0.83%	0.07012
DADIN_2	0.66605	4.28%	0.69786	0.78012	1.03%	0.07078
DADIN_3	0.53549	23.04%	**0.60213**	0.78181	0.82%	0.06884
DADIN_4	0.69117	0.67%	0.72632	0.75590	4.11%	0.07154
DADIN_5	0.64947	6.66%	0.84493	0.75744	3.91%	0.07187
DADIN_6	0.51749	25.63%	2.80954	0.50000	36.57%	0.24745
DADIN_7	0.65085	6.46%	0.62219	0.77122	2.16%	0.07112
DADIN_8	0.59945	13.85%	0.65974	0.78230	0.76%	0.07015
DADIN_9	0.57983	16.67%	0.90197	0.76980	2.34%	0.07215
DADIN++	**0.69581**	–	0.61499	**0.78828**	–	**0.06799**

impacts CTR prediction loss optimization. From a domain adaptation perspective, DADIN6 only realizes marginal distribution alignment in the feature space across domains, neglecting conditional distribution alignment. Introducing the skip-connection-based *Domain Agnostic Layer* ensures both alignments, bolstering model robustness. DADIN++ shows a 3.91% improvement over DADIN5. Given the multitude of features across various fields in the Huawei dataset, it is imperative to use stacked feature crossing modules for serial, explicit, and implicit feature crossing. Compared to the non-adversarial version of DADIN9, DADIN++ just improves by 2.34% on the AUC. Therefore, the scheme using domain adaption based on domain adversarial learning to enhance cross-domain knowledge transfer has no obvious gain on this dataset.

On the Amazon dataset, due to the inherent challenge of transferring knowledge from movie to music rating records, coupled with biases arising from converting item rating data to binary labels, the DADIN model and its variants see reductions in AUC absolute values compared to the Huawei dataset. However, DADIN++ still achieves an acceptable near-0.7 AUC. In line with Huawei dataset results, DADIN++ greatly improves upon DADIN6 (without *Domain Agnostic Layer*) and DADIN3 (without explicit feature crossing module), increasing AUC by 25.63% and 23.04%, respectively. Interestingly, due to the vast distribution distance between source domain film rating records and target domain music rating records, DADIN++ shows significant improvement over the non-adversarial version of DADIN9 (16.67%). This validates the importance of our domain adversarial learning-based domain adaptation approach in enhancing cross-domain knowledge transfer when the source and target domain are distant, guaranteeing method effectiveness and scalability.

The role of attention-based sequence aggregation and attention-based weighted concatenation is highlighted in DADIN++ versus DADIN1 and DADIN2 results. This enhancement is slightly more evident on the Amazon dataset compared to the Huawei dataset due to the larger distributional distance between source and target domains in the former. Under such conditions, the model must capture user interest in the current information more comprehensively. For instance, attention-based sequence aggregation can improve the representation of user's historical behavior (DADIN++ versus DADIN1). Alternatively, when a given item i^* is presented, the model can assign varying significance to user field features based on the correlation of the embedding vector, enabling attention-based weighted concatenation (DADIN++ versus DADIN2). The resulting concatenation embedding vector is thus more suitable for the CTR prediction task. To ascertain that the weights in the attention-based weighted concatenation (Eq. (5)) employed in DADIN++ are meaningful and not equal as in Eq. (4), weights v_u, v_t, v_s are recorded and visualized during the model's forward process on the test set.

Fig. 7 shows that the interest-level attention weights for both positive and negative samples in the test set exceed 1. Embedding vectors from different fields carry distinct weights, corroborating the efficacy of our approach, which deviates from equal weight embedding concatenation. Comparing the three embedding weights v_u, v_s, v_t, regardless of

Fig. 7. The mean of interest-level attention weights of positive and negative samples, and the mean of the means. The darker the color, the larger the value. The figure shows the specific values of three kinds of weights v_u, v_s and v_t.

negative or positive samples, the result is $v_u > v_s > v_t$. This indicates that in the model, the user profile feature $\mathbf{f}_u$ is more significant to the CTR prediction task than the user target history field feature $\mathbf{a}_i^*$. The user source history field feature $\mathbf{b}_i^*$ is least important. The mean values of the weights v_u, v_s, v_t for negative samples all exceed those of positive samples, suggesting that for positive samples, the contribution of the user part information to the final CTR prediction is less than that of the item part information. That is, when a click behavior occurs, candidate items themselves tend to possess appealing characteristics. If a user does not click on the item, it is determined by the user's inherent attributes, including occupation, gender, and preferences reflected in historical behavior.

4.5. The loss landscape of DADIN

To better understand the model's multi-task joint training and validate the model convergence, total loss L, CTR prediction loss L_y, global domain confusion loss $L_d^\dagger$, and intra-class domain confusion losses $\alpha L_d^0 + (1-\alpha)L_d^1$ are recorded. Primarily, the loss changes in the model training process for four different weight λ settings are considered. The forward results on the data for each batch, along with the number of iterations, are depicted in Fig. 8.

In Fig. 8, the blue curve signifying the primary task, is of particular interest. In all four settings, this curve swiftly drops post-training commencement and stabilizes at a lower level around the 300 iteration, exhibiting no oscillation in subsequent training. This affirms the

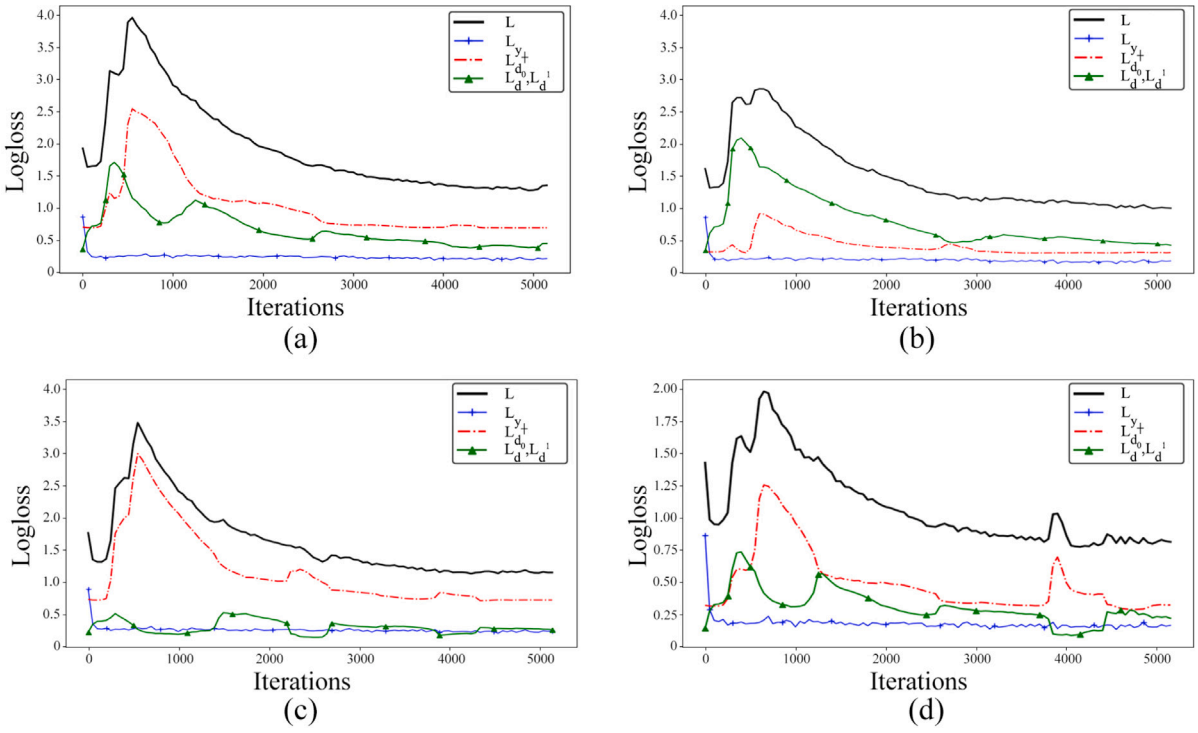


Fig. 8. (a)–(d) show the variation of each part of the loss during the training process when different λ_2, λ_3 are selected. (a) $\lambda_2 = 1.0$, $\lambda_3 = 1.0$, AUC : 0.78343. (b) $\lambda_2 = 0.5$, $\lambda_3 = 1.0$, AUC : 0.77695. (c) $\lambda_2 = 1.0$, $\lambda_3 = 0.5$, AUC : 0.77712. (d) $\lambda_2 = 0.5$, $\lambda_3 = 0.5$, AUC : 0.77665.

convergence of our model on the main task, irrespective of the weight λ of auxiliary tasks. All settings reveal an upward trend in the $L_d^{\dagger}$ and $\alpha L_d^0 + (1-\alpha)L_d^1$ curves, and consequently total loss L, during the rapid descent of L_y. This is attributed to the varying objectives in the multi-task joint training process, a normal phenomenon in multi-task learning. Following L_y's convergence, $L_d^{\dagger}$ and $\alpha L_d^0 + (1-\alpha)L_d^1$ exhibit a trade-off, with inverse changes at about 800 iterations in 8(a). Once $\alpha L_d^0 + (1-\alpha)L_d^1$ stabilizes, $L_d^{\dagger}$ declines consistently, mirroring total loss L's convergence process. Similar trends are observable in Fig. 8(b), (c), (d). The default setting ($\lambda_1 = \lambda_2 = \lambda_3 = 1$) outperforms other λ settings on AUC metrics (0.78343), affirming our hyperparameter setting. Observing various λ settings, it can be concluded that diminishing the weight λ of $L_d^{\dagger}$ or $\alpha L_d^0 + (1-\alpha)L_d^1$ in L incurs some performance degradation.

In essence, the model's multi-task joint learning is an alternate optimization process of different loss functions. Specifically, optimizing L_y may inflate total loss L while reducing L_y; $\alpha L_d^0 + (1-\alpha)L_d^1$ optimization after L_y's convergence may increase $L_d^{\dagger}$. Finally, the process of $L_d^{\dagger}$ decreasing until convergence parallels total loss L.

5. Conclusion and future work

In this paper, an innovative deep learning model, denoted DADIN, for cross-domain CTR prediction is present. DADIN leverages domain adversarial learning to facilitate the transfer of knowledge from the source domain to the target domain while simultaneously mitigating negative transfer effects in user-shared CDR. To achieve joint probability distribution alignment of user preferences, skip-join-based domain agnostic layers and tailored domain classifiers are introduced. The proposed model outperforms other single-domain and cross-domain models, setting a new state-of-the-art benchmark. Results on artificial and real-world datasets confirm the effectiveness of the model components.

Our research endeavors to "user-level relevance" in cross-domain recommendation (CDR), especially on shared users (Zhu, Wang, et al., 2021). In subsequent research, we intend to explore the utilization of domain adversarial learning for joint distribution alignment in other situations, including shared items and shared features. Although these two scenarios are intriguing, they have not been extensively studied due to the limited available benchmark datasets. Hence, we will gather and organize CDR datasets encompassing shared items and shared features to facilitate research in this domain, as well as to verify the generalization capability of DADIN method. Additionally, we will concentrate on devising a more flexible neural network architecture to enable dynamic joint distribution alignment across multiple domains, effectively accommodating diverse input data with varying modalities and training procedures.

CRediT authorship contribution statement

Menglin Kong: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Muzhou Hou:** Conceptualization, Methodology, Resources Writing – review & editing, Supervision, Project administration, Funding acquisition. **Shaojie Zhao:** Software, Validation, Formal analysis, Data curation, Writing – review & editing, Visualization. **Feng Liu:** Conceptualization, Methodology, Software, Validation, Investigation, Supervision, Resources. **Ri Su:** Writing – review & editing, Supervision, Project administration. **Yinghao Chen:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – review & editing, Supervision, Project administration Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset generated and/or analyzed as well as the codes used during the current study are available at https://github.com/KongMLin/C2DR.

Acknowledgments

This work is supported by high performance computing center at Eastern Institute for Advanced Study, Ningbo. This work is also under the aegis of the Fundamental Research Funds for Central University of Central South University No. 2022zyts0611 and the China Scholarship Council No. 202206370078.

References

Bi, Y., Song, L., Yao, M., Wu, Z., Wang, J., & Xiao, J. (2020). A heterogeneous information network based cross domain insurance recommendation system for cold start users. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 2211–2220). http://dx.doi.org/10.1145/3397271.3401426.

Cao, J., Li, S., Yu, B., Guo, X., Liu, T., & Wang, B. (2023). Towards universal cross-domain recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*. http://dx.doi.org/10.1145/3539597.3570366.

Cao, J., Lin, X., Cong, X., Ya, J., Liu, T., & Wang, B. (2022). DisenCDR: learning disentangled representations for cross-domain recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. http://dx.doi.org/10.1145/3477495.3531967.

Cao, J., Sheng, J., Cong, X., Liu, T., & Wang, B. (2022). Cross-domain recommendation to cold-start users via variational information bottleneck. In *2022 IEEE 38th international conference on data engineering* (pp. 2209–2223). http://dx.doi.org/10.1109/ICDE53745.2022.00211.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 7–10). http://dx.doi.org/10.1145/2988450.2988454.

Elkahky, A. M., Song, Y., & He, X. (2015). A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web* (pp. 278–288). http://dx.doi.org/10.1145/2736277.2741667.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, *17*(1), 1–35. http://dx.doi.org/10.48550/arXiv.1505.07818.

Ghifary, M., Kleijn, W. B., & Zhang, M. (2014). Domain adaptive neural networks for object recognition. In *Pacific rim international conference on artificial intelligence* (pp. 898–904). Springer, http://dx.doi.org/10.1007/978-3-319-13560-1_76.

Hu, G., Zhang, Y., & Yang, Q. (2018). Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 667–676). http://dx.doi.org/10.1145/3269206.3271684.

Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 43–50). http://dx.doi.org/10.1145/2959100.2959134.

Kang, S., Hwang, J., Lee, D., & Yu, H. (2019). Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1563–1572). http://dx.doi.org/10.1145/3357384.3357914.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. http://dx.doi.org/10.48550/arXiv.1412.6980, arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. http://dx.doi.org/10.1145/3065386.

Li, P., Brost, B., & Tuzhilin, A. (2022). Adversarial learning for cross domain recommendations. *ACM Transactions on Information Systems*, *14*(1), 1–25. http://dx.doi.org/10.1145/3548776.

Li, J., Li, J., Li, J., Zheng, H., Liu, Y., Lu, M., et al. (2023). ADL: Adaptive distribution learning framework for multi-scenario CTR prediction. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 1786–1790). http://dx.doi.org/10.1145/3539618.3591944.

Lian, J., Zhang, F., Xie, X., & Sun, G. (2017). CCCFNet: a content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th international conference on world wide web companion* (pp. 817–818). http://dx.doi.org/10.1145/3041021.3054207.

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning transferable features with deep adaptation networks. In *International conference on machine learning* (pp. 97–105). PMLR, http://dx.doi.org/10.48550/arXiv.1502.02791.

Mai, J., Fan, Y., & Shen, Y. (2009). A neural networks-based clustering collaborative filtering algorithm in e-commerce recommendation system. In *2009 international conference on web information systems and mining* (pp. 616–619). IEEE, http://dx.doi.org/10.1109/WISM.2009.129.

Man, T., Shen, H., Jin, X., & Cheng, X. (2017). Cross-domain recommendation: An embedding and mapping approach. In *IJCAI, vol. 17* (pp. 2464–2470).

Min, E., Luo, D., Lin, K., Huang, C., & Liu, Y. (2023). Scenario-adaptive feature interaction for click-through rate prediction. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 4661–4672). http://dx.doi.org/10.1145/3580305.3599936.

Misra, I., Shrivastava, A., Gupta, A., & Hebert, M. (2016). Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3994–4003). http://dx.doi.org/10.1109/CVPR.2016.433.

Ouyang, W., Zhang, X., Zhao, L., Luo, J., Zhang, Y., Zou, H., et al. (2020). Minet: Mixed interest network for cross-domain click-through rate prediction. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 2669–2676). http://dx.doi.org/10.1145/3340531.3412728.

Pan, W., Xiang, E., Liu, N., & Yang, Q. (2010). Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 24, no. 1* (pp. 230–235). http://dx.doi.org/10.1609/aaai.v24i1.7578.

Richardson, M., Dominowska, E., & Ragno, R. (2007). Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web* (pp. 521–530). http://dx.doi.org/10.1145/1242572.1242643.

Shan, Y., Hoens, T. R., Jiao, J., Wang, H., Yu, D., & Mao, J. (2016). Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 255–262). http://dx.doi.org/10.1145/2939672.2939704.

Singh, A. P., & Gordon, G. J. (2008). Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 650–658). http://dx.doi.org/10.1145/1401890.1401969.

Song, K., Huang, Q., Zhang, F. e., & Lu, J. (2021). Coarse-to-fine: A dual-view attention network for click-through rate prediction. *Knowledge-Based Systems*, *216*, Article 106767. http://dx.doi.org/10.1016/j.knosys.2021.106767.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. http://dx.doi.org/10.48550/arXiv.1412.3474, arXiv preprint arXiv:1412.3474.

Wu, J., Guo, S., Huang, H., Liu, W., & Xiang, Y. (2018). Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives. *IEEE Communications Surveys & Tutorials*, *20*(3), 2389–2406. http://dx.doi.org/10.1109/COMST.2018.2812301.

Wu, J., Guo, S., Li, J., & Zeng, D. (2016). Big data meet green challenges: Big data toward green applications. *IEEE Systems Journal*, *10*(3), 888–900. http://dx.doi.org/10.1109/JSYST.2016.2550530.

Yang, X., Guo, Y., Liu, Y., & Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer communications*, *41*, 1–10. http://dx.doi.org/10.1016/j.comcom.2013.06.009.

Yu, C., Wang, J., Chen, Y., & Huang, M. (2019). Transfer learning with dynamic adversarial adaptation network. In *2019 IEEE international conference on data mining* (pp. 778–786). IEEE, http://dx.doi.org/10.1109/ICDM.2019.00088.

Zhang, Y., Cheng, D. Z., Yao, T., Yi, X., Hong, L., & Chi, E. H. (2021). A model of two tales: Dual transfer learning framework for improved long-tail item recommendation. In *Proceedings of the web conference 2021* (pp. 2220–2231). http://dx.doi.org/10.1145/3442381.3450086.

Zhang, X., Li, J., Su, H., Zhu, L., & Shen, H. T. (2023). Multi-level attention-based domain disentanglement for BCDR. *ACM Transactions on Information Systems*, *41*(4), 1–24. http://dx.doi.org/10.1145/3576925.

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 353–362). http://dx.doi.org/10.1145/2939672.2939673.

Zhao, C., Li, C., Xiao, R., Deng, H., & Sun, A. (2020). CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 229–238). http://dx.doi.org/10.1145/3397271.3401169.

Zhao, Y., Wang, K., Guo, G., & Wang, X. (2022). Learning compact yet accurate generative adversarial networks for recommender systems. *Knowledge-Based Systems*, *257*, Article 109900. http://dx.doi.org/10.1016/j.knosys.2022.109900.

Zhu, Y., Ge, K., Zhuang, F., Xie, R., Xi, D., Zhang, X., et al. (2021). Transfer-meta framework for cross-domain recommendation to cold-start users. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 1813–1817). http://dx.doi.org/10.1145/3404835.3463010.

Zhu, Y., Tang, Z., Liu, Y., Zhuang, F., Xie, R., Zhang, X., et al. (2022). Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining* (pp. 1507–1515). http://dx.doi.org/10.1145/3488560.3498392.

Zhu, F., Wang, Y., Chen, C., Zhou, J., Li, L., & Liu, G. (2021). Cross-domain recommendation: challenges, progress, and prospects. http://dx.doi.org/10.48550/arXiv.2103.01696, arXiv preprint arXiv:2103.01696.