Cross Domain Deep Collaborative Filtering without Overlapping Data

Meng Liu*, Jianjun Li*†, Guohui Li*, Zhiqiang Guo*, Chaoyang Wang*, Peng Pan*
*Huazhong University of Science and Technology (HUST), WuHan, China
Email: {sunshinel, jianjunli, guohuili, zhiqiangguo, sunwardtree, panpeng}@hust.edu.cn

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191115

***Abstract*—Cross-domain collaborative filtering (CDCF) is an effective method to alleviate the data sparsity problem by transferring knowledge from a source domain to assist the learning of a target domain. However, most of the existing CDCF approaches require that the two domains have at least one overlapping side (either on user or item) and the raw data can be fully shared across domains, which is difficult to be satisfied in reality due to corporate barriers and the risk of user privacy leakage. Although there are some attempts on applying CDCF to the scenario without overlapping data by transferring cluster-level rating patterns, these methods fail to mine the complex connections between the two domains, which makes their performance still not satisfactory. To address these problems, we propose a novel deep Interaction Distribution Transfer (IDT) framework, which extracts and transfers knowledge from the feature distribution formed by the whole dataset rather than specific data. In this way, the knowledge is embedded into high-order features for transfer, which can effectively avoid privacy leakage during the data sharing process. Moreover, as a flexible framework, IDT obtains powerful feature extraction ability from the base model, which guarantees its superior performance. Extensive experiments on three benchmark datasets are conducted and the results verify the effectiveness of the proposed framework.**

***Index Terms*—Cross-domain recommendation, Graph convolutional network**

I. Introduction

Modern recommender system in general are based on collaborative filtering (CF), since it does not require auxiliary information. Among various CF techniques, matrix factorization (MF) [1], which represents users and items by learning a latent space, has become a *de facto* standard for latent factor based recommendation. Recently, some studies try to introduce deep learning techniques, such as multi-layer perceptron (MLP) [2]–[4] and graph neural networks (GNN) [5]–[7], into recommendation and achieves remarkable progress. However, these models also require more data, which makes the data sparsity problem become a major limitation of them.

The development of transfer learning brings a new opportunity for addressing the data sparsity problem. By applying transfer learning to collaborative filtering, many cross-domain collaborative filtering (CDCF) algorithms have been proposed, whose main idea is to first learn knowledge from an auxiliary domain with sufficient data, and then use the learned knowledge to assist the learning of the target domain. In general, CDCF models can be classified into four types [8]: 1) Based on overlapping users; 2) Based on overlapping items; 3) Based on both overlapping users and items; and 4) Based on non-overlapping data. Most of existing methods [9]–[15] fall into the first three types, which rely on overlapping data as a bridge for knowledge transfer. Though effective, the assumptions of these methods that the overlapping data always exists and the raw interaction data can be fully shared across domains may not be realistic in some cases. For example, if the two different domains (websites) are from different companies, it is usually difficult to let them share user interaction data due to the constraint of company policy. Therefore, it poses strong demand for the research on the fourth type. But the lack of any overlapping data between domains also greatly increases the difficulty of knowledge transfer, which makes researches on this type relatively rare.

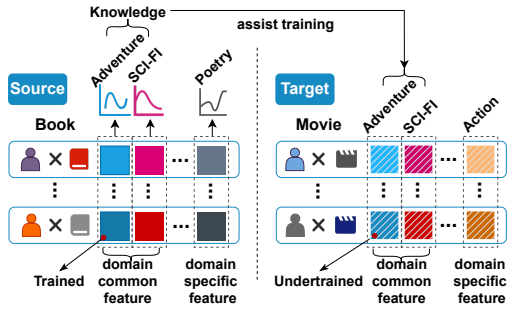


Fig. 1. An example illustrating knowledge transfer from source domain Book to target domain Movie.

As far as we know, Code Book Transfer (CBT) [16] is the first CDCF work that falls into the fourth type, whose main idea is transferring the cluster-level user-item rating patterns (also called codebook) across domains. Later, there are also some attempts on improving CBT [17]–[20]. However, these methods are limited by the inherent way of transferring cluster-level rating patterns, and have not tried to mine more complex relation with neural network. In addition, Cremonesi *et al.* [21] have shown that the improvement in CBT actually may not come from the transfer of the source domain information, which poses questions on the effectiveness of these methods.

To address these problems, we propose a novel CDCF framework named Interaction Distribution Transfer (IDT) for the scenario without any overlapping data. Fig. 1 presents an example to help understand our idea. Consider a non-overlapping cross-domain recommendation scenario with Book as the source domain and Movie as the target domain. Suppose that each genre of book or movie is regarded as the

† Jianjun Li is the corresponding author

explicit representation of a dimension in the latent interaction feature. It can be observed that *Adventure* and *SCI-FI* are the common genres of the two domains, while *Poetry* and *Action* are domain specific genres. In the latent feature space, we refer to them as domain common features and domain specific features, respectively. A horizontal rounded rectangle with blue line in Fig. 1 represents an interaction behavior. The colored squares show the preference values of a user on the multi-dimension latent feature, and each dimension of the latent feature implies a distribution correspond to a genre, which is marked as a vertical rectangle with dotted line. We believe that there is some similarity between the distributions of domain common features in these two domains. We notice that [22] makes similar assumption as ours, which assumes that user and item latent vectors in different domains can be generated from a common Gaussian distribution, and experimentally verifies the assumption. Thus, the main goal of IDT is to transfer more accurate distribution of domain common features learned from the source domain to guide feature learning in the target domain. To this end, we also design a mask operation to distinguish domain common features and domain specific features. Based on two typical CF models, BPRMF and LightGCN, we instantiate our framework IDT to form two methods, BPRMF-IDT and LightGCN-IDT, and show their effectiveness experimentally. In sum, the main contributions of this paper are as follows,

- We propose a cross-domain recommendation framework named IDT suitable for the scenario without any overlapping data. Specifically, IDT transfers the distribution of domain common features, a high-level abstracted knowledge extracted by CF base models with strong feature extraction capability. In this way, the risk of user privacy leakage can be avoided as compared to existing methods that utilize user behavior data.
- We conduct extensive experiments on two recommendation scenarios with three benchmark datasets. The results demonstrate the superior performance of IDT. Moreover, we also design controlled experiments according to [21] to verify the effectiveness of our methods in transferring source domain knowledge.

II. Related Work

In this section, we briefl y review some researches that are closely related to our work.

A. Learning-based CF Methods in Single Domain

CF is a commonly used technology in modern recommendation systems. Early CF models, such as matrix factorization (MF), map user (or Item) ID to the embedded space, and establish the matching relationship between the user and item through inner product. These models are comparatively easy to implement, but lack nonlinear feature extraction ability, which limits their performance. Later, NeuMF [2] and DMF [4] propose to model projection and matching functions utilizing neural network. More recently, attention mechanisms are introduced to automatically learn the importance of each historical interaction, such as in ACF [23] and DeepICF [24] Inspired by the development of graph neural networks (GNN), there are some efforts on exploiting user-item interaction graph to infer users preference. For example, GCMC [5] updates the model parameters in multiple rating levels by the standard of GNN. SpectralCF [6] designs a new spectral convolution model, to explore deep connections in the spectral domain between users and items. NGCF [25] injects the collaborative signal into the embedding process via propagating embeddings on it. LightGCN [7] simplifies and elevates NGCF by removing some unnecessary and harmful operations. Corso et al. [26] prove the need for multiple aggregators mathematically and propose PNA architecture combining them with degree-scalers. Huang et al. [27] present MixGCF, a general negative sampling plugin to synthesize hard negatives instead of sampling raw negatives from data. However, data sparsity remains a limiting factor for the accuracy of existing models, especially with the increase of model complexity.

B. Cross-Domain Collaborative Filtering Methods

As an effective technique to address the data sparsity problem, the research on Cross Domain Collaborative Filtering (CDCF) is increasing. Among them, research [12], [28], [29] on CDCF with overlapping users develops the fastest. Early, CMF [9] jointly factorizes the rating matrix from two domains by sharing user latent factors. CoNet [30] completes the transfer of interaction features between domains through cross-mapping. With the development of Domain adaption (DA) technique, DARec [31] models the difference between two domains by combining DA instead of cross-mapping in CoNet. BiTGCF [32] further distinguishing users common features and specific features between two domains on the graph structure. Additionally, there have been studies [33] that prioritize the protection of user privacy and opt for reliance on overlapping items. However, the existence of overlapping data is still a must for these methods.

On the other hand, the research on CDCF with non-overlapping data is relatively rare. Li et al. [16] propose a codebook transfer (CBT) model to transfer the cluster-level rating patterns compressed from the source domain data to the target domain. Gao et al. [18] further consider domain specific features to form the CLFM algorithm. Later, many variants based on CBT have been proposed [17], [34], but the full source domain matrix constraint is not relaxed until He et al. [35] propose the incomplete orthogonal nonnegative matrix tri-factorization (IONMTF) method. Recently, mixed heterogeneous factorization (MHF) [36] is proposed, which accounts for domain heterogeneity comprehensively. However, these methods still face the three major limitations mentioned earlier.

III. Problem Definition

Given two domains, one source domain $\mathcal{D}_s$ and one target domain $\mathcal{D}_t$. Let $(u_s, i_s, r^s_{ui}) \in \mathcal{D}_s$ denote the interaction sample of the source domain, where u_s, i_s, and r^s_{ui} are users, items, and interactions in $\mathcal{D}_s$ respectively. Similarly, let

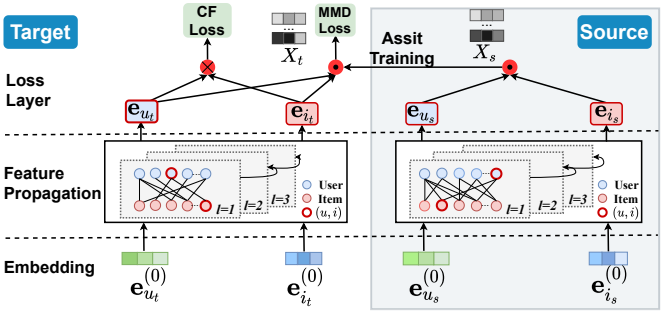


Fig. 2. The Architecture of IDT. The source domain provides the trained X_s to the target domain.

(u_t, i_t, r^t_{ui}) denote the interaction sample in $\mathcal{D}_t$. U_s and U_t are the sets of users in the corresponding domain. Since we consider non-overlap in this work, $U_s \cap U_t = \emptyset$ and $I_s \cap I_t = \emptyset$. In learning based CF methods, users and items are mapped to the latent feature by learnable parameters. We believe that $\mathcal{D}_s$ and $\mathcal{D}_t$, after being decoupled in high dimensions, have partially common features–domain common features, and partially specific features–domain specific features. The purpose of our cross-domain transfer framework IDT is to use the knowledge (the distribution on domain common feature) learned from $\mathcal{D}_s$ to assist the learning of model in $\mathcal{D}_t$, so as to improve the recommendation performance of $\mathcal{D}_t$.

IV. Methodology

The overall structure of our framework is shown in Figure 2, which mainly includes two modules: 1) The single-domain recommendation module acting on the two domains respectively, which serves as the base model in our framework. 2) The distribution transfer module, which is responsible for extracting knowledge (domain common distribution) from the source domain to assist the training in the target domain, is the key module in our framework.

A. Base Models

Existing CF models can be roughly classified into two categories [3]: representation learning-based CF methods (RL-CF) and matching function learning-based CF methods (ML-CF). RL-CF methods are committed to learning more accurate feature representations, and often use simple matching functions, such as inner product or cosine similarity, to calculate the interaction score. Our framework IDT is applicable to RL-CF models. In order to better verify its effect, we choose two representative RL-CF methods as the base models: BPRMF [1] and LightGCN [7]. The architecture of our framework is shown in Fig. 2. Note that since we use the same architecture for source and target domains, we use u (i) to denote u_s (i_s) and u_t (i_t) collectively if no confusion arises.

Embedding: This module maps the IDs of user u and item i into a dense feature space. Specifically,

$$\begin{aligned} \mathbf{e}_u^{(0)} &= \mathbf{P}^\top \mathbf{z}_u \in \mathbb{R}^d \\ \mathbf{e}_i^{(0)} &= \mathbf{Q}^\top \mathbf{z}_i \in \mathbb{R}^d \end{aligned} \tag{1}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are learnable parameter matrices of u and i, respectively, and are also often referred to as embedding matrices. d denotes the embedding size, and $\mathbf{z}_u$ and $\mathbf{z}_i$ represent one-hot (or multi-hot in some other RL-CF methods) encoding for u and i. For ID embedding, this module can also be seen as a look-up table building by a parameter matrix, which will be optimized end-to-end.

Feature Propagation Layer: This layer captures the non-linearity of features obtained by the embedding layer. Note that if the base model is BPRMF, then such a layer is not included. Therefore, we only detail the feature propagation rules in LightGCN here. As shown in Fig. 2, the interaction data between users and items can form a bipartite graph $\mathcal{G}$. The main idea of graph convolution is to integrate the information from neighbors to improve the features of the current node. By stacking multi-layer graph convolution, deeper relationships between nodes can be digged out by multi-layer feature propagation. The feature propagation layer in LightGCN can be abstracted as,

$$\begin{aligned} \mathbf{e}_u^{(l+1)} &= \sum\nolimits_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(l)} \\ \mathbf{e}_i^{(l+1)} &= \sum\nolimits_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(l)} \end{aligned} \tag{2}$$

where $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ respectively denote the refined features of user u and item i after l layers propagation, $\mathcal{N}_u$ and $\mathcal{N}_i$ are the sets of first-hop neighbors of user u and item i, and $\frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}$ indicates the discount factor on the path $u \leftrightarrow i$, which uses symmetric regularization about the degree of two nodes to reduce the influence of active users or popular items in neighbors on the features of the current node.

Output Layer: This layer predicts the interaction probability between a given user and an item by a matching function based on their final features. Specifically,

$$\hat{r}_{ui} = \mathbf{e}_u^\top \mathbf{e}_i \tag{3}$$

where $\mathbf{e}_u$ $(\mathbf{e}_i)$ represents the final features of u (i). Note for BPRMF, $\mathbf{e}_u = \mathbf{e}_u^{(0)}$. For LightGCN, considering that the features from different layers can complement each other, $\mathbf{e}_u = \frac{1}{(L+1)} \sum_{l=0}^{L} \mathbf{e}_u^{(l)}$, and $\mathbf{e}_i$ can be obtained similarly.

B. Distribution Transfer Module

Acquisition of interaction distribution: In our base model BPRMF, u and i are mapped to $\mathbf{e}_u$ and $\mathbf{e}_i$, respectively, which represent the preference (or attribution) mapping of users and items in a common latent space. We can capture the interactive feature $\mathbf{x}$ in the d-dimensional latent space by utilizing an interaction function $f(\mathbf{e}_u, \mathbf{e}_i) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$. There are many interaction functions, such as element-wise product or nonlinear neural network layers. Here, we use the element-wise product $\odot$ as the interaction function, which is simple yet effective, and moreover, does not introduce extra parameters,

$$\mathbf{x} = f(\mathbf{e}_u, \mathbf{e}_i) = \mathbf{e}_u \odot \mathbf{e}_i \tag{4}$$

By compressing the information of all interaction instances into a multi-dimensional feature space, we can obtain a set that contains all interaction features as $\mathcal{X} = \{\mathbf{x}|(u,i) \in \mathcal{D}\}$, which characterizes a multi-dimensional interaction feature distribution (MDIFD) in the current domain. We use $\mathcal{X}_s$ and $\mathcal{X}_t$ to represent the interaction feature set in $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively. The MDIFD $\mathcal{X}_s$ learned from $\mathcal{D}_s$ is the knowledge we plan to transfer to $\mathcal{D}_t$.

Compared with BPRMF, LightGCN add multiple feature propagation layers. In each layer, the nodes are perfected through the information integration of neighbor nodes, which is also the reason why LightGCN can mine higher-order relationships between nodes. However, the adjacency matrices of the two domains, indicating the connected edges in the U-I graph, are completely different. Therefore, the high-order interaction features obtained by LightGCN are more specific to the task of the current domain, which may limit the effect of knowledge transfer. This point has also been verified in the study [37]. To address this issue, we obtain the interaction feature $\mathbf{x}$ in LightGCN by explicitly considering low-order features. Specifically,

$$\mathbf{x} = f(\mathbf{e}_u, \mathbf{e}_i) + \epsilon f(\mathbf{e}_u^{(0)}, \mathbf{e}_i^{(0)}) = \mathbf{e}_u \odot \mathbf{e}_i + \epsilon(\mathbf{e}_u^{(0)} \odot \mathbf{e}_i^{(0)}) \quad (5)$$

where ϵ is a hyper-parameter, controlling the intensity of the low-order interactive features.

Transfer of MDIFD: The goal of cross-domain transfer is to use $\mathcal{X}_s$ to guide and assist the model learning in $\mathcal{D}_t$. Note the target domain itself will also produce $\mathcal{X}_t$ during training. Consequently, we have two expectations for model training in $\mathcal{D}_t$: 1) complete the CF task by using its own interaction data; and 2) adjust $\mathcal{X}_t$ to make it close to $\mathcal{X}_s$, which is a reasonable way to reflect domain similarities based on the assumptions we mentioned above. For the first expectation, we construct a conventional CF loss L_{cf}. For the second expectation, we define a new transfer loss L_{tr} to maintain the similarity between $\mathcal{X}_s$ and $\mathcal{X}_t$. There are many ways to measure the similarity between two distributions, such as the well-known KL divergence and its symmetric form, JS divergence. However, they become inapplicable or illogical here, because two distributions from different domains are not encoded against the same data. Here, we employ Maximum Mean discrepancies (MMD) distance [38], a commonly used metric in domain adaptation, as similarity metrics. Specifically, MMD is defined as the distance between mean values of the two distributions in reproducing kernel Hilbert space (RKHS). Formally,

$$\begin{aligned}\mathrm{MMD}_k^2(\mathcal{X}_s, \mathcal{X}_t) &= ||\frac{1}{|\mathcal{X}_s|}\sum_{\mathbf{x}_s \in \mathcal{X}_s}\phi(\mathbf{x}_s) - \frac{1}{|\mathcal{X}_t|}\sum_{\mathbf{x}_t \in \mathcal{X}_t}\phi(\mathbf{x}_t)||_{\mathcal{H}_k}^2 \\ &= \frac{1}{|\mathcal{X}_s|^2}\sum_{\mathbf{x}_s \in \mathcal{X}_s}\sum_{\mathbf{x}'_s \in \mathcal{X}_s} k(\mathbf{x}_s, \mathbf{x}'_s) + \frac{1}{|\mathcal{X}_t|^2}\sum_{\mathbf{x}_t \in X_t}\sum_{\mathbf{x}'_t \in \mathcal{X}_t} k(\mathbf{x}_t, \mathbf{x}'_t) \\ &\quad - \frac{2}{|\mathcal{X}_s||\mathcal{X}_t|}\sum_{\mathbf{x}_s \in \mathcal{X}_s}\sum_{\mathbf{x}_t \in \mathcal{X}_t} k(\mathbf{x}_s, \mathbf{x}_t)\end{aligned} \quad (6)$$

where ϕ is mapping function, transforming the data to RKHS $\mathcal{H}_k$ endowed with kernel k. The relationship between ϕ and kernel k is $k(\mathbf{x}_s, \mathbf{x}_t) = \langle\phi(\mathbf{x}_s), \phi(\mathbf{x}_t)\rangle$, where $\langle\cdot,\cdot\rangle$ denotes the inner product. Multi-kernel MMD [39] is usually applied to reduce the huge influence of the selection of k on the result by defining k as the convex combination of m Gaussian kernels $\{k_a(\mathbf{x}_s, \mathbf{x}_t) = e^{-||\mathbf{x}_s - \mathbf{x}_t||^2/\gamma_a}\}$, where γ_a is a varying bandwidth. We also adopt multi-kernel MMD and derive multi-kernel k by $k = \sum_{a=1}^{m}\beta_a k_a$, where β_a is an coefficient satisfying $\sum_{a=1}^{m}\beta_a = 1, \beta_a \geqslant 0, \forall a$. In order to simplify calculation, we set $\beta_a = \frac{1}{m}$ for any a and $m = 5$ in this work. In the actual optimization process, we convert it into matrix form by constructing kernel matrix for easy implementation. The detail derivation and formula can be found in subsection 3.3 of [**?**]. By minimizing the MMD loss between two domains, we can purposefully guide the learning of interaction distribution in $\mathcal{D}_t$ with source domain information as prior knowledge.

Mask domain specific features: When transferring $\mathcal{X}_s$, we need to eliminate the influence of domain specific features. One possible way to achieve this goal is masking these features when calculating MMD loss. It is well recognized that the existence of entanglement between features due to latent space mapping makes the mask operation rather difficult. In this work, for better model efficiency, we do not follow the typical principle to first adopt complex disentanglement operation and then mask, but use a sub-optimal but efficient algorithm to eliminate the influence of entanglement between features on the mask operation. Specifically,

- We define our mask operation $R(\mathcal{X}, \{o\})$ as setting the $\{o\}$-th dimension values of all $\mathbf{x}$ in $\mathcal{X}$ to 0. This operation can be seen as removing the effect of $\{o\}$-th dimension values on the similarity between the overall distributions from all dimensions. The formula of the Gaussian kernel function $k(\mathbf{x}_s, \mathbf{x}_t) = e^{-||\mathbf{x}_s - \mathbf{x}_t||^2/\gamma_a}$ guarantees the rationality of this statement[1].
- By traversing the set $\{o|o \in [1, \cdots, d]\}$, we can calculate and obtain a set $\mathcal{S} = \{y_o|y_o = \mathrm{MMD}^2(R(X_s, \{o\}), R(X_t, \{o\}))\}$.
- We use the dimensions corresponding to K (K is a hyper-parameter) smallest values from $\mathcal{S}$ to form a set $\mathcal{M} = \{o|y_o \in top\text{-}K(\mathcal{S})^2\}$, whose elements are specified as domain-specific features. Take K=1 as an example, the principle of our mask operation is selecting a dimension such that removing it greatly reduces the MMD distance between the two domains. We believe that the distribution differences between two similar domains on specific feature should be much larger than on common feature. Therefore, if the MMD distance between distributions becomes the smallest in set $\mathcal{S}$ only because the o-th dimension feature is removed, then it has a high probability of being domain-specific features with large distribution differences.

[1]For convenience, the single-kernel kernel function is listed here, and the same conclusion can be obtained for multi-kernel kernel function mentioned above.

[2]Represent the top-K smallest values from set $\mathcal{S}$.

Our way of computing the MMD distance for all features except one specified dimension every time preserves the entanglement among the remaining $d-1$ dimensions, which eliminate the influence of entanglement between features on mask operation as much as possible.

After obtaining a relatively mature MDIFD by a certain steps of training in the target domain (e.g., 100 epochs), we can use the mask operation $R(\mathcal{X}_s, \mathcal{M})$ and $R(\mathcal{X}_t, \mathcal{M})$ to mask the two MDIFDs to exclude the influence of domain-specific features in both domains. Subsequently, the transfer loss L_{tr} can be expressed as $L_{tr} = \mathrm{MMD}^2(R(\mathcal{X}_s, \mathcal{M}), R(\mathcal{X}_t, \mathcal{M}))$.

C. Model Training

Learning based on interactive data in the target domain is the main part of model training. In this paper, we consider a pair-wise CF loss function L_{cf}, which focuses on learning ranks, and leave the point-wise loss as future work. The BPR loss, which is the most commonly used pair-wise loss, is employed,

$$L_{cf}(\hat{r}^t_{ui}, \hat{r}^t_{uj}) = \sum_{(u_t, i_t, j_t) \in \mathcal{O}_t} -ln\sigma(\hat{r}^t_{ui} - \hat{r}^t_{uj}) + \gamma||\Theta||_2^2 \quad (7)$$

where i_t is the observed interactive item by u_t, j_t is the sample from un-interacted item, $\hat{r}^t_{ui}$ (or $\hat{r}^t_{uj}$) indicates the predicted scores of pair (u_t, i_t) (or (u_t, j_t)). $\Theta = \{\mathbf{P}, \mathbf{Q}\}$ denotes the learnable parameter set in our recommendation task, and γ is a hyper-parameter, controlling the $L2$ regularization strength.

L_{cf} in the target domain and L_{tr} from the transfer module are weighted and summed together to constrain the training of the target domain. The joint loss is then defined as,

$$L_{joint} = L_{cf} + \lambda * L_{tr} \quad (8)$$

where λ is a hyper-parameter, controlling the intensity of knowledge transfer. We adopt mini-batch Adam to optimize the model and update the parameters, therefore, $\mathcal{X}_t = \{\mathbf{x_t}|(u_t, i_t) \in \mathcal{D}_t^{batch}\}$ ($\mathcal{D}_t^{batch}$ denotes the sample batch). It is worth noting that $\mathcal{X}_s$ is obtained from the trained source domain and can be reused by other target domains. In order to facilitate the training of the model, in the actual experiment process, we randomly sampled 1000 samples $\mathbf{x}_s$ from $\mathcal{X}_s$ each time as the source domain MDIFD.

D. Model Extension

Based on the above analysis, we can find that the guidance from source domain is the main factor to improve the performance of the target domain. Then, it poses a question whether utilizing multiple source domains can help achieve better results or make the performance more stable? To answer this question, we conduct a model extension by applying our model to a multi-source domain scenario. Specifically, given N source domains, denoted by $\mathcal{D}_i$ where $i \in \{1, \cdots, N\}$, and $\mathcal{X}_{s_i}$ is the trained MDIFD of the corresponding source domain, one thing needs to be considered is the allocation for the proportion (or weight) of the source domains. Intuitively, we expect that the closer the source and target domains are, the higher the weight will be. However, the similarity between source and target domains is difficult to measure. Here, we abstract and simplify it to L_{tr}. The lower the $L_{tr}(\mathcal{X}_{s_i}, \mathcal{X}_t)$, the more "similar" the two domains $\mathcal{D}_i$ and $\mathcal{D}_t$ are. On this basis, we form the loss function with multi-source domains,

$$L_{join} = L_{cf}(\hat{r}^t_{ui}, \hat{r}^t_{uj}) + \lambda \sum_i \beta_i * L_{tr}(\mathcal{X}_t, \mathcal{X}_{s_i}) \quad (9)$$

$$\beta_i = \frac{1}{N-1}(1 - \frac{L_{tr}(\mathcal{X}_t, \mathcal{X}_{s_i})}{\sum_j L_{tr}(\mathcal{X}_t, \mathcal{X}_{s_j})}) \quad (10)$$

Note when L_{tr} in all source domains are equal, $L_{tr}(\mathcal{X}_{s_i}, \mathcal{X}_t) = L_{mean} = \frac{1}{N}\sum_j^N L_{tr}(\mathcal{X}_t, \mathcal{X}_{s_j})$, the weighted coefficient $\beta_i = 1/N$, and when $L_{tr}(\mathcal{X}_{s_i}, \mathcal{X}_t) > L_{mean}$, $\beta_i < 1/N$. According to Equation (10), the weights of multi-source domains can be automatically adjusted according to the "similarity" between them and the target domain during training.

TABLE I
STATISTICS OF DATASETS

Type	Dataset	#User	#Item	#Interaction	Sparsity
Source	Amusic	844	9,714	37,041	99.55%
	Avideo	1913	15,767	73,416	99.76%
Target	BookCross	4,969	43,479	504,592	99.77%
	Amovie	15,067	69,629	877,736	99.92%
Overlapped	Amovie_V	6,064	39,154	198,647	99.91%

V. EXPERIMENT

A. Experimental setup

1) Dataset: We evaluate our proposed model in two recommendation scenarios: 1) Without any overlapping, in which we take Digital Music (**Amusic** for short)[3] and Video Games (**Avideo** for short)[3] as the source domain, and **BookCross**[4] and Movies and TV (**Amovie** for short)[3] as the target domains. Those datasets are filtered to retain users with interactions greater than 20 and items with interactions greater than 5. To guarantee a non-overlapping cross-domain scenario, an independent remapping of source domain and target domain is carried out. 2) With user-overlapping, in which we take **Avideo** as the source domain, and then extract the common users between **Avideo** and **Amovie** to derive a sub dataset of **Amovie**, denoted by **Amovie_V**, as the target domain. For both **Avideo** and **Amovie_V**, we retain users and items with interactions greater than 5 to keep more interactive data compared to the previous data processing. Table I summarizes the statistics of the datasets we used.

2) Evaluation protocol: We adopt the leave-one-out evaluation method, which has been widely used in the literature [2], [4], [30], [32]. Specifically, we randomly select 99 items from each user's non-interacted items to form negative samples. The recommendation model sorts these 100 items by prediction scores and output top-N items. We use the commonly used Hit Ratio (**HR**), Normalized Discounted Cumulative Gain (**NDCG**) and Mean Average Precision (**MAP**) to evaluate the ranking performance. For all the three measures, we truncate the ranked list at $\{5, 10\}$.

[3]http://jmcauley.ucsd.edu/data/amazon/

[4]http://www2.informatik.uni-freiburg.de/ cziegler/BX/

TABLE II

PERFORMANCE COMPARISON IN TERMS OF HR, NDCG AND MAP IN THREE DATASET. (·) INDICATES SOURCE DOMAIN DATASET WE USED. - INDICATES THE MODEL IS NOT APPLICABLE TO THE CURRENT SCENARIO. THE BEST PERFORMANCE IS IN BOLDFACE AND SECOND-BEST UNDERLINED. ALL OF OUR RESULTS ARE STATISTICALLY SIGNIFICANT WITH $p < 0.05$ UNDER T-TEST. * INDICATES THE VALUES ARE TO THE BASE MODELS.

Methods	BookCross						Amovie						Amovie_V					
	top-5			top-10			top-5			top-10			top-5			top-10		
	HR	NDCG	MAP	HR	NDCG	MAP	HR	NDCG	MAP	HR	NDCG	MAP	HR	NDCG	MAP	HR	NDCG	MAP
CDAE	0.4918	0.3639	0.3216	0.6154	0.4035	0.3371	0.6794	0.5347	0.4866	0.7882	0.5696	0.5008	0.4976	0.3733	0.332	0.6118	0.4103	0.3473
NGCF	0.5016	0.3717	0.33	0.6277	0.4145	0.3477	0.6848	0.5446	0.4947	0.7922	0.5724	0.5091	0.4963	0.3745	0.3371	0.6079	0.4122	0.3492
CoNet	-	-	-	-	-	-	-	-	-	-	-	-	0.4726	0.3391	0.2956	0.5989	0.3791	0.3118
U-DARec	-	-	-	-	-	-	-	-	-	-	-	-	0.4363	0.3246	0.2876	0.5638	0.3621	0.3021
BiTGCF	-	-	-	-	-	-	-	-	-	-	-	-	**0.5312**	**0.4123**	**0.3727**	**0.6395**	**0.4474**	**0.3872**
BPRMF	0.4484	0.3317	0.2932	0.5721	0.3716	0.3096	0.6255	0.4814	0.4339	0.744	0.5211	0.4515	0.4411	0.3327	0.2971	0.5537	0.3645	0.3063
BPRMF-IRT	0.3576	0.2392	0.2005	0.5031	0.2856	0.2188	0.5014	0.3365	0.2825	0.6608	0.3874	0.3028	0.4203	0.2847	0.2405	0.5482	0.3234	0.2513
BPRMF-IDT(M)	0.4969	0.3699	0.3283	0.6148	0.4092	0.3464	0.6654	0.5226	0.4753	0.7734	0.5587	0.4913	-	-	-	-	-	-
BPRMF-IDT(V)	0.4905	0.3655	0.3248	0.6033	0.4031	0.341	0.668	0.5285	0.4813	0.7775	0.5627	0.4952	0.4554	0.3445	0.308	0.5594	0.375	0.3179
Improvement*	10.82%	11.52%	11.97%	7.46%	10.12%	11.89%	6.79%	9.78%	10.92%	4.50%	7.98%	9.68%	3.24%	3.55%	3.67%	1.03%	2.88%	3.79%
LightGCN	0.5232	0.393	0.3503	0.6454	0.4254	0.3574	0.6987	0.5549	0.5064	0.8029	0.588	0.5187	0.509	0.3924	0.3447	0.6182	0.4206	0.3548
LightGCN-IRT	0.4923	0.3593	0.3154	0.6173	0.4006	0.3336	0.6764	0.5233	0.4727	0.7872	0.5622	0.4927	0.5000	0.3701	0.3275	0.612	0.4047	0.3394
LightGCN-IDT(M)	0.5371	0.4045	0.3609	0.6592	0.4417	0.3753	0.7098	0.5607	0.5112	0.8126	0.5928	0.5234	-	-	-	-	-	-
LightGCN-IDT(V)	**0.5408**	**0.4057**	**0.3612**	**0.6593**	**0.4431**	**0.3759**	**0.7102**	**0.5656**	**0.5177**	**0.8134**	**0.598**	**0.5301**	0.5221	0.4045	0.367	0.6261	0.4398	0.382
Improvement*	3.36%	3.23%	3.11%	2.15%	4.16%	5.18%	1.65%	1.93%	2.23%	1.31%	1.70%	2.20%	2.57%	3.08%	6.47%	1.28%	4.56%	7.67%

3) Compared methods: We compare our two instantiated methods BPRMF-IDT and LightGCN-IDT with several representative methods. Considering that the CBT-based approaches are not suitable for implicit recommendation and pose unaffordable computation cost due to the codebook construction operation, we do not include them as competitors.

Baselines in Single domain:

- **BPRMF**[5] [1] optimizes MF with the BPR loss.
- **CDAE**[6] [40] extends Denoising Auto-Encoder for item recommendation.
- **NGCF**[7] [25] explores the high-order relations between users and items by graph convolutional network.
- **LightGCN**[5] [7] simplifies the aggregation function in NGCF, resulting in better recommendation performance.

Baselines in Cross domain:

- **CoNet**[8] [30] is a deep model, which connects hidden layers in two base MLP networks by cross mapping.
- **U-DARec**[9] [31] is a domain adaption-based model that shares rating patterns of the same user in different domains after encoding with autoencoder.
- **BiTGCF**[10] [32] is a graph-based model, which realizes two-way transfer of knowledge across domains by using common users as bridge.

To make a fair comparison, we change the losses of all methods to BPR. Moreover, to avoid the fake transfer effect as in CBT-based methods pointed by Cremonesi et al. [21], we also conduct a set of controlled experiments by replacing X_s with X_{random}, which is generated by a randomly initialized embedding matrix. We call the transfer method with X_{random} as **IRT**, and instantiate it to obtain two methods, **BPRMF-IRT** and **LightGCN-IRT**, for comparison.

[5]https://github.com/gusye1234/LightGCN-PyTorch
[6]https://github.com/jasonyaw/CDAE
[7]https://github.com/huangtinglin/NGCF-PyTorch
[8]http://home.cse.ust.hk/ghuac/
[9]https://github.com/Yu-Fangxu/DARec
[10]https://github.com/sunshinelium/Bi-TGCF

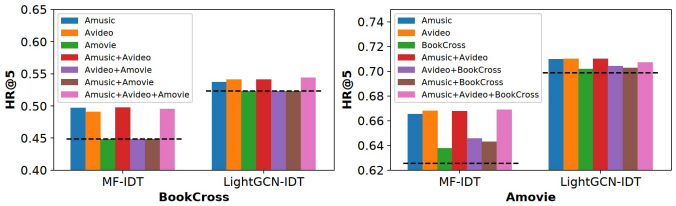
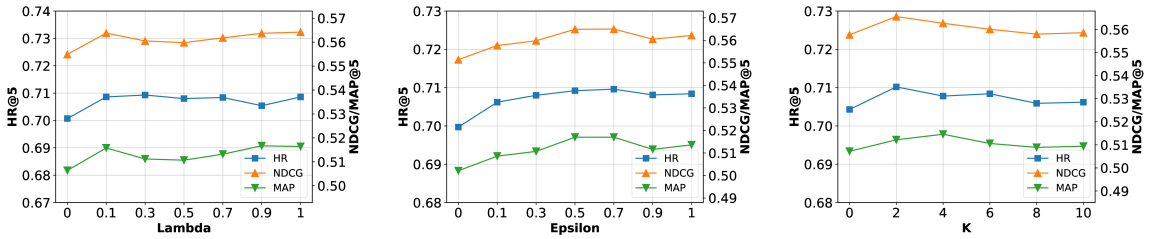


Fig. 3. The performance of IDT with multiple source domains on target domains BookCross and Amoive, respectively. Legends show the dataset of source domain. The black dotted line shows the scores of the base model.

4) Parameter settings: We set the common parameters embedding size as 64, learning rate as 0.001, the same as most compared methods [7], [25]. Batch size is set as 1048, except for CDAE and DARec, in which the number is 256. Regularization coefficient γ is varied within the range $[1e-3, 1e-4, 1e-5]$. Loss function is optimized with mini-batch Adam. Other model-specific parameters are tuned to the best fit according to the values recommended in their respective papers. Unless otherwise specified, we report the performance of IDT with the following default settings: $\epsilon = 0.3, \lambda = 0.5$ and $K = 2$. All our code will be made publicly available.

B. Performance comparison

Table II shows the summarized results on the two target datasets, from which we have the following key observations:(1) The performance improvements of BPRMF-IDT over BPRMF as well as LightGCN-IDT over LightGCN on Bookcross and AMovie demonstrate the effectiveness of our IDT framework, indicating that transferring domain common feature distributions indeed can help achieve better recommendation performance. Moreover, the promotion of IDT on LightGCN is not significant as compared to that on BPRMF. The reason might be that the latent features learned by LightGCN is more accurate and specific, which makes it more difficult to achieve cross-domain transfer. (2) The two IRT methods performs significantly worse than their IDT counterparts, and even worse than corresponding base models. This may be due



(a) value of λ (b) value of ϵ (c) value of K

Fig. 4. Impact factor curves of LightGCN-IDT on $Avideo \rightarrow Amovie$ task.

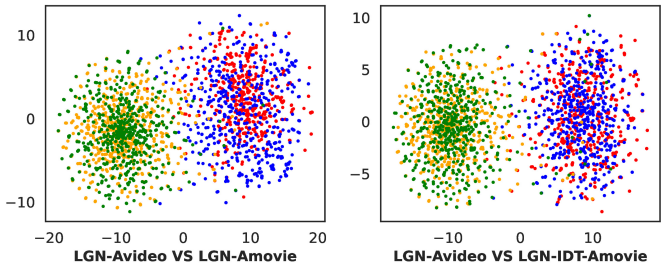


Fig. 5. Visual presentation about IDT on $Avideo \rightarrow Amovie$ task.

to the occurrence of negative transfer when transferring the random matrix without any knowledge. This result also verifies that IDT can effectively transfer knowledge from the source domain. (3) LightGCN-IDT performs better than CoNet, U-DARec and on Amovie_V, indicating that the introduction of graph learning further improves the knowledge extraction ability of cross-domain model. It can also be observed that LightGCN-IDT performs worse than BiTGCF. The reason might be that BiTGCF adopts the way of joint training, and the mutual improvement on both sides in joint training further strengthen its advantages. However, BiTGCF also requires the raw data from both domains, which is often unaccessible due to the risk of privacy leakage. Although the performance is slightly inferior to that of BiTGCF, LightGCN-IDT still achieves the second best result, and does not require any overlapping data.

C. Detail analysis about IDT

1) Performance with multiple source domains: Obviously, the selection of source domains may have a significant impact on the performance of IDT. Fig. 3 shows the HR@5 performance (NDCG and MAP have the same performance trends) of two instantiated models of IDT with one or multiple source domains. As can be seen, with Avideo or Amusic as single source domain, IDT achieves the best performance, while IDT's performance on the $Amovie \rightarrow BookCross$ task is the worst, and its performance on the $BookCross \rightarrow Amovie$ task is only slightly better. The results are basically consistent with the sparsity of the data sets, demonstrating the importance of density in modeling interaction features. On the other hand, IDT's performance with two source domains usually fall in between the performances of using the two single source domains individually. A similar trend can be observed for the case with three source domains. The reason for this might be that interaction features extracted under the guidance of multiple source domains are usually more common but less frequent, especially when the similarity and density of multiple source domains and target domains are significantly different. On the other hand, however, this phenomenon has the advantage of being more stable and less susceptible to source domain selection sensitivity.

2) Impact factor analysis: In order to figure out the influences of hyper-parameters on IDT, we discuss three important hyper-parameters: λ in (8), which controls the intensity of knowledge transfer; ϵ in (5), which controls the weight of low-order features of MDIFD when transferring on LightGCN; and K, which is the number of domain-specific feature selected from MDIFD. We show their influences on the performance in Fig. 4. Due to space concern, only the Top-5 performance of LightGCN-IDT on Amovie is presented.

From Fig. 4(a), it can be found that the model performs the worst when $\lambda = 0$, possibly due to the lack of assistance from the source domain. When λ ranges from 0.1 to 1, the constraints imposed by L_{tr} takes effect and the model achieves significantly better performance. In sum, the existence of L_{tr} has more prominent influence than the variation of λ.

From Fig. 4(b), it can be observed that the model performs the best when $\epsilon = 0.7$, and performs the worst when $\epsilon = 0$. The reason might be that the higher-order relations between users and items, captured by the base model LightGCN, specify the interaction relationship of the domain and may affect the subsequent transfer. Therefore, appropriately increasing the proportion of low-order features of MDIFD actually will lead to performance improvement.

From Fig. 4(c), it can be seen that the model performs the best when K is 2 or 4. With the increase of K, the performance first decreases and then tends to be stable. This indicates that properly relaxation of the constraint L_{tr} by increasing the number of domain specific features K can improve the performance, but too much increment will also cause under-utilization of knowledge transfer from the source domain.

3) Visual presentation about IDT: To further demonstrate the effectiveness of IDT on domain adaptation, we plot the t-SNE in Fig. 5 to visualize the learned multi-dimensional interaction feature distribution (MDIFD) on $Avideo \rightarrow Amovie$ task, using LightGCN (abbreviated as LGN in the Figure) as the base model. Red and orange points in the figure represent positive and negative samples from the source domain Avideo,

while blue and green points represent positive and negative samples from the target domain Amovie. It can be observed that when the IDT module is included, the distributions of the target and source domains become more similar, and the distinction between positive and negative points becomes more clear. This demonstrates the effectiveness of IDT, showing that the IDT module does, in fact, utilize the source distribution.

VI. Conclusion

In this paper, we propose a novel cross-domain collaborative filtering framework IDT for Top-N recommendation, which does not require any data overlapping between domains. Based on the assumption that there are partially shared latent features in similar domains, we extract the distribution on domain-common features from the source domain to assist the model learning of the target domain. A weight operation is used to balance and coordinate the knowledge transfer loss with the CF loss in the target domain. This process does not involve any raw data exchange between the two domains, and hence can avoid leaking user privacy effectively. Extensive experiments have been conducted, and remarkable performance improvements on several benchmark datasets demonstrate the effectiveness of our proposed framework.

References

[1] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *Proc. of UAI*, 2009, pp. 452–461.

[2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. of WWW*, 2017, pp. 173–182.

[3] Z. Deng, L. Huang, C. Wang, J. Lai, and P. S. Yu, "Deepcf: A unified framework of representation learning and matching function learning in recommender system," in *Proceedings of AAAI*, 2019, pp. 61–68.

[4] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proceedings of IJCAI*, 2017, pp. 3203–3209.

[5] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *CoRR*, vol. abs/1706.02263, 2017.

[6] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proc. of ACM RecSys*, 2018, pp. 311–319.

[7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proc. of SIGIR*, 2020, pp. 639–648.

[8] I. Cantador, I. Fernández-Tobías, S. Berkovsky, and P. Cremonesi, "Cross-domain recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 919–959.

[9] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of SIGKDD*. ACM, 2008, pp. 650–658.

[10] B. Loni, Y. Shi, M. A. Larson, and A. Hanjalic, "Cross-domain collaborative filtering with factorization machines," in *Proc. of ECIR*, 2014, pp. 656–661.

[11] T. Man, H. Shen, X. Jin, and X. Cheng, "Cross-domain recommendation: An embedding and mapping approach," in *Proc. of IJCAI*, 2017, pp. 2464–2470.

[12] P. Li and A. Tuzhilin, "Ddtcdr: Deep dual transfer cross domain recommendation," in *Proc. of WSDM*. Houston, USA: ACM, 2020, pp. 331–339.

[13] H. Liu, L. Guo, P. Li, P. Zhao, and X. Wu, "Collaborative filtering with a deep adversarial and attention network for cross-domain recommendation," *Inf. Sci.*, vol. 565, pp. 370–389, 2021.

[14] X. Yu, Q. Peng, L. Xu, F. Jiang, J. Du, and D. Gong, "A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm," *Inf. Process. Manag.*, vol. 58, no. 6, p. 102691, 2021.

[15] X. Hao, Y. Liu, R. Xie, K. Ge, L. Tang, X. Zhang, and L. Lin, "Adversarial feature translation for multi-domain recommendation," in *proc. of KDD*, 2021, pp. 2964–2973.

[16] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction," in *Proc. of IJCAI*, 2009, pp. 2052–2057.

[17] O. Moreno, B. Shapira, L. Rokach, and G. Shani, "Talmud: transfer learning for multiple domains," in *Proc. of CIKM*, 2012, pp. 425–434.

[18] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo, "Cross-domain recommendation via cluster-level latent factor model," in *Proc. of ECML-PKDD*, 2013, pp. 161–176.

[19] M. He, J. Zhang, and S. Zhang, "ACTL: adaptive codebook transfer learning for cross-domain recommendation," *IEEE Access*, vol. 7, pp. 19 539–19 549, 2019.

[20] K. Shu, S. Wang, J. Tang, Y. Wang, and H. Liu, "Crossfire: Cross media joint friend and item recommendations," in *Proc. of WSDM*, 2018, pp. 522–530.

[21] P. Cremonesi and M. Quadrana, "Cross-domain recommendations without overlapping data: myth or reality?" in *Proc. of RecSys*, 2014, pp. 297–300.

[22] T. Iwata and K. Takeuchi, "Cross-domain recommendation without shared users or items by sharing latent vector distributions," in *Proc. of AISTATS*, vol. 38, 2015.

[23] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. of SIGIR*, 2017, pp. 335–344.

[24] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Trans. Inf. Syst.*, vol. 37, no. 3, pp. 33:1–33:25, 2019.

[25] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *Proc. of the 42nd SIGIR*, 2019, pp. 165–174.

[26] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Velickovic, "Principal neighbourhood aggregation for graph nets," in *Proc. of NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[27] T. Huang, Y. Dong, M. Ding, Z. Yang, W. Feng, X. Wang, and J. Tang, "Mixgcf: An improved training method for graph neural network-based recommender systems," in *Proc. of SIGKDD*, 2021, pp. 665–674.

[28] F. Zhu, Y. Wang, C. Chen, G. Liu, M. A. Orgun, and J. Wu, "A deep framework for cross-domain and cross-system recommendations," in *Proc. of IJCAI*, 2018, pp. 3711–3717.

[29] C. Zhao, C. Li, and C. Fu, "Cross-domain recommendation via preference propagation graphnet," in *Proc. of CIKM*, 2019, pp. 2165–2168.

[30] G. Hu, Y. Zhang, and Q. Yang, "Conet: Collaborative cross networks for cross-domain recommendation," in *Proc. of CIKM*, 2018, pp. 667–676.

[31] F. Yuan, L. Yao, and B. Benatallah, "Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns," in *Proceedings of IJCAI*, 2019, pp. 4227–4233.

[32] M. Liu, J. Li, G. Li, and P. Pan, "Cross domain recommendation via bi-directional transfer graph collaborative filtering networks," in *Proc. of CIKM*. ACM, 2020, pp. 885–894.

[33] C. Gao, X. Chen, F. Feng, K. Zhao, X. He, Y. Li, and D. Jin, "Cross-domain recommendation without sharing user-relevant data," in *WWW*. ACM, 2019, pp. 491–502.

[34] Y. Zang and X. Hu, "LKT-FM: A novel rating pattern transfer model for improving non-overlapping cross-domain collaborative filtering," in *Proc. of ECML PKDD*, vol. 10535. Springer, 2017, pp. 641–656.

[35] M. He, J. Zhang, P. Yang, and K. Yao, "Robust transfer learning for cross-domain collaborative filtering using multiple rating patterns approximation," in *Proceedings of WSDM*, 2018, pp. 225–233.

[36] T. Yu, J. Guo, W. Li, and M. Lu, "A mixed heterogeneous factorization model for non-overlapping cross-domain recommendation," *Decis. Support Syst.*, vol. 151, p. 113625, 2021.

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. of NeurIPS*, 2014, pp. 3320–3328.

[38] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010.

[39] A. Gretton, B. K. Sriperumbudur, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, and K. Fukumizu, "Optimal kernel choice for large-scale two-sample tests," in *Proceedings of NeurIPS*, 2012, pp. 1214–1222.

[40] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proc. of WSDM*. ACM, 2016, pp. 153–162.