Information Processing and Management 61 (2024) 103689

Contents lists available at ScienceDirect

Information Processing and Management

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

Decoupled domain-specific and domain-conditional representation learning for cross-domain recommendation

Yu Zhang [a], Zhiyong Cheng [a,*], Fan Liu [b], Xun Yang [c], Yuxin Peng [d]

[a] Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[b] School of Computing, National University of Singapore, Singapore, Singapore
[c] School of Information Science and Technology, University of Science and Technology of China, Hefei, China
[d] Wangxuan Institute of Computer Technology, Peking University, Beijing, China

ARTICLE INFO

Keywords:
Cross-domain recommendation
Collaborative filtering
GCN
Disentangled representation learning

ABSTRACT

Cross-domain recommendation (CDR) has become popular to alleviate the sparsity problem in target-domain recommendation by utilizing auxiliary domain knowledge. A basic assumption of CDR is that users have shared preferences across domains, but most existing CDR models do not distinguish between users' unique preferences and shared preferences. We propose a new CDR model, called DRLCDR, which adopts a variational bipartite graph encoder to learn domain-specific representations and domain-shared representations, respectively. To make the domain-shared representation learned from different domains similar, the domain-specific representations learned from one domain is used as conditional information to guide the domain-shared representations (which is also called domain-conditional representation in our model) in another domain. In addition, a bridge function loss is adopted to further encourage the proximity of domain-conditional representations in the embedding space. Experiments on four public datasets show that DRLCDR outperforms strong baselines, including the recent CDR method using disentangled learning, with an average improvement of 3.32% and 3.01% for HR and NDCG, respectively.

1. Introduction

Recommender systems (RSs) have been extensively deployed in various online information platforms with the aim of comprehending users' information needs and facilitating the discovery of desired items amidst the vast volume of available information. Among various recommendation techniques, collaborative filtering (CF) (Sarwar, Karypis, Konstan, & Riedl, 2001), as a simple and effective solution, has achieved great success in modeling user preferences by using historical user interaction data. Many CF-based recommendation methods have been proposed and gained ever-increasing progress on recommender accuracy from the early shallow models (i.e., MF (Koren, Bell, & Volinsky, 2009), PMF (Mnih & Salakhutdinov, 2007), WRMF (Hu, Koren, & Volinsky, 2008)) to current deep models (i.e., NeuMF (He et al., 2017), NGCF (Wang, He, Wang, Feng, & Chua, 2019), LightGCN (He et al., 2020)), with decades of development. Despite these advancements, CF-based models still face the challenge of data sparsity. The user-item interaction data required for training these models are often limited, resulting in a sparse user–item matrix. This sparsity issue can lead to a dramatic drop in performance and can make it challenging for the models to accurately capture user preferences and provide reliable recommendations.

* Corresponding author.
E-mail addresses: zhangyucs@outlook.com (Y. Zhang), jason.zy.cheng@gmail.com (Z. Cheng), liufancs@gmail.com (F. Liu), xyang21@ustc.edu.cn (X. Yang), pengyuxin@pku.edu.cn (Y. Peng).

https://doi.org/10.1016/j.ipm.2024.103689
Received 22 July 2023; Received in revised form 15 September 2023; Accepted 14 February 2024
Available online 22 February 2024

Researchers have explored various strategies to address the sparsity issue in recommendation systems, including utilizing side information such as user reviews (Catherine & Cohen, 2017; Cheng et al., 2018; McAuley & Leskovec, 2013) or item images (He & McAuley, 2016; Liu et al., 2022; McAuley, Targett, Shi, & Van Den Hengel, 2015; Yang, Wang, Dong, Dong, Wang, & Chua, 2022). The additional information provides new insights of items and users and thus can help learn user preferences and item features. However, utilizing this information requires extracting effective features from textual or visual data. From the perspective of exploiting user-item interaction data in CF, a direct approach is to collect more user-item interaction data from other sources to help better understand user preferences, which is the focus of cross-domain recommendation (CDR) (Wang, Ye, Ma, Li, & Zhuang, 2023; Yu et al., 2021; Zhu et al., 2022). CDR is a type of recommendation system that enhances the recommendation performance of a target domain by utilizing user-item interaction data from multiple domains or sources (Man, Shen, Jin, & Cheng, 2017; Zhu et al., 2021). It involves transferring knowledge from one domain to another to address data sparsity and enhance the accuracy of the recommendation system. In recent years, CDR has garnered significant research attention, and numerous methods have been reported in this field (Cao et al., 2022; Li & Tuzhilin, 2020). These methods can be broadly classified into two approaches based on the distinct approaches used to model the auxiliary and target domains. The first approach involves extending the classical single-domain recommendation approach. For example, CMF (Singh & Gordon, 2008) collaboratively factorizes the user–item interaction matrix of both the auxiliary and target domains, while sharing a common user representation. PPGN (Zhao, Li, & Fu, 2019) constructs a unified user–item bipartite graph in two domains, then performs graph convolution operations to model the higher-order interaction data on the unified graph. Methods in this approach focus on modeling data within individual domains and disregard the correlations between various domains (Man et al., 2017; Zhu, Chen, Wang, Liu, & Zheng, 2019). Consequently, these methods cannot effectively utilize cross-domain information, which restricts their overall performance. The other approach is to first learn user representations independently in each domain, and then adopt different strategies to transfer knowledge from the source domain to the target domain. For example, CoNet (Hu, Zhang, & Yang, 2018) utilizes a neural network for each domain, then transfers information between domains through a cross-connected network. BiTGCF (Liu, Li, Li, & Pan, 2020) adopts GCN to independently learn the embedding of each domain, and then subsequently incorporates the propagation layer to transfer knowledge bi-directionally across two domains. While this approach utilizes various transfer learning strategies to transfer abundant information from the auxiliary domain to the target domain, it falls short in eliminating irrelevant information (e.g., domain-specific preferences) that could cause negative transfer issues in cross-domain recommendations.

It is an acknowledged fact that users have diverse preferences for various items (Ma, Zhou, Cui, Yang, & Zhu, 2019; Wang et al., 2020). In CDR, a basic assumption is that users shared some common interests among the various domains, which are derived from users' distinct personality (Liu et al., 2020; Zhu et al., 2022). Besides the common interests, users also have some distinct preferences which are specific for different domains. For ease of presentation, we denote the common and unique interests of different domains by domain-shared and domain-specific preferences. Most existing CDR methods do not distinguish between domain-specific and domain-shared preferences when performing information transfer (Hu et al., 2018; Zhao et al., 2019; Zhu et al., 2021). Consequently, it is possible that unique preferences from auxiliary domains may be transferred to the target domain and negatively impacts the preference learning of users in target domain. Therefore, it is crucial to decouple the diverse preferences of users and provide relevant information for target domains based on different scenarios and needs to improve the performance of CDR.

Disentangled representation learning has recently attracted significant interest because of its ability to extract valuable knowledge from user interactions and disentangle it into representation vectors of varying dimensions or distributions. A recent work DisenCDR (Cao et al., 2022) applied this technique in CDR and has demonstrated its effectiveness. Specifically, DisenCDR disentangles domain-specific and domain-shared information by leveraging an exclusive regularizer, an informative regularizer and a variational bipartite graph encoder (Cao et al., 2022). The domain-shared representation is a combination of domain-specific information based on the number of items which users interact with in each domain as a ratio to the total number of items interacted within both domains. Despite achieving improved results, it is worth noting that since most user feedback is implicit (e.g., clicks, purchases) and observed user interactions could be noisy. For example, users may accidentally click on an item and subsequently realize they are not actually interested in it (Wang et al., 2020). This noise in user interactions can significantly influence the domain-shared information generated via taking control of the ratio to combine domain-specific information in both domains.

Motivated by the above observations, in this work, we propose a novel CDR model called decoupled domain-specific and domain-conditional representation learning for CDR (short for DRLCDR). Different from the previous work, our model first learns the domain-specific and domain-shared preferences separately within each domain, and then uses a disentangled loss function to further decouple the two types of preferences in both domains, as well as a bridge loss function to connect the domain-shared preferences learned in the two domains. In addition, to make the domain-shared preferences learn separately from both domains more similar, we treat the domain-specific preference from one domain as conditional information to guide the domain-shared preference in the other domain. In this way, the influence of user interaction noise in existing method (Cao et al., 2022) for domain-shared representation learning can be avoided. To achieve the goal, we adopt the variational bipartite graph encoder (Cao et al., 2022) to model data distributions and learn domain-specific representations in one domain, which are then used as the conditional representations to learn the domain-shared representations in the other domain. Extensive experiments on four real-world datasets demonstrate the effectiveness of our proposed approach. We compared our approach with several strong baselines, including the ones employing disentangled representation learning techniques. The results show that DRLCDR outperforms all competing methods. In addition, we carried out ablation studies and detailed analysis to investigate the validity of each component in our model. The main contributions of this paper are as follows:

- We propose a CDR method based on disentangled representation learning, called DRLCDR, which decouples user preferences into domain-specific and domain-conditional preferences and transfers the shared user preferences of users across domains, hence improving recommendation performance on both domains.

- In the design of our model, we utilize the variational bipartite graph encoder to decouple user preferences and employ a decouple loss to ensure mutual independence between the preferences. Furthermore, we adopt a bridge loss to achieve knowledge transfer between domains.
- We conducted extensive experiments on four real-world datasets to validate the generalizability and a comprehensive ablation study to evaluate the effectiveness of our model. The experimental results demonstrate the superiority of our proposed method over the state-of-the-art baselines.

2. Related work

2.1. CF-based methods

The simplicity and effectiveness of collaborative filtering models have contributed to their remarkable success in recommender systems. Matrix factorization (MF) (Koren et al., 2009), a classical approach in collaborative filtering methods, aims to learn user and item representations by minimizing the error between the reconstruction matrix and the user-item interaction matrix. Based on this simple and effective idea, various MF-based variants have been proposed, e.g., PMF (Mnih & Salakhutdinov, 2007) and WRMF (Hu et al., 2008). Due to its excellent feature extraction capabilities, deep learning has been a great success in various fields and has also been applied to recommender systems for modeling user representations (Zhang, Yao, Sun, & Tay, 2019). A typical example is NeuMF (He et al., 2017), which learns user and item features in complex interaction matrices through nonlinear neural networks.

Recently, graph convolutional networks (GCNs) are widely used in recommender systems (Berg, Kipf, & Welling, 2017; Najafabadi, Mohamed, & Onn, 2019; Tao et al., 2020; Wu, Zhong, & Ye, 2023; Yan et al., 2022) for their capability of leveraging high-order neighborhood information. NGCF (Wang et al., 2019) iteratively propagates node embedding over interaction graph and achieves excellent performance. To perform the GCN operation more efficiently, LightGCN (He et al., 2020) simplifies the model by removing the transformation function and nonlinear activation function from NGCF, which effectively improves recommendation performance. UltraGCN (Mao et al., 2021) takes a different approach to simplify the graph convolution-based propagation strategy. Specifically, it directly approximates the limit of infinite layer graph convolution through constrained loss, which guarantees high accuracy while significantly reducing the computational cost.

To better model users' diverse preferences towards various items, researchers have also explored the disentangled representation learning in recommendation (Li et al., 2022; Liu et al., 2022), which can effectively differentiate the factors in user preferences that drive the user to make decisions on selecting items. For instance, MacridVAE (Ma et al., 2019) disentangles user intent from both macro- and micro- levels based on VAE. DGCF (Wang et al., 2020) adopts GCN as the backbone network to construct a set of intent-aware graphs by separating user interactions to produce a disentangled representation of user intents.

Although the aforementioned recommendation methods have achieved remarkable success, when there are only limited interactions available for users and items, they still suffer from the data sparsity issue. Various approaches have been developed to deal with the data sparsity issues, such as exploiting side information (Catherine & Cohen, 2017; Chang et al., 2023; Cheng et al., 2018; McAuley & Leskovec, 2013; Yang, Feng, Ji, Wang, & Chua, 2021) and high-order interactions among users and items (He et al., 2020; Wang et al., 2019). In this work, we focus on the cross-domain recommendation, which has garnered considerable attention result of its effectiveness. We will make a brief review of the recent progress of CDR in the next subsection.

2.2. Cross-domain recommendation

Cross-domain recommendation (CDR) aims to leverage knowledge from auxiliary domains to enhance modeling of users in target domain. Early CDR methods directly extend single-domain recommendation methods to multi-domains to enhance performance in target domain. For example, CMF (Singh & Gordon, 2008) collaboratively factorizes user–item interaction matrix by sharing user representations in different domains. Another example is TiDA-GCN (Guo et al., 2022), which constructs a unified graph of two domains and applies a time interval-enhanced graph convolution message propagation strategy to learn user and item representations. Later on, the transfer learning-based CDR methods gained more attention due to the success of transfer learning in various fields (Sung, Cho, & Bansal, 2022; Zamir et al., 2018). One typical CDR method is EMCDR, which constructs a two-domain user mapping function from aligned common user representations (Man et al., 2017). Recently, more advanced techniques have been applied in CDR to enhance performance, including dual learning (Li & Tuzhilin, 2020), meta-learning (Zhu et al., 2021, 2022), reinforcement learning (Guo, Zhang, Chen, Wang, & Yin, 2023), and disentangled representation learning (Cao et al., 2022). PTUPCDR (Zhu et al., 2022), for example, uses a meta-network to learn user features in auxiliary domains and generate personalized bridge functions that transfer user preferences to target domains. RL-ISN (Guo et al., 2023) achieves efficient knowledge transfer across both domains by utilizing a reinforcement learning-enhanced domain filter to eliminate irrelevant user behaviors that may impair cross-domain recommendations. Despite the advancements made by these methods, they disregard domain-specific preferences when performing information transfer, which may erroneously transfer domain-specific preferences across domains, and lead to negative transfer problems. To achieve more robust user preference modeling and effective information transfer, disentangled representation learning-based CDR approaches have been developed more recently. For instance, DisenCDR (Cao et al., 2022) adopts a variational bipartite graph encoder (VBGE) to learn domain-specific user representations for each domain and domain-shared representations of both domains; and then decouples domain-specific and domain-shared representations using exclusive and informative regularizers. The domain-shared representations are generated based on a weighting combination of the domain-specific representations from different domains, which means it needs to carefully design a weighting scheme to measure the contributions of different domains for good performance. In fact, the weighting method used in DisenCDR may introduce interaction noises into the domain-shared representation learning, as discussed in Section 1.

To overcome this limitation, our model trains domain-specific and domain-shared representations (or domain-conditional representations in our model) separately for different domains. To ensure that the domain-conditional representations learned from different domains are similar, we propose a conditional VBGE which utilizes domain-specific representations in auxiliary domain as a conditional information to guide the learning of domain-conditional representations in target domain. Furthermore, we design a bridge loss function to encourage their proximity in the embedding space. Consequently, our model avoids the requirement of designing a weighting scheme to measure the contributions of different domains in order to obtain the domain-shared representations and offers greater flexibility in learning domain-shared representations.

3. Proposed model

3.1. Preliminaries

Problem setting. In this paper, we consider two domains, X and Y, which share a common user set denoted by $\mathcal{U}$ with a size of M, and the item sets for the two domains are denoted as $\mathcal{V}^{\mathcal{X}}$ and $\mathcal{V}^{\mathcal{Y}}$ with sizes of I and J, respectively. Let $\mathcal{G}^{\mathcal{X}} = \left(\mathcal{U}, \mathcal{V}^{\mathcal{X}}, \mathcal{E}^{\mathcal{X}}\right)$ and $\mathcal{G}^{\mathcal{Y}} = \left(\mathcal{U}, \mathcal{V}^{\mathcal{Y}}, \mathcal{E}^{\mathcal{Y}}\right)$ denote the user-item bipartite graph for domain X and Y, where $\mathcal{E}^{\mathcal{X}}$ and $\mathcal{E}^{\mathcal{Y}}$ represent two observed user interactions in domain X and domain Y, respectively. Next, DRLCDR aims to learn domain-specific user representations $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}$ and $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}$, item representations $\boldsymbol{z}_{v}^{\boldsymbol{x}}$ and $\boldsymbol{z}_{v}^{\boldsymbol{y}}$, and to transfer knowledge across domains through the domain-conditional representations $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ and $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}_c}$, in order to improve the recommendation performance in both domains.

Embedding Initialization. For embedding initialization, we follow previous work (He et al., 2020; Liu et al., 2020) to map user and item IDs into an embedding space. For independence of the learned domain-specific and domain-conditional representations, DRLCDR initializes them separately. For each user u, we use $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}^{(0)}} \in \mathbb{R}^d$, $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}^{(0)}} \in \mathbb{R}^d$, $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}_c{}^{(0)}} \in \mathbb{R}^d$, and $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}_c{}^{(0)}} \in \mathbb{R}^d$ as the initial embedding vectors for user domain-X-specific, domain-Y-specific, domain-X-conditional, and domain-Y-conditional representations, respectively. d is the embedding size. Similarly, $\boldsymbol{e}_{\boldsymbol{v}}^{\boldsymbol{x}^{(0)}}$ and $\boldsymbol{e}_{\boldsymbol{v}}^{\boldsymbol{y}^{(0)}}$ represent the initial embedding vectors for items in domain X and domain Y, respectively. Specifically, for domain X, the initial embeddings of user u_m and item v_i are:

$$\boldsymbol{e}_{\boldsymbol{u_m}}^{\boldsymbol{x}^{(0)}} = \boldsymbol{P}^{\boldsymbol{x}} \cdot \boldsymbol{ID}_{\boldsymbol{u}}^{\boldsymbol{x}}; \qquad \boldsymbol{e}_{\boldsymbol{u_m}}^{\boldsymbol{x}_c{}^{(0)}} = \boldsymbol{P}^{\boldsymbol{x}_c} \cdot \boldsymbol{ID}_{\boldsymbol{u}}^{\boldsymbol{x}}; \qquad \boldsymbol{e}_{\boldsymbol{v_i}}^{\boldsymbol{x}^{(0)}} = \boldsymbol{Q}^{\boldsymbol{x}} \cdot \boldsymbol{ID}_{\boldsymbol{v}}^{\boldsymbol{x}}, \tag{1}$$

where $\boldsymbol{P}^{\boldsymbol{x}} \in \mathbb{R}^{M \times d}$ and $\boldsymbol{P}^{\boldsymbol{x}_c} \in \mathbb{R}^{M \times d}$ are the learnable parameter matrices for user u_m for domain-specific and domain-conditional representation learning, respectively. And $\boldsymbol{Q}^{\boldsymbol{x}} \in \mathbb{R}^{I \times d}$ is the learnable parameter matrix for item v_i. The matrices $\boldsymbol{ID}_{\boldsymbol{u}}^{\boldsymbol{x}}$ and $\boldsymbol{ID}_{\boldsymbol{v}}^{\boldsymbol{x}}$ are ID embedding matrices for user $u \in \mathcal{U}$ and item $v \in \mathcal{V}^{\mathcal{X}}$, respectively. The embeddings $\boldsymbol{e}_{\boldsymbol{u_m}}^{\boldsymbol{y}^{(0)}}$, $\boldsymbol{e}_{\boldsymbol{u_m}}^{\boldsymbol{y}_c{}^{(0)}}$ and $\boldsymbol{e}_{\boldsymbol{v_j}}^{\boldsymbol{y}^{(0)}}$ in domain Y are initialized in the same way.

3.2. Our model

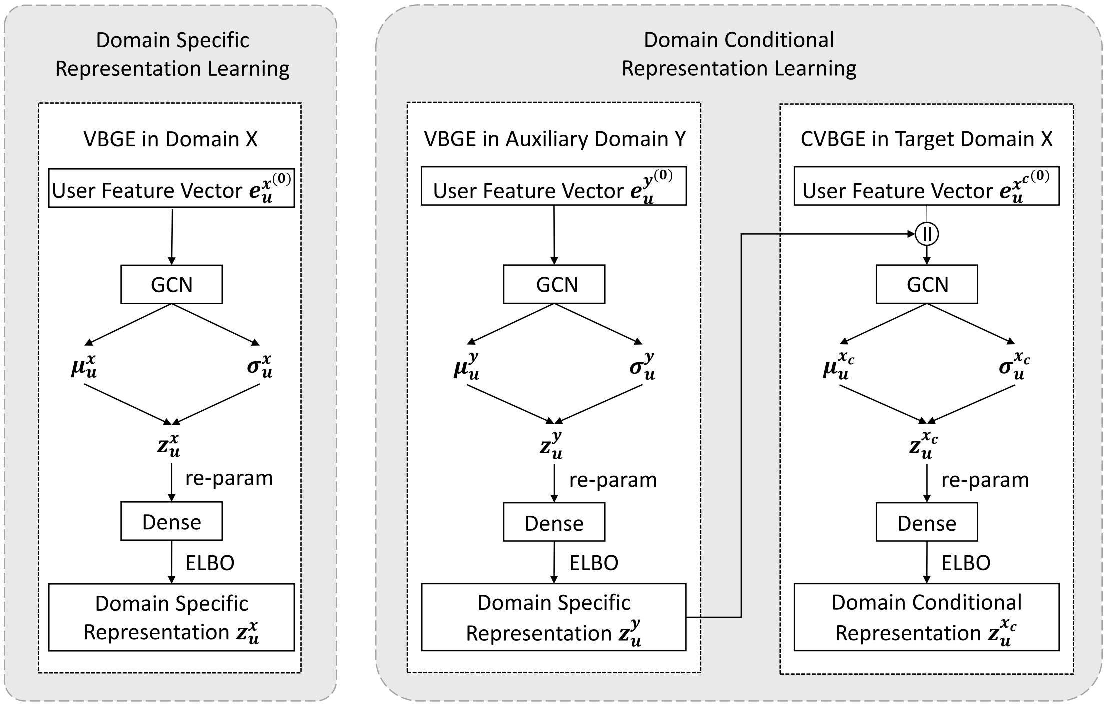
Before delving into the details, we would like to first provide an overview of our model. DRLCDR learns *domain-specific representations* (**DSRs**) and *domain-conditional representations* (**DCRs**) of user for each domain based on the user-item interaction data in that domain. We adopt the variational bipartite graph encoder (VBGE) (Cao et al., 2022) to learn both representations from the user-item interaction data. Specifically, VBGE is used to learn the DSRs for each domain firstly. Then, we use the learned DSRs from one domain as a source of conditional information to guide the DCRs learning in the other domain. Fig. 1 illustrates the learning process of DCRs and DSRs in our model. It is worth mentioning that the DCRs in our model represent the shared preferences across different domains. This guidance ensures that the DCRs learned in different domains are close in the embedding space. Furthermore, we introduce a *decouple loss function* to further separate DSRs and DCRs within each domain and a *bridge loss function* to encourage the proximity of the DCRs from different domain in the embedding space.

In the following subsections, we provide a brief introduction to GCN, which is the core model used in VBGE for modeling user-item interaction data. Subsequently, we provide a comprehensive description of the representation learning process for DSRs and DCRs with VBGE, and introduce the decouple loss and bridge loss functions. Finally, we present the prediction function employed by our model.

3.2.1. GCN brief

The core of GCN is to iteratively aggregate messages from neighboring nodes for the current node (Cheng et al., 2023). In our model, we retain the feature transformation matrix and nonlinear activation function which are considered unnecessary in the single-domain recommendation systems (Chen, Wu, Hong, Zhang, & Wang, 2020; He et al., 2020). This is because our model learns with domain X and domain Y as source and target domains of mutual. To avoid noise in one domain affecting the learning of the other, DRLCDR adopts the two components to enhance the generalization of the model. Taking user domain-specific representation learning as an example, for the user u and item v in domain X, the propagation process of this node on the user-item interaction graph is expressed as:

$$\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}^{(l+1)}} = \delta\left(\sum_{v \in \mathcal{N}_u^x} \frac{1}{\sqrt{|\mathcal{N}_u^x|\,|\mathcal{N}_v^x|}} \left(\boldsymbol{W}_{\boldsymbol{u}}^{\boldsymbol{x}^{(l)}} \boldsymbol{e}_{\boldsymbol{v}}^{\boldsymbol{x}^{(l)}}\right)\right), \qquad \boldsymbol{e}_{\boldsymbol{v}}^{\boldsymbol{x}^{(l+1)}} = \delta\left(\sum_{u \in \mathcal{N}_v^x} \frac{1}{\sqrt{|\mathcal{N}_u^x|\,|\mathcal{N}_v^x|}} \left(\boldsymbol{W}_{\boldsymbol{v}}^{\boldsymbol{x}^{(l)}} \boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}^{(l)}}\right)\right), \tag{2}$$



(a) Domain Specific Representation Learning

(b) Domain Conditional Representation Learning

Fig. 1. Illustration of DSRs and DCRs learning in DRLCDR.

where $e_u^{x(l+1)}$ and $e_v^{x(l+1)}$ denote the updated embedding of user u and item v after propagation at layer l. $\mathcal{N}_u^x$ and $\mathcal{N}_v^x$ denote the set of neighboring nodes to the user u and item v in domain X, respectively. $W_u^{x(l)} \in \mathbb{R}^{d' \times d}$ and $W_v^{x(l)} \in \mathbb{R}^{d' \times d}$ are the trainable weight matrix and d' is the dimension size of the transformation. $\delta(\cdot)$ is the activation function and we use LeakyReLU in DRLCDR. After L layers' propagation, we can obtain $L+1$ embeddings to describe the user. The final embedding e_u^x and e_v^x are obtained by aggregating these embeddings as follows:

$$e_u^x = \sum_{l=0}^{L} e_u^{x(l)}, \qquad e_v^x = \sum_{l=0}^{L} e_v^{x(l)}. \tag{3}$$

For the sake of simplicity in presentation, we use $GCN(\cdot)$ to denote the final embedding obtained by GCN hereafter. For example, the embedding of user u in domain X obtained by GCN (i.e., e_u^x) is denoted as $GCN(e_u^{x(0)})$.

3.2.2. Domain-specific representation learning

Following the learning of user and item embeddings with GCN, we adopt variational bipartite graph encoder (VBGE) (Cao et al., 2022) to approximate their feature distributions, i.e., posterior distributions. Specifically, as shown in Fig. 1(a), for a user u in domain X, VBGE leverages GCN to explicitly encode the collaborative signals of higher-order neighbors and follows VAE paradigm to learn the mean and standard deviation of the domain-specific representation distribution of user to model its domain-specific representation z_u^x, as follows:

$$\mu_u^x = GCN_\mu^x\left(e_u^{x(0)}\right); \qquad \sigma_u^x = GCN_\sigma^x\left(e_u^{x(0)}\right); \qquad z_u^x \sim \mathcal{N}\left(\mu_u^x, \left[diag\left\{\sigma_u^x\right\}\right]^2\right), \tag{4}$$

where μ_u^x and σ_u^x are the mean and standard deviation of the Gaussian distribution z_u^x. Note that they are learned separately with different GCNs. Therefore, we use $GCN_\mu^x(\cdot)$ and $GCN_\sigma^x(\cdot)$ to denote different GCN models for learning mean μ_u^x and standard deviation σ_u^x representation in domain X, respectively.

VBGE approximates the posterior distribution of the domain-specific representation by maximizing the likelihood of the observed interactions based on the Gaussian distribution defined by the learned mean and standard deviation (as shown in Eq. (4)) (Liang, Krishnan, Hoffman, & Jebara, 2018; Ma et al., 2019). However, due to the sampling operation for this distribution is non-differentiable, a re-parameterization trick is often adopted to deal with this issue. Concretely, the sampling operation is transformed into a differentiable operation by introducing a differentiable variable to replace the random sampling process, i.e., sampling z_u^x for user u from the standard Gaussian distribution $\mathcal{N}(0,1)$ with the following operation:

$$\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}} = \boldsymbol{\mu}_{\boldsymbol{u}}^{\boldsymbol{x}} + \boldsymbol{\sigma}_{\boldsymbol{u}}^{\boldsymbol{x}} \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0,1), \tag{5}$$

where $\odot$ is the dot product operation. The approximate posterior distributions of users and items in domain X and domain Y (i.e., $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}$, $\boldsymbol{z}_{v}^{\boldsymbol{y}}$ and $\boldsymbol{z}_{v}^{\boldsymbol{x}}$) can be obtained by using re-parameterization sampling in a similar way.

To learn the domain-specific representation of user u in domain X, we need to minimize the difference between the approximate posterior distribution obtained through re-parameterization sampling and the true distribution. This is achieved by maximizing the evidence lower bound (ELBO) (Kingma & Welling, 2013; Rezende, Mohamed, & Wierstra, 2014) to optimize the variational lower bound for domain-specific representation learning, i.e., to maximize the likelihood of observed input data. Thus, the objective function is transformed to maximize the likelihood estimates of the learned domain-specific representation and the true domain-specific representation and minimize the difference between the sampled domain-specific distribution and the true distribution. Take $L\left(u^x;\theta_u^x,\phi_u^x\right)$ in domain X as an example, the objective function is:

$$L\left(u^x;\theta_u^x,\phi_u^x\right) = D_{KL}\left[q_{\phi_u^x}\left(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}|u^x\right) \parallel p\left(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}\right)\right] - \mathbb{E}_{q_{\phi_u^x}(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}|u^x)}\left[log p_{\theta_u^x}\left(u^x|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}\right)\right], \tag{6}$$

where $L\left(u^x;\theta_u^x,\phi_u^x\right)$ is variational inference loss for user u. θ_u^x and ϕ_u^x are parameter of VBGE. $L(u^y;\theta_u^y,\phi_u^y)$, $L(v^y;\theta_v^y,\phi_v^y)$ and $L(v^x;\theta_v^x,\phi_v^x)$ can be derived analogously for the DSRs of users and items in both domains.

3.2.3. Domain-conditional representation learning

To learn a user's DCRs, we leverage the DSRs learned from in the source domain as guiding information in VBGE, as shown in Fig. 1(b). Specifically, in domain X, we construct a conditional representation vector $\boldsymbol{c}_{\boldsymbol{u}}^{\boldsymbol{x}}$ based on its initial embedding vector $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}_c(\mathbf{0})}$ and the DSRs from Domain Y (i.e., $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}$) with a concatenation operation, namely:

$$\boldsymbol{c}_{\boldsymbol{u}}^{\boldsymbol{x}} = \boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}_c(\mathbf{0})} \parallel \boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}, \tag{7}$$

where $\parallel$ is the concatenation operation. In the next, we take the conditional representation vector $\boldsymbol{c}_{\boldsymbol{u}}^{\boldsymbol{x}}$ as input and adopt the VAE framework to learn the domain conditional representation $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ of user u as follows:

$$\boldsymbol{\mu}_{\boldsymbol{u}}^{\boldsymbol{x}_c} = GCN_{\mu}^{x_c}\left(\boldsymbol{c}_{\boldsymbol{u}}^{\boldsymbol{x}}\right); \qquad \boldsymbol{\sigma}_{\boldsymbol{u}}^{\boldsymbol{x}_c} = GCN_{\sigma}^{x_c}\left(\boldsymbol{c}_{\boldsymbol{u}}^{\boldsymbol{x}}\right); \qquad \boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{u}}^{\boldsymbol{x}_c}, \left[diag\left\{\boldsymbol{\sigma}_{\boldsymbol{u}}^{\boldsymbol{x}_c}\right\}\right]^2\right), \tag{8}$$

where $\boldsymbol{\mu}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ and $\boldsymbol{\sigma}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ are the Gaussian distribution $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ of the mean and standard deviation. $GCN_{\mu}^{x_c}(\cdot)$ and $GCN_{\sigma}^{x_c}(\cdot)$ denote the GCNs for learning mean $\boldsymbol{\mu}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ and standard deviation $\boldsymbol{\sigma}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ representation in domain X, respectively. The domain conditional representation $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}_c}$ of user u in domain Y can be obtained in a similar way.

In the next, the re-parameterization trick also has been used to optimize the variational lower bound by maximizing ELBO (Kingma & Welling, 2013; Rezende et al., 2014). The optimization objective is to maximize the reconstruction loss and the negative KL divergence between the approximation posterior distribution of domain-conditional representation and the true distribution, in this case. The variational lower bound optimization objective can be rewritten as:

$$L\left(u^{x_c},\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}};\theta_u^{x_c},\phi_u^{x_c}\right) = D_{KL}\left[q_{\phi_u^{x_c}}\left(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}|u^x,\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}\right) \parallel p\left(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}\right)\right] - \mathbb{E}_{q_{\phi_u^{x_c}}\left(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}|u^x,\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}\right)}\left[log p_{\theta_u^x}\left(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}},u^x|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}\right)\right], \tag{9}$$

where $q_{\phi_u^{x_c}}\left(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}|u^x,\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}\right)$ is the approximate posterior distribution of domain-X-conditional representation with $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}}$ as the conditional information. The other notations are defined in the same way as in Eq. (6). Similarly, $L\left(u^{y_c},\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}};\theta_u^{y_c},\phi_u^{y_c}\right)$ can be derived for the domain-conditional representation of user u in domain Y.

Finally, the total variational inference loss for learning DCRs and DSRs in our model can be summarized as:

$$\begin{aligned} L_V \quad &= \textstyle\sum_{u\in\mathcal{U},v\in\mathcal{V}^{\mathcal{X}}\cup\mathcal{V}^{\mathcal{Y}}}(L(u^x;\theta_u^x,\phi_u^x) + L(u^y;\theta_u^y,\phi_u^y) + L(v^y;\theta_v^y,\phi_v^y) + L(v^x;\theta_v^x,\phi_v^x) \\ &+ L\left(u^{x_c},\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{y}};\theta_u^{x_c},\phi_u^{x_c}\right) + L\left(u^{y_c},\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}};\theta_u^{y_c},\phi_u^{y_c}\right)). \end{aligned} \tag{10}$$

3.2.4. Decouple loss & bridge loss

Decouple loss. For good performance in CDR, it is preferable for the representations of domain-specific and domain-conditional to be independent of mutual. However, adopting different initialization embedding vectors does not ensure that they are mutually exclusive. Therefore, we introduce a decouple loss in our model to further disentangle the DSRs and DCRs. Following previous work (Cao et al., 2022), we utilize Kullback–Leibler divergence to encourage the independence of DCRs and DSRs because it directly computes the divergence between probability distributions to represent their differences. Specifically, for DCRs $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}$ and DSRs $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ of user u in domain X, the decouple loss $L_{KL_u}^{x}$ is defined as:

$$L_{KL_u}^{X} = -\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}} log\frac{\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}}}{\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{x}_c}} \tag{11}$$

The decouple loss in domain Y (i.e., $L_{KL_u}^{Y}$) can be computed in the same way. Finally, the total decouple loss of our model is summarized as:

$$L_D = \sum_{u\in\mathcal{U}}(L_{KL_u}^{X} + L_{KL_u}^{Y}). \tag{12}$$

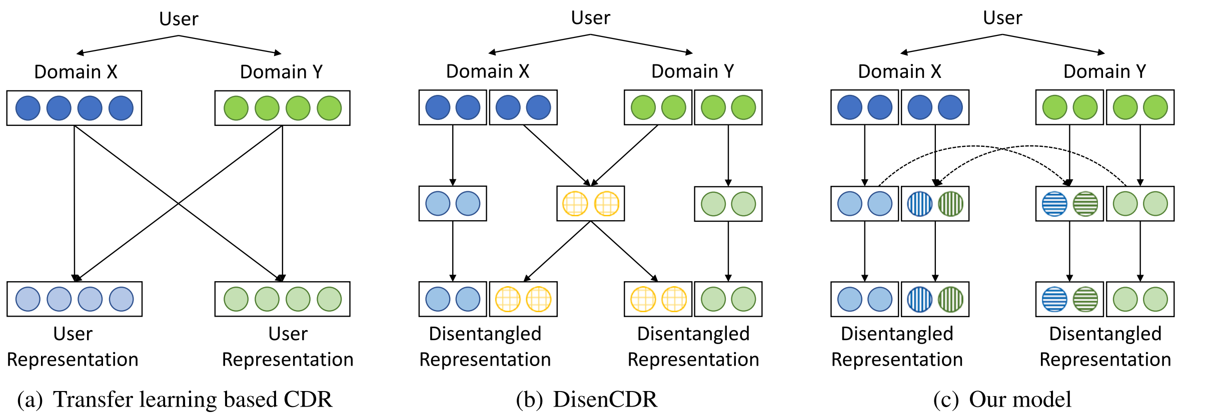


Fig. 2. A model comparison: (a) the transfer learning-based CDR models which utilize transfer strategies to fuse knowledge between two domains; (b) DisenCDR learns a domain-shared representation for both domains and disentangles the domain-shared and domain-specific representations; (c) our model learns domain-specific representations (DSRs) and domain-conditional representations (DCRs) separately for each domain and uses the DSRs in one domain as conditional information to learn the DCRs in the other domain.

Bridge loss. In our model, DCRs represent user preferences that are shared across different domains. Therefore, it is crucial to ensure that the DCRs learned in different domains (X and Y) are as similar as possible in the embedding space. To achieve this, we introduce a bridge loss to encourage their closeness. Specifically, given the domain-X-conditional representation $z_u^{x_c}$ and the domain-Y-conditional representation $z_u^{y_c}$ for each user u, the bridge loss in our model is defined as:

$$L_B = \sum_{u \in \mathcal{U}} z_u^{x_c} log \frac{z_u^{x_c}}{z_u^{y_c}}. \tag{13}$$

3.2.5. Prediction

Our model represents user preferences in two parts: DCRs and DSRs. To make predictions, we combine them with addition for simplicity. More sophisticated combination methods are left for future exploration. The final prediction is made as follows:

$$r_{ui}^x = \sigma\left(S\left(\left(z_u^{x_c} + z_u^x\right), z_v^x\right)\right), \tag{14}$$

where $\sigma(\cdot)$ is the sigmoid function; $S(\cdot)$ is the score function, where the dot product operation is adopted.

3.3. Model training

In this study, we followed the Top-N ranking task in the recommender system by ranking all unobserved items per test user. Following previous works (Cao et al., 2022; Liu et al., 2020), the binary cross-entropy function is adopted for the recommendation loss function. Formally, the objective function is defined as:

$$L_{BCE} = -\sum_{(u,v)\in \mathcal{R}^+ \cup \mathcal{R}^-} r_{uv} log \hat{r}_{uv} + \left(1 - r_{uv} log\left(1 - \hat{r}_{uv}\right)\right) + \lambda \|\Theta\|_2^2, \tag{15}$$

where $\mathcal{R}^+$ is the set of observed in the interaction matrix, $\mathcal{R}^-$ is the set of random sampling from unobserved in the interaction matrix. λ is the l_2 regularization parameter, Θ denotes the model parameter set. Considering all the loss in our model, the final loss function can be formulated as:

$$L = L_{BCE} + \alpha L_V + \beta L_D + \gamma L_B, \tag{16}$$

where α is the weight of variational inference loss; β is the coefficient to control the degree of disentanglement between the domain-specific and domain-conditional representations in the two domains; γ is the coefficient to control the degree of knowledge transfer in the two domains.

The mini-batch Adam algorithm is utilized to optimize the model. In addition, to alleviate the overfitting problem, we adopt the dropout techniques during the model training process, which is consistent with previous studies (He et al., 2020; Liu et al., 2020). The drop ratios were carefully optimized through experimentation.

4. Relation to other models

In this section, we discuss the difference of our model to other transfer learning-based CDR methods and the recent DisenCDR model.

4.1. Transfer learning-based CDR

The general paradigm of transfer learning-based CDR methods first learns user representations with an interaction encoder (like MLP or GNN) in each domain, and then transfers knowledge learned across domain with various techniques, such as mapping (Man et al., 2017), meta-learning (Zhu et al., 2021, 2022), and parameter sharing (Xi et al., 2020), to assist modeling user preference in target domain. For example, DDTCDR (Li & Tuzhilin, 2020) adopts MLP as the encoder to transfer user similarity across domains by learning potential orthogonal mapping functions. PTUPCDR (Zhu et al., 2022) presents a generic framework that can use different base encoders to create personalized bridge functions, which is accomplished by training a meta-network with user preference embeddings to transfer personalized preferences to each user. This paradigm can be formulated as:

$$\boldsymbol{u}^{\boldsymbol{x}} = f\left(h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}}), h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}})\right), \tag{17}$$

where $\boldsymbol{u}^{\boldsymbol{x}}$ denotes the user representations of user u in the domain X; $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}}$, $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}}$ are initialization embedding of user u in domain X and domain Y, respectively. $h(\cdot)$ denotes the modeling function of base encoder and $f(\cdot)$ is the transfer learning function.

The paradigm of transfer learning-based CDR methods is illustrated in Fig. 2(a). The limitation of transfer learning-based methods lies in their inability to differentiate between domain-specific and domain-shared preferences while transferring information across domain. This can result in negative transfer problems.

4.2. DisenCDR recap

To tackle the limitation in transfer learning-based CDR methods, the recent DisenCDR models adopt the disentangled learning techniques to disentangle domain-shared and domain-specific information. More specifically, DisenCDR (Cao et al., 2022) uses a variational bipartite graph encoder (VBGE) within the framework of a Variational Autoencoder (VAE) by using GCN as the encoder. Based on the VBGE, it learns a domain-specific representation and a domain-shared representation for each user with an information regularizer. Additionally, an exclusive regularizer is employed to ensure independence between the two types of representations. The illustration of DisenCDR is shown in Fig. 2(b). In domain X, the representation of user u is defined as:

$$\boldsymbol{u}^{\boldsymbol{x}} = \underbrace{g\left(h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}})\right)}_{domain-specific} + \underbrace{g\left(\lambda \cdot h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}_{sh}}) + (1-\lambda)\cdot h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}_{sh}})\right)}_{domain-shared}; \qquad \lambda = \frac{\mathcal{N}_u^x}{\mathcal{N}_u^x + \mathcal{N}_u^y}, \tag{18}$$

where $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}}$, $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}_{sh}}$ and $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}_{sh}}$ are domian-X-specific, domain-X-shared and domain-Y-shared initialization embedding of user u, respectively. $h(\cdot)$ is the GCN encoder and $g(\cdot)$ denotes VBGE. λ is a parameter to control the contribution of the two domains. It is computed as the ratio of the number of u's interacted items in domain X to the total number of items interacted in both domains. Note that the ratio of user interactions cannot effectively learn the similar preferences of users between two domains. Because the observed interactions often contain noise, which may negatively impact the learning of the domain-shared representation.

4.3. Our model

The framework of our model is illustrated by Fig. 2(c). It can be noted from Fig. 2(b) that DisenCDR learns domain-specific user representations for each domain and a common domain-shared representation for both domains. To learn the domain-shared representation, it needs to carefully design a mechanism to control the contributions of different domains. In DRLCDR, we replace domain-shared representations with domain-conditional representations, which are also learned for each domain. To ensure the domain-conditional representations and facilitate the information transferring across two domains, we use two strategies: (1) Using the learned domain-specific user representation in one domain as an information source to guide the domain-conditional representation learning in the other domain. In this way, the domain-conditional representations in both domains are actually learned conditionally on information extracted from the other domain and the interaction information of their own domain, which will make them closer in the embedding space. And (2) using a bridge loss function to encourage them to be closer in the feature space. The representation of user u in domain X can be expressed as:

$$\boldsymbol{u}^{\boldsymbol{x}} = \underbrace{g\left(h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}})\right)}_{domain-specific} + \underbrace{g\left(h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}_c}), g\left(h(\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}})\right)\right)}_{domain-conditional}, \tag{19}$$

where $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{x}_c}$ and $\boldsymbol{e}_{\boldsymbol{u}}^{\boldsymbol{y}}$ are domain-X-conditional and domain-Y-specific initialization embedding of user u, respectively. $h(\cdot)$ is the GCN encoder and $g(\cdot)$ denotes VBGE. Compared with existing studies, from the equation, it can be seen that our model does not need the parameter to control the contribution of different domains on learning the domain-shared representations. It is more flexible and can naturally avoid the problem introduced by the parameter as in DisenDCR.

5. Experiment

The effectiveness of DRLCDR is validated by extensive experiments on four real-world datasets. In this section, we mainly focus on the following three research questions:

RQ1: How does DRLCDR perform on ranking recommendation tasks comparing single-domain and cross-domain methods baseline?

RQ2: How do different components in our model impact the recommendation performance?

RQ3: How do the key hyperparameters affect the recommended performance of our model?

Table 1
Basic statistics of the couple datasets.

Dataset	#user	#items	#train	#test	Sparsity
Cloth	9928	41,303	87,829	7562	99.98%
Sport	9928	32,310	92,612	8314	99.97%
Phone	20,448	28,657	142,790	18,851	99.97%
Elec	20,448	60,756	303,896	18,701	99.97%
Cloth	5860	30,870	50,016	3640	99.97%
Phone	5860	17,685	47,671	4680	99.95%
Sport	4998	22,101	50,558	3763	99.95%
Phone	4998	14,618	42,446	3984	99.94%

5.1. Experimental setup

5.1.1. Datasets

The Amazon dataset[1] is widely adopted in cross-domain recommendation studies (Liu et al., 2020; Zhao et al., 2019). We conduct experimentations on four categories in Amazon dataset, including Phones, Cloth, Elec and Sport. These four datasets contain historical interaction data of Amazon users from 1996 to 2014, and corresponding information such as user reviews and item attributes. We divide them into four experimental groups that mutually serve as source and target domains, Cloth&Sport, Phone&Elec, Cloth&Phone and Sport&Phone. For each experimental group, we retain only the overlapping users' historical interaction data, which includes users and items with at least 10 interactions. In other words, for the experimental group Cloth&Sport, we can either use the information from Cloth to assist Sport training or use Sport as a source domain to improve performance of recommendations in Cloth. And following the common setting in CDR, only common users in both domains are used in experiments to evaluate effectiveness of CDR models. Table 1 summarizes the detailed statistics for the four pairs of datasets.

5.1.2. Evaluation protocol

We adopt the Leave-One-Out (LOO) evaluation method to evaluate the recommendation performance of all methods include DRLCDR and the competitors following prior works (Cao et al., 2022; Liu et al., 2020). In particular, we conducted a random sampling process in the pre-processed dataset, where we selected one interaction per user as the test set, while considering the remaining data as the training set. DRLCDR utilizes the learned users and items representation to predict scores for 1000 candidates, which include 999 negative items and 1 positive item sampled from the interaction data (Zhao, Chen, Wang, Gu, & Wen, 2020). We adopt Hit Ratio (HR) and Normalized Discount Cumulative Gain (NDCG) for performance comparison over the Top-N items which are set as 20.

5.1.3. Baselines

We compare DRLCDR among the following single-domain and cross-domain baselines.

Single-domain baselines:

- **NeuMF** (He et al., 2017): It is a typical neural collaborative method which takes advantage of a multi-layer perceptron to capture the non-linear interactions between the features above the concatenation of the user and item vectors.
- **LightGCN** (He et al., 2020): This classical recommendation approach is based on GCN and simplifies the GCN framework by removing unnecessary components to improve recommendation performance.
- **DGCF** (Wang et al., 2020): The method is based on GCN and models user intent by introducing user intent-aware interaction graphs and encourages independence among the intents by applying disentangled representation learning techniques.
- **SimGCL** (Yu et al., 2022): This method replaces the graph augmentation technique with adding uniform noise in the embedding space and simplifies the contrast learning model.

Cross-domain baselines:

- **CoNet** (Hu et al., 2018): This method introduces a cross-connection network to link the base networks of both domains and enhances bi-directional knowledge transfer between the two domains.
- **BiTGCF** (Liu et al., 2020): This is a GCN-based CDR method. It achieves bi-directional knowledge transfer between two domains by a newly designed feature propagation module, while improving the recommendation performance of each domain.
- **TMCDR** (Zhu et al., 2021): This method implicitly transforms the user representations learned by the base network across domain via the meta-network after pre-training the two-domain models on the source and target domains with the base network, respectively.
- **DisenCDR** (Cao et al., 2022): This method is the latest CDR method based on disentangled representation learning, which disentangles the user intent of source and target domains by two mutual information-based disentangled regularization methods to learn the representations of domain-specific and domain-shared.

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

Table 2
Performance comparison in terms of HR@20 and NDCG@20.

Dataset	Metrics	NeuMF	LightGCN	DGCF	SimGCL	CoNet	Bi-TGCF	TMCDR	DisenCDR	DRLCDR	
Cloth	HR	0.1708	0.1560	0.1775	0.2132	0.1732	0.2241	0.1627	0.2479	**0.2571**	3.58% ↑
	NDCG	0.0771	0.0782	0.0821	0.1019	0.0779	0.1171	0.0756	0.1273	**0.1309**	2.75% ↑
Sport	HR	0.1907	0.2178	0.2254	0.2379	0.1932	0.2827	0.1864	0.3132	**0.3214**	2.55% ↑
	NDCG	0.0855	0.1071	0.1085	0.1197	0.0863	0.1448	0.0832	0.1543	**0.1570**	1.72% ↑
Phone	HR	0.3304	0.3827	0.3873	0.4052	0.3142	0.4163	0.3422	0.4357	**0.4427**	1.58% ↑
	NDCG	0.1635	0.2047	0.2067	0.2208	0.1507	0.2222	0.1706	0.2348	**0.2394**	1.92% ↑
Elec	HR	0.3265	0.3280	0.3345	0.3547	0.3103	0.3677	0.3260	0.3852	**0.3987**	3.39% ↑
	NDCG	0.1667	0.1676	0.1693	0.1893	0.1586	0.1931	0.1693	0.2069	**0.2126**	2.68% ↑
Cloth	HR	0.1188	0.1286	0.1353	0.1493	0.1302	0.1572	0.0973	0.1742	**0.1835**	5.08% ↑
	NDCG	0.0520	0.0537	0.0572	0.0632	0.0540	0.0724	0.0432	0.0833	**0.0863**	3.47% ↑
Phone	HR	0.2394	0.2613	0.2704	0.2890	0.2416	0.2988	0.2335	0.3139	**0.3421**	8.24% ↑
	NDCG	0.1113	0.1345	0.1392	0.1473	0.1045	0.1502	0.1102	0.1507	**0.1638**	7.98% ↑
Sport	HR	0.1967	0.2163	0.2274	0.2457	0.2007	0.2563	0.2053	0.2965	**0.2996**	1.03% ↑
	NDCG	0.0910	0.1102	0.1185	0.1239	0.0923	0.1247	0.1097	0.1394	**0.1427**	2.31% ↑
Phone	HR	0.2661	0.2769	0.2886	0.2997	0.2319	0.3021	0.2542	0.3388	**0.3426**	1.11% ↑
	NDCG	0.1203	0.1369	0.1463	0.1501	0.1047	0.1496	0.1256	0.1557	**0.1576**	1.21% ↑

5.1.4. Implementation details

We implement the DRLCDR model in PyTorch[2] and our codes are released for reproducibility[3]. To ensure a fair comparison, we adopt the hyperparameter settings reported in the original baseline paper, which are considered the best, and further fine-tune them. For all methods, we fix the embedding size d to 128 and the mini-batch size to 1024, and search for learning rates in the range {0.01, 0.001, 0.0001}. In our model, the layer number of all GCNs (i.e., L) is set to 2. For other parameters, L2 regularization coefficients are carefully searched in the range $\{1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}\}$; the variational inference loss weight α is selected from {0.1, 0.2, 0.3, 0.4, 0.5}; the intra-domain decouple loss weight β is selected from {5, 10, 15, 20, 25}; the inter-domain bridge loss weight γ is selected from {5, 10, 15, 20, 25}. Furthermore, we implement the warm-up strategy and specify the number of epochs to be 10. We test the model every 10 epochs during training and save the best parameters.

5.2. Performance comparison (RQ1)

Table 2 shows the results of our model and all competitors on the four datasets. In particular, the best results are highlighted in bold font, while the second best results are underlined. Our experiments demonstrate superior recommendation performance of the cross-domain model over the single-domain model across four real-world datasets, with several noteworthy observations as detailed below.

- **Performance comparisons among single-domain recommendation (SDR) methods.** LightGCN takes advantage of the GCN technique to exploit the high-order neighbor information and achieves better performance than NeuMF, which adopts a neural network as the base network. DGCF outperforms LightGCN by utilizing both the GCN and disentangled learning techniques to model users' multiple intents. The more recent SimGCL adopts a contrastive learning strategy by adding uniform noise into the embedding space to create contrasting views and achieves the best performance among the single-domain recommendation models.
- **SDR *vs.* CDR.** Comparing the CDR model with the single-domain model, it is evident that the CDR model beats out its corresponding single-domain approach that uses the same backbone model. This highlights the benefits of leveraging additional information from other domains. For instance, both NeuMF and CoNet adopt neural networks as the backbone model. However, CoNet surpasses NeuMF by introducing a cross-connected network that allows bidirectional knowledge transfer across domains. In the case of GCN-based models, BiTGCF performs better than LightGCN by using a bidirectional feature transfer module to model the user representation. This is in contrast to LightGCN, which uses single-domain user-item interaction data. It is worth noting that even though single-domain models like DGCF and SimGCL use advanced techniques like disentangled learning and contrastive learning, Bi-TGCF consistently outperforms them. This further demonstrates the effectiveness of CDR in enhancing recommendation accuracy.
- **Performance comparisons among CDR methods.** Regarding the CDR models, TMCDR uses task-oriented meta-networks as a knowledge transfer strategy and achieves better recommendation results than CoNet, which uses cross-connected networks for knowledge transfer. This is might because that TMCDR can implicitly transform auxiliary domain user embeddings into target feature space. BiTGCF, that uses GCN as a base encoder, achieves better recommendation performance than CoNet,

[2] https://pytorch.org/.
[3] https://github.com/zhangyucs/DRLCDR.

Table 3
Effects of different components in DRLCDR.

Variational inference		✓	✓	✓	✓	✓
Decouple			✓		✓	✓
Bridge				✓	✓	✓
Conditional information						✓
Cloth	HR	0.2408	0.2438	0.2441	0.2456	**0.2571**
	NDCG	0.1223	0.1235	0.1239	0.1247	**0.1309**
Sport	HR	0.3007	0.3053	0.3098	0.3102	**0.3214**
	NDCG	0.1473	0.1492	0.1517	0.1526	**0.1570**
Phone	HR	0.4002	0.4128	0.4283	0.4292	**0.4427**
	NDCG	0.2196	0.2263	0.2278	0.2283	**0.2394**
Elec	HR	0.3525	0.3676	0.3762	0.3786	**0.3987**
	NDCG	0.1902	0.1972	0.2019	0.2037	**0.2126**
Cloth	HR	0.1611	0.1735	0.1772	0.1783	**0.1835**
	NDCG	0.0752	0.0812	0.0819	0.0823	**0.0863**
Phone	HR	0.3017	0.3122	0.3367	0.3137	**0.3421**
	NDCG	0.1431	0.1495	0.1563	0.1498	**0.1638**
Sport	HR	0.2583	0.2849	0.2936	0.2962	**0.2996**
	NDCG	0.1254	0.1338	0.1376	0.1387	**0.1427**
Phone	HR	0.3005	0.3193	0.3302	0.3318	**0.3426**
	NDCG	0.1428	0.1492	0.1503	0.1513	**0.1576**

which uses a neural network as the base encoder. Both DisenCDR and BiTGCF adopt the same knowledge transfer strategy. However, DisenCDR utilizes the variational autoencoder (VAE) framework to disentangle user preferences, which can avoid transferring domain-specific information to the target domain and learning for better performance. As a reminder, DisenCDR obtains domain-shared representations by combining domain-specific representations (DSRs) with a weight scheme that can be biased by noise in user interactions. In contrast, our model learns domain-conditional representations (DCRs) separately using DSRs as conditional information in VBGE and a bridge loss to ensure the proximity of DCRs learned in different domains. Our approach is more flexible in capturing shared preferences across domains and avoids the bias issue occurred in DisenCDR. In experiments, our model consistently outperforms DisenCDR across all datasets.

5.3. Ablation study (RQ2)

In this section, we study the contribution of different components of our model to the final performance. They are four main components in our DRLCDR model: a variational inference component for learning DSRs and DCRs (Variational Inference), a decouple loss for decoupling DSRs and DCRs (Decouple), a bridge loss for connecting user DCRs learned from different domains (Bridge), and a conditional information component using the DSRs to learn DCRs (Conditional Information). The recommendation performances of using one to all the components in DRLCDR are reported in Table 3.

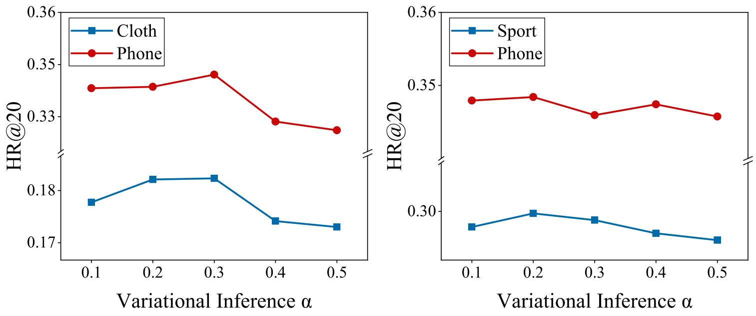
The base model of DRLCDR utilizes variational inference to learn DCR and DSR. By adopting the decoupled loss, we observe improved performance, highlighting the significance of distinguishing between domain-specific and domain-shared preferences in CDR. With the consideration of the bridge loss, it is observed that the performance can be further improved. Note that the DCRs are learned separately in each domain to represent the domain-shared preferences in our model. The improvement demonstrates that it is crucial to ensure their proximity in the embedding space. Although our model does not employ DSRs as conditional information when learning DCRs, the marked improvement in recommendation performance on most datasets clearly demonstrates the effectiveness of the decouple loss and bridge loss. We include this component in the final step to show the utility of DSRs as guiding information in the DCRs learning process. Significantly, it further enhances performance, even after employing the bridge loss, emphasizing the effectiveness of our design. This experiment effectively demonstrates the validity of each component in our model.

5.4. Influence of hyperparameters (RQ3)

In this section, we investigate the effect of three hyperparameters: α, β, and γ in the loss function. Due to space limitations, we only show the results on Cloth&Phone and Sport&Phone experimental groups. The results on other groups exhibit a similar trend.

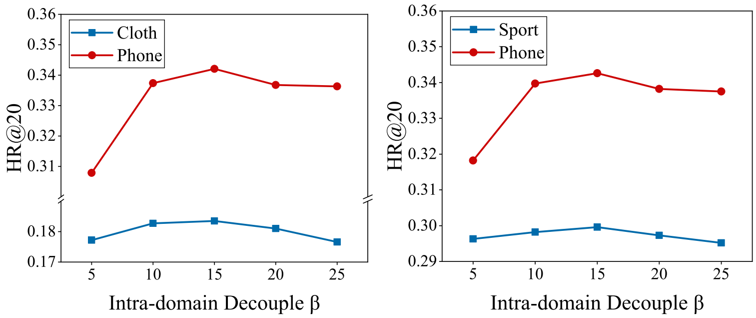
Influence of α. The parameter α controls the level of approximation loss between the generated distribution and the true distribution. The results in Fig. 3 demonstrate that setting a smaller value for α results in the model being unable to learn a suitable representation of user preferences, leading to suboptimal recommendation performance. On the other hand, a larger value of α may weaken the effects of the decouple loss and bridge loss, causing recommendation performance to drop sharply. We recommend carefully selecting the value of α between 0.2 and 0.4 for optimal results

Influence of β. The parameter β is used to control the degree of mutual exclusion of DSRs and DCRs. The results in Fig. 4 indicate that a smaller β fails to effectively decouple these representations, resulting in lower recommendation performance. Conversely, a



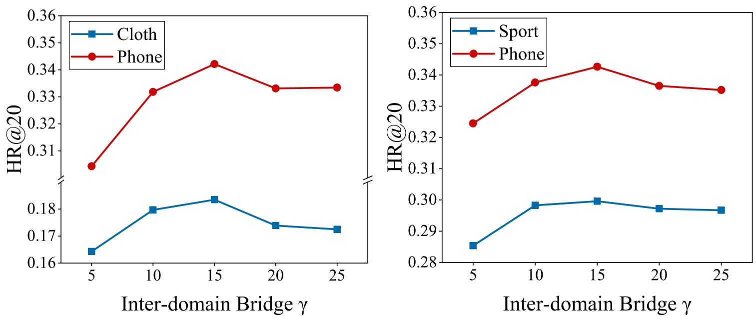
(a) Results in Cloth&Phone (b) Results in Sport&Phone

Fig. 3. Influence of α.



(a) Results in Cloth&Phone (b) Results in Sport&Phone

Fig. 4. Influence of β.



(a) Results in Cloth&Phone (b) Results in Sport&Phone

Fig. 5. Influence of γ.

larger β overemphasizes the mutual exclusion of these representations and causes loss of information in the user representation, leading to reduced generalization ability and suboptimal recommendation performance. In our experiments, a value between 10 and 20 would be better for β.

Influence of γ. The parameter γ controls the degree of approximation for domain condition representation of two domains. The results in Fig. 5 show a smaller γ leads to ineffective knowledge transfer between the domains, resulting in relatively worse recommendation performance. Conversely, a larger γ may make the model put more focus on the closeness of DCRs in different domains, while affecting user representation learning. An optimal value of γ would be 10 and 20 in our experiments.

6. Conclusion

In this paper, we presented a novel cross-domain recommendation model called DRLCDR, which learns the domain-specific representations (DSRs) and domain-shared representations (called domain-conditional representations or DCRs in our model) separately in each model by using the variational bipartite graph encoder. In particular, to ensure the DCRs learned in different domains to be similar, we use the DSRs learned in one domain as conditional information to guide the learning of DCRs in the other domain. In addition, a bridge loss is designed to further encourage the their closeness in the embedding space. To prevent the transfer of DSRs to the target domain, we use a decouple loss to disentangle the DSRs and DCRs in each domain. We evaluate the effectiveness of our model through experiments on four real-world datasets and compare it to a robust set of baselines, and the results demonstrate the superiority of our model.

CRediT authorship contribution statement

Yu Zhang: Methodology, Writing – original draft. **Zhiyong Cheng:** Writing – review & editing, Project administration. **Fan Liu:** Formal analysis, Visualization. **Xun Yang:** Writing – review & editing. **Yuxin Peng:** Conceptualization, Writing – review & editing.

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

Data will be made available on request.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China under Grants 62272254, U22A2094, 61925201, and 62132001; in part by the Shandong Project towards the Integration of Education and Industry under Grants 2022PY009.

References

Berg, Rianne van den, Kipf, Thomas N., & Welling, Max (2017). Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263.

Cao, Jiangxia, Lin, Xixun, Cong, Xin, Ya, Jing, Liu, Tingwen, & Wang, Bin (2022). DisenCDR: Learning disentangled representations for cross-domain recommendation. In *Proceedings of the 45th ACM international conference on research and development in information retrieval* (pp. 267–277).

Catherine, Rose, & Cohen, William (2017). Transnets: Learning to transform for recommendation. In *Proceedings of the 8th ACM conference on recommender systems* (pp. 288–296).

Chang, Yao, Zhou, Wei, Cai, Haini, Fan, Wei, Hu, Linfeng, & Wen, Junhao (2023). Meta-relation assisted knowledge-aware coupled graph neural network for recommendation. *Information Processing & Management*, *60*(3), Article 103353.

Chen, Lei, Wu, Le, Hong, Richang, Zhang, Kun, & Wang, Meng (2020). Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the 34th AAAI conference on artificial intelligence* (pp. 27–34).

Cheng, Zhiyong, Ding, Ying, He, Xiangnan, Zhu, Lei, Song, Xuemeng, & Kankanhalli, Mohan S (2018). A^3NCF: An adaptive aspect attention model for rating prediction. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 3748–3754).

Cheng, Zhiyong, Han, Sai, Liu, Fan, Zhu, Lei, Gao, Zan, & Peng, Yuxin (2023). Multi-behavior recommendation with cascading graph convolution networks. In *Proceedings of the 32nd ACM web conference* (pp. 1181–1189).

Guo, Lei, Zhang, Jinyu, Chen, Tong, Wang, Xinhua, & Yin, Hongzhi (2023). Reinforcement learning-enhanced shared-account cross-domain sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(7), 7397–7411.

Guo, Lei, Zhang, Jinyu, Tang, Li, Chen, Tong, Zhu, Lei, & Yin, Hongzhi (2022). Time interval-enhanced graph neural network for shared-account cross-domain sequential recommendation. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

He, Xiangnan, Deng, Kuan, Wang, Xiang, Li, Yan, Zhang, Yongdong, & Wang, Meng (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd ACM international conference on research and development in information retrieval* (pp. 639–648).

He, Xiangnan, Liao, Lizi, Zhang, Hanwang, Nie, Liqiang, Hu, Xia, & Chua, Tat-Seng (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

He, Ruining, & McAuley, Julian (2016). VBPR: visual Bayesian personalized ranking from implicit feedback. In *Proceedings of the 30th AAAI conference on artificial intelligence*.

Hu, Yifan, Koren, Yehuda, & Volinsky, Chris (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE international conference on data mining* (pp. 263–272).

Hu, Guangneng, Zhang, Yu, & Yang, Qiang (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).

Kingma, Diederik P., & Welling, Max (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Koren, Yehuda, Bell, Robert, & Volinsky, Chris (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Li, Ansong, Cheng, Zhiyong, Liu, Fan, Gao, Zan, Guan, Weili, & Peng, Yuxin (2022). Disentangled graph neural networks for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Li, Pan, & Tuzhilin, Alexander (2020). Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th international conference on web search and data mining* (pp. 331–339).

Liang, Dawen, Krishnan, Rahul G, Hoffman, Matthew D, & Jebara, Tony (2018). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference* (pp. 689–698).

Liu, Fan, Chen, Huilin, Cheng, Zhiyong, Liu, Anan, Nie, Liqiang, & Kankanhalli, Mohan (2022). Disentangled multimodal representation learning for recommendation. *IEEE Transactions on Multimedia*.

Liu, Meng, Li, Jianjun, Li, Guohui, & Pan, Peng (2020). Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In *Proceedings of the 29th ACM international conference on information and knowledge management* (pp. 885–894).

Ma, Jianxin, Zhou, Chang, Cui, Peng, Yang, Hongxia, & Zhu, Wenwu (2019). Learning disentangled representations for recommendation. *Advances in Neural Information Processing Systems*, *32*.

Man, Tong, Shen, Huawei, Jin, Xiaolong, & Cheng, Xueqi (2017). Cross-domain recommendation: An embedding and mapping approach. *Vol. 17*, In *Proceedings of the 26th international joint conference on artificial intelligence* (pp. 2464–2470).

Mao, Kelong, Zhu, Jieming, Xiao, Xi, Lu, Biao, Wang, Zhaowei, & He, Xiuqiang (2021). UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM international conference on information and knowledge management* (pp. 1253–1262).

McAuley, Julian, & Leskovec, Jure (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 165–172).

McAuley, Julian, Targett, Christopher, Shi, Qinfeng, & Van Den Hengel, Anton (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th ACM international conference on research and development in information retrieval* (pp. 43–52).

Mnih, Andriy, & Salakhutdinov, Russ R. (2007). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, *20*.

Najafabadi, Maryam Khanian, Mohamed, Azlinah, & Onn, Choo Wou (2019). An impact of time and item influencer in collaborative filtering recommendations using graph-based model. *Information Processing & Management*, *56*(3), 526–540.

Rezende, Danilo Jimenez, Mohamed, Shakir, & Wierstra, Daan (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 13th international conference on machine learning* (pp. 1278–1286).

Sarwar, Badrul, Karypis, George, Konstan, Joseph, & Riedl, John (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on world wide web* (pp. 285–295).

Singh, Ajit P., & Gordon, Geoffrey J. (2008). Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM international conference on knowledge discovery and data mining* (pp. 650–658).

Sung, Yi-Lin, Cho, Jaemin, & Bansal, Mohit (2022). Lst: Ladder side-tuning for parameter and memory efficient transfer learning. arXiv preprint arXiv:2206.06522.

Tao, Zhulin, Wei, Yinwei, Wang, Xiang, He, Xiangnan, Huang, Xianglin, & Chua, Tat-Seng (2020). Mgat: Multimodal graph attention network for recommendation. *Information Processing & Management*, *57*(5), Article 102277.

Wang, Xiang, He, Xiangnan, Wang, Meng, Feng, Fuli, & Chua, Tat-Seng (2019). Neural graph collaborative filtering. In *Proceedings of the 42nd ACM international conference on research and development in information retrieval* (pp. 165–174).

Wang, Xiang, Jin, Hongye, Zhang, An, He, Xiangnan, Xu, Tong, & Chua, Tat-Seng (2020). Disentangled graph collaborative filtering. In *Proceedings of the 43rd ACM international conference on research and development in information retrieval* (pp. 1001–1010).

Wang, Chenyu, Ye, Yawen, Ma, Liyuan, Li, Dun, & Zhuang, Lei (2023). Dual disentanglement of user-item interaction for recommendation with causal embedding. *Information Processing & Management*, *60*(5), Article 103456.

Wu, Bin, Zhong, Lihong, & Ye, Yangdong (2023). Graph-augmented social translation model for next-item recommendation. *IEEE Transactions on Industrial Informatics*.

Xi, Dongbo, Zhuang, Fuzhen, Zhu, Yongchun, Zhao, Pengpeng, Zhang, Xiangliang, & He, Qing (2020). Graph factorization machines for cross-domain recommendation. arXiv preprint arXiv:2007.05911.

Yan, Mingshi, Cheng, Zhiyong, Gao, Chen, Sun, Jing, Liu, Fan, Sun, Fuming, et al. (2022). Cascading residual graph convolutional network for multi-behavior recommendation. arXiv preprint arXiv:2205.13128.

Yang, Xun, Feng, Fuli, Ji, Wei, Wang, Meng, & Chua, Tat-Seng (2021). Deconfounded video moment retrieval with causal intervention. In *Proceedings of the 44th ACM international conference on research and development in information retrieval* (pp. 1–10).

Yang, Xun, Wang, Shanshan, Dong, Jian, Dong, Jianfeng, Wang, Meng, & Chua, Tat-Seng (2022). Video moment retrieval with cross-modal neural architecture search. *IEEE Transactions on Image Processing*, *31*, 1204–1216.

Yu, Xu, Peng, Qinglong, Xu, Lingwei, Jiang, Feng, Du, Junwei, & Gong, Dunwei (2021). A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm. *Information Processing & Management*, *58*(6), Article 102691.

Yu, Junliang, Yin, Hongzhi, Xia, Xin, Chen, Tong, Cui, Lizhen, & Nguyen, Quoc Viet Hung (2022). Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th ACM international conference on research and development in information retrieval* (pp. 1294–1303).

Zamir, Amir R, Sax, Alexander, Shen, William, Guibas, Leonidas J, Malik, Jitendra, & Savarese, Silvio (2018). Taskonomy: Disentangling task transfer learning. In *Proceedings of the 31th IEEE conference on computer vision and pattern recognition* (pp. 3712–3722).

Zhang, Shuai, Yao, Lina, Sun, Aixin, & Tay, Yi (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, *52*(1), 1–38.

Zhao, Wayne Xin, Chen, Junhua, Wang, Pengfei, Gu, Qi, & Wen, Ji-Rong (2020). Revisiting alternative experimental settings for evaluating top-n item recommendation algorithms. In *Proceedings of the 29th ACM international conference on information and knowledge management* (pp. 2329–2332).

Zhao, Cheng, Li, Chenliang, & Fu, Cong (2019). Cross-domain recommendation via preference propagation graphnet. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2165–2168).

Zhu, Feng, Chen, Chaochao, Wang, Yan, Liu, Guanfeng, & Zheng, Xiaolin (2019). Dtcdr: A framework for dual-target cross-domain recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1533–1542).

Zhu, Yongchun, Ge, Kaikai, Zhuang, Fuzhen, Xie, Ruobing, Xi, Dongbo, Zhang, Xu, et al. (2021). Transfer-meta framework for cross-domain recommendation to cold-start users. In *Proceedings of the 44th ACM international conference on research and development in information retrieval* (pp. 1813–1817).

Zhu, Yongchun, Tang, Zhenwei, Liu, Yudan, Zhuang, Fuzhen, Xie, Ruobing, Zhang, Xu, et al. (2022). Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the 15th ACM international conference on web search and data mining* (pp. 1507–1515).