PDF Download
3742855.pdf
28 December 2025
Total Citations: 1
Total Downloads: 818

Published: 15 July 2025
Online AM: 05 June 2025
Accepted: 20 February 2025
Revised: 20 October 2024
Received: 23 April 2024

RESEARCH-ARTICLE

Coherence-guided Preference Disentanglement for Cross-domain Recommendations

ZONGYI XIANG, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

YAN ZHANG, Charles Darwin University, Darwin, NT, Australia

LIXIN DUAN, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

HONGZHI YIN, The University of Queensland, Brisbane, QLD, Australia

IVOR WAI HUNG TSANG, Nanyang Technological University, Singapore City, Singapore

Open Access Support provided by:

The University of Queensland

University of Electronic Science and Technology of China

Nanyang Technological University

Charles Darwin University

ACM Transactions on Information Systems, Volume 43, Issue 4 (July 2025)
https://doi.org/10.1145/3742855
EISSN: 1558-2868

Coherence-guided Preference Disentanglement for Cross-domain Recommendations

ZONGYI XIANG, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
YAN ZHANG, Faculty of Science and Technology, Charles Darwin University, Darwin, Australia
LIXIN DUAN, Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China
HONGZHI YIN, The University of Queensland, Brisbane, Australia
IVOR W. TSANG, CFAR and IHPC, Agency for Science, Technology and Research, Singapore College of Computing and Data Science, Nanyang Technological University, Singapore

Discovering user preferences across different domains is pivotal in cross-domain recommendation systems, particularly when platforms lack comprehensive user-item interactive data. The limited presence of shared users often hampers the effective modeling of common preferences. While leveraging shared items' attributes, such as category and popularity, can enhance cross-domain recommendation performance, the scarcity of shared items between domains has limited research in this area. To address this, we propose a Coherence-guided Preference Disentanglement (CoPD) method aimed at improving cross-domain recommendation by (i) explicitly extracting shared item attributes to guide the learning of shared user preferences and (ii) disentangling these preferences to identify specific user interests transferred between domains. CoPD introduces coherence constraints on item embeddings of shared and specific domains, aiding in extracting shared attributes. Moreover, it utilizes these attributes to guide the disentanglement of user preferences into separate embeddings for interest and conformity through a popularity-weighted loss. Experiments conducted on real-world datasets demonstrate the superior performance of our proposed CoPD over existing competitive baselines, highlighting its effectiveness in enhancing cross-domain recommendation performance. The code is available at https://github.com/XiangZongyi/CoPD.

CCS Concepts: • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Cross-domain recommendation, preference disentanglement, graph collaborative filtering, user popularity

This work is supported by the National Natural Science Foundation of China (Grant No. 62002052), the Shenzhen Longhua Fundamental Research Program (Grant No. 10162A20230325B73A546), and the Start-up funding provided by Charles Darwin University. The Australian Research Council partially supports this work under the streams of Future Fellowship (Grant No. FT210100624), the Discovery Project (Grant No. DP240101108), and the Linkage Project (Grant No. LP230200892).
Authors' Contact Information: Zongyi Xiang, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China; e-mail: xzongyi@foxmail.com; Yan Zhang (corresponding author), Faculty of Science and Technology, Charles Darwin University, Darwin, Australia; e-mail: yan.zhang@cdu.edu.au; Lixin Duan (corresponding author), Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China; e-mail: lxduan@uestc.edu.cn; Hongzhi Yin, The University of Queensland, Brisbane, Australia; e-mail:h.yin1@uq.edu.au; Ivor W. Tsang, CFAR and IHPC, Agency for Science, Technology and Research, Singapore College of Computing and Data Science, Nanyang Technological University, Singapore; e-mail: ivor_tsang@cfar.a-star.edu.sg.

ACM 1558-2868/2025/7-ART109
https://doi.org/10.1145/3742855

ACM Reference format:
Zongyi Xiang, Yan Zhang, Lixin Duan, Hongzhi Yin, and Ivor W. Tsang. 2025. Coherence-guided Preference Disentanglement for Cross-domain Recommendations. *ACM Trans. Inf. Syst.* 43, 4, Article 109 (July 2025), 28 pages.
https://doi.org/10.1145/3742855

1 Introduction

The utilization of e-commerce platforms is steadily increasing as consumers seek the best products to meet their requirements. Recommender systems facilitate precise item selection based on sufficient user-item interactive data, user profiles, and item attributes. For instance, various recommender systems, taking Netflix for movies as an example, try to help users discover relevant movies based on users' past preferred movies. These systems commonly leverage user-item interactive data, including explicit or implicit ratings of users to movies, together with user reviews associated with particular movies to enhance the recommendation accuracy. Due to the limited number of user-item interactions, existing **Cross-domain Recommendations (CDRs)** generally transfer user preference information from other domains into a target domain.

CDR [2, 6, 25, 30, 41, 48, 62] provides a strategy to alleviate the issues caused by sparse user-item interaction data, which generally utilize transfer learning approaches [6, 19, 25] to transfer users' preference information from a source domain to a target domain by employing their shared users. For example, when a user views a movie, the CDR methods can recommend a book with the same theme as the movie in the Book domain because the CDR methods can capture the shared users' preferences. On the other hand, understanding a user's preference for books facilitates us in recommending related movies. However, the scarcity of shared users across two domains significantly constrains the performance of CDRs [25, 41]. Tang et al. [41] emphasized the challenge of sparse overlapping users between domains, hindering the effective transfer of knowledge. Similarly, Li et al. [26] highlighted domain shift and heterogeneity issues, where differences in item distributions and user behaviors across domains posed obstacles in aligning representations and transferring preferences. Fernandez et al. [6] noted that the cold-start problem, particularly affecting new user or item entries in cross-domain systems, exacerbates the challenge of accurate recommendations based on limited shared information. Recently, some dual-target CDR methods have successfully improved the results with sparse overlapping users [25, 56, 62, 63]. These methods integrate the information of different domains by combining the representations of overlapping users, establishing the connection between domains, and enhancing recommendation performance across multiple domains [53, 64]. Nevertheless, Zhu et al. [62] identified a potential loss of domain-specific information during shared preference transfer, risking oversimplification and excluding crucial domain-specific characteristics that could enhance accuracy. To tackle this problem, some dual-target CDR methods aim to disentangle domain-shared and domain-specific information [2, 66], thereby improving model transfer effectiveness.

Although the aforementioned dual-target CDR methods achieve promising results on the task with limited overlap among users, they mainly disentangled users' preferences based on domains (e.g., domain-shared and domain-specific), thus neglecting the users' inherent intention towards items. Recent research on disentangling users' intents (preferences) in a single domain has achieved more precise recommendations [37, 60]. This encourages us to investigate the disentanglement of domain-shared users' preferences. To this end, we disentangle the shared user preferences into "real interest" and "conformity." A user might click on an item due to conformity, disrupting the model's ability to learn the user's real interests. An illustrative example is depicted in Figure 1, demonstrating a user's clicking behavior across both the Movie

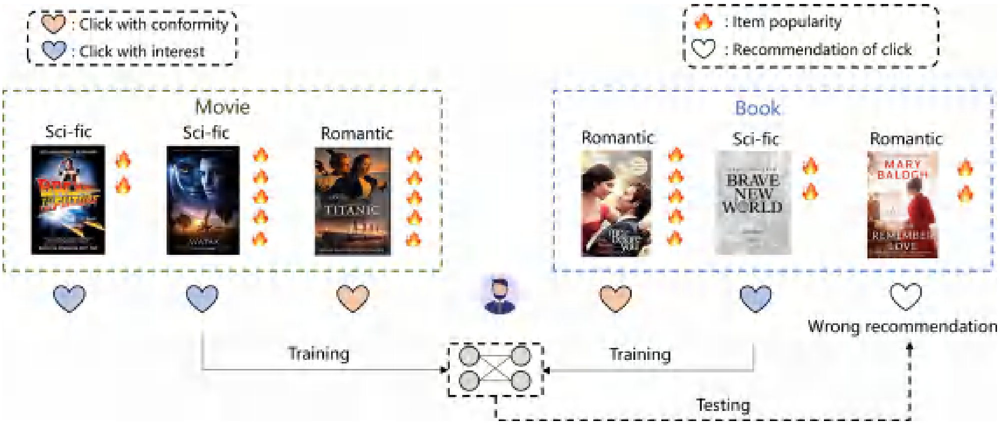


Fig. 1. An example of a user's click behavior across two domains. The recommendation model, trained on user preferences from both domains, incorrectly suggests a romantic book. The user clicks on romantic movies/books with his conformity (orange heart-shaped icon) but demonstrates genuine interest by clicking on science fiction movies/books (blue heart-shaped icon).

and Book domains. The user engages with romantic movies/books out of conformity rather than genuine interest. In fact, the user's true preference lies in scientific movies or books, irrespective of popularity. However, the recommendation model has learned the shared user preferences for science fiction and romance, leading to a recommendation for an unpopular romantic book.

Disentangling user interest and conformity preferences is challenging due to the sparse overlap among users. Previous dual-target CDR methods primarily emphasized transferring shared user preferences, while neglecting locating which specific preferences are transferred, thereby limiting performance. For example, the recommendation shown in Figure 1 overlooks the user's shared conformity, such as when a user clicks on romantic books and movies due to their popularity rather than romantic attributes. By disentangling users' preferences, we can pinpoint which specific preferences are transferred across two domains. Some sequential recommendation methods utilize mixed sequence data [50, 51] and others construct a **Cross-domain Sequence (CDS)** graph that links different domains [10, 63] to obtain domain-shared embeddings. This approach inspired us to acquire domain-shared item attributes using global cross-domain interaction data. Additionally, if shared item attributes (category, popularity, etc.) exist between two domains, the performance of CDR would be greatly improved by transferring both shared user preferences and item attributes. Therefore, we can utilize the learned shared item attributes to guide the learning process of user interest preferences and conformity preferences. Motivated by this intuition, we devise a global-local graph encoder framework to learn the domain-shared and domain-specific embeddings. We aim to learn the shared item attributes through a cross-domain global graph G^S, where information can be propagated across both domains. Additionally, we utilize coherence constraints across the two domains to eliminate redundant item attributes. Specifically, Figure 2 illustrates a cross-domain global graph denoted as G^S and a single-domain local graph denoted as G^B. In G^S, each edge represents an interaction (rating, click, etc.) of a user depicted as an orange circle) with an item in a shared domain S, comprising items from both Book and Movie domains. Meanwhile, edges of G^B denote user-item interactions solely within the Movie domain. Items in the Book domain (A) are denoted as blue rectangles, and items in the Movie domain (B) are denoted as green triangles. For an item i in the Movie domain, we can extract the item embeddings z_i^B from the Movie domain

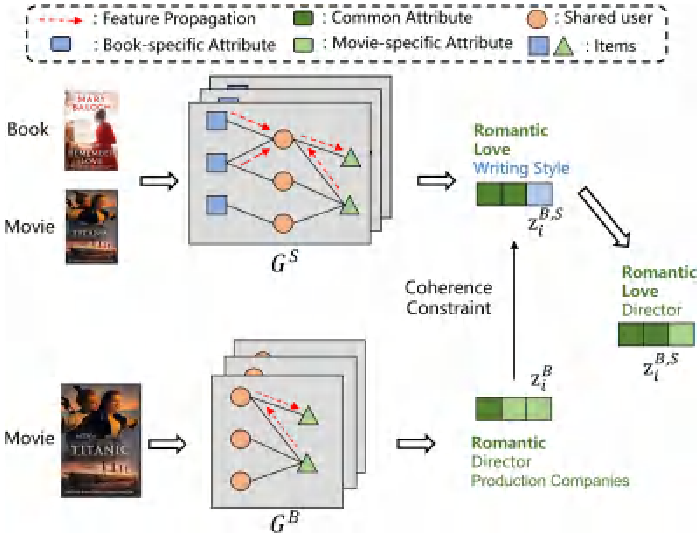


Fig. 2. Domain-shared item attributes extracted by coherence constraints. Dark green, blue, and green represent the shared, book-specific, and movie-specific attributes, respectively.

and $z_i^{B,S}$ from the shared domain via **Graph Neural Network (GNN)**, where $z_i^{B,S}$ carries attribute information from both Movie and Book domains through multi-layer information propagation. To keep domain-shared attributes and remove redundant information in $z_i^{B,S}$, we enforce coherence constraints between item embeddings z_i^B and $z_i^{B,S}$ and attain an enhanced domain-shared item embedding.

This article proposes a **Coherence-guided Preference Disentanglement (CoPD)** method to improve CDR performance by extracting shared item attributes and applying them to guide preference disentanglement. Firstly, to learn shared item attributes, we generate the domain-shared item embeddings and enforce coherence constraints on items' domain-specific (source or target domain) and domain-shared embeddings. The coherence constraints in the proposed CoPD emphasize boosting item attribute coherence between a domain-shared item embedding and a domain-specific item embedding. Specifically, Figure 2 shows the coherence constraints imposed on the two embeddings $z_i^{B,S}$ and z_i^B is capable of learning an enhanced shared item embedding including domain-shared attributes and domain-specific attributes. Secondly, the learned enhanced shared item embedding guides the preference disentanglement. Specifically, shared user preferences are disentangled as the user's real interest and conformity, and we learn their embeddings respectively by employing a popularity-weighted loss. Finally, we integrate the domain-shared embeddings and domain-specific embeddings to form comprehensive user and item embeddings. The main contributions are summarized as follows:

- We propose a CoPD framework designed to improve the performance of CDRs to tackle the challenge of scarcity shared users between domains. CoPD consists of three components: coherence-guided item attribute alignment, user preference disentanglement guided by the aligned attribute, and embedding fusion and recommendation.
- To learn comprehensive item embedding, we propose a global-local graph framework, and add coherence constraints on the embeddings of specific domains and shared domain (the union of two domains) to extract shared item attributes; meanwhile, the comprehensive item

embedding keeps essential domain-specific information and drops redundant information in another domain.

- To disentangle shared user preferences into the user's real interest and conformity, we construct a popularity-weighted loss to facilitate this learning process and employ comprehensive item embedding to guide the process of preference disentanglement.
- We design experiments to demonstrate the effectiveness of our proposed CoPD. The results show the superior performance of our method over state-of-the-art baselines. Additionally, we conduct ablation studies to evaluate the effectiveness of coherence constraints and preference disentanglement.

2 Related Work

2.1 CDRs

The CDR aims to augment the recommendation effectiveness within the target domain by leveraging insights from other domains. Based on the inference goal of recommendation task, CDR methodologies can be roughly categorized into three categories [29]: single-target domain [33, 67], dual-target domain [2, 11, 25, 30, 63, 66], and multi-target domain [5, 22]. Our CoPD belongs to dual-target CDRs, focusing on transferring knowledge across two domains to improve recommendation performance in both domains. Dual-target CDR models emphasize identifying overlapping entities across disparate fields and achieving integration by sharing or enhancing the representations of such entities in distinct fields. Some approaches such as CDFM [31] employ shallow models as a backbone. CDFM aims to proficiently transfer knowledge by incorporating source domain interactions as supplementary features, based on the principles of Factorization Machines [38]. Furthermore, other approaches such as CoNet [18] utilize neural networks as a backbone. CoNet facilitates knowledge transfer across domains through the use of cross-connection networks between feed-forward multi-level perceptrons. In a different view, DDTCDR [25] pursues the learning of a latent orthogonal mapping function for shared users, enabling the transfer of user preferences across domains. Some methods leverage GNNs as a backbone, achieving state-of-the-art performance. For example, PPGN [56] involves stacking several graph convolution networks on two domains and achieves knowledge transfer by sharing common hidden user features. Similarly, BiTGCF [30] introduced a graph collaborative filtering network, leveraging public users as conduits to facilitate bidirectional cross-domain knowledge transfer, thereby enhancing performance across both domains. DA-GCN [10] and TiDA-GCN [12] construct a CDS graph to model complex associations among users and items across domains, where users and items from each domain are linked as nodes in a graph, with their associations represented as edges. Additionally, they are domain-aware in the sense that they adapt to the specific preferences and behaviors of users within each domain. However, previous methods neglect the disentanglement of domain-shared and domain-specific information, leading to the problem of negative transfer [61, 65]. Wang et al. [44] utilize the explicit reconstruction loss integrated within the Evidence Lower Bound, complemented by additional regularizers such as the disentanglement loss, to learn the desirable disentangled representations. DisenCDR [2] proposes a **Variational Bipartite Graph Encoder (VBGE)**, and use different VBGEs encode domain-shared and domain-specific representations. Then, DisenCDR learns these representation with two disentanglement regularizers based on mutual information. DIDA-CDR [66] captures the domain-shared user representations using an interpolative data augmentation approach and proposes a disentangling module to disentangle domain-specific and domain-independent information to capture comprehensive user preferences.

2.2 Disentangled Representation Learning

Disentangled representation learning, initially introduced within the domain of computer vision [9], primarily extracts visual characteristics, including shape, color, and spatial attributes of objects. Beyond its application in computer vision, disentangled representation learning has increasingly found utility within recommender systems in recent years. Disentangled representation learning in the recommender system focuses on factorizing unobservable structural factors, including domain-based features, item attributes, and user preferences towards items. The β-VAE [17] based disentanglement model proposed by Nema et al. [35] can identify user tastes by concerning disentangled attributes, enabling users to manipulate recommendations across these attributes. DGCF [46] focuses on uncovering users' hidden intentions by disentangling the interaction graph into different intent sub-graphs. It segments the embedding of users or items into multiple blocks corresponding to decoupled intentions. DICE [60] disentangles the embedding representation into specific intents, distinguishing whether users purchase items based on their own interests or blind obedience. DICE established an inverse relationship between interest and popularity, in order to effectively learn interest embeddings. From a different view of conformity, Zhao et al. [59] disentangle item attributes into quality (benign conformity) and conformity (harmful conformity), and remove the harmful conformity during inference. Previous approaches mainly segment embeddings into different intent components or assign separate embeddings to users and items to achieve disentanglement. The authors in [4, 23, 36] validate its capability to disentangle embeddings through a specifically designed disentanglement module. Qian et al. [37] propose an intent prototype strategy to model popularity and preference intents separately, and learn these intent representations using a novelty-weighted recommendation loss.

2.3 Differences

Our CoPD differs significantly from previous studies in several aspects: (1) Before disentangling user preference, we introduce a global-local graph encoder to disentangle domain-shared and domain-specific information. Different from DisenCDR [2] and DIDA-CDR [66], CoPD learns domain-shared user representations through a global graph encoder, allowing information from both domains to influence the user representation. Additionally, we can learn domain-shared item representations using the global graph encoder and two coherence constraint regularizers, which have not been explored in previous dual-target CDR methods. Unlike PPGN [56], which constructs a cross-domain user-item interaction graph to learn user and item representations, CoPD acquires domain-specific user representations through a local graph encoder. This approach allows CoPD to address the negative transfer problem and ensure independent learning of domain-specific representations. (2) Our preference disentanglement strategy differs from other disentangled representation learning methods. While previous methods aim to disentangle both user and item embeddings [46, 60], accomplishing this in dual-target CDR tasks presents challenges due to the data sparsity problem. Hence, CoPD focuses on disentangling user embeddings into interest and conformity embeddings and disentangling user and item embeddings into domain-shared and domain-specific embeddings. The learning of users' interest and conformity embeddings is guided by disentangled domain-shared item embeddings. Specifically, to learn domain-shared user preferences, CoPD disentangles domain-shared user embeddings into domain-shared interest and conformity embeddings, which are guided by domain-shared item attribute embeddings. Furthermore, CoPD employs two preference encoders to disentangle preference embeddings. However, it is difficult to sample negative samples in CDR [32]. To facilitate the effective training of these preference encoders, we employ a popularity-weighted Softmax loss, which enables the selection of challenging examples from multiple negative. Based on the causality analysis in DICE [60], although user interests can

Table 1. The Important Notations

Symbol	Definition
A, B	Domain A and domain B.
$\mathcal{U}$	The shared user set.
$\mathcal{I}^A, \mathcal{I}^B$	The item sets in domain A and domain B.
$\mathcal{R}^A, \mathcal{R}^B$	The binary interaction matrices in domain A and domain B.
G^A, G^B, G^S	The graphs constructed by interaction matrices in domain A, domain B, and shared domain.
$\mathcal{N}_u$	The item set interacted by user u.
$\mathcal{D}^A_{train}, \mathcal{D}^B_{train}$	The training datasets in domain A and domain B.
$\mathcal{D}^A_{test}, \mathcal{D}^B_{test}$	The test datasets in domain A and domain B.
H^A_u, H^B_u, H^S_u	The initialized user embedding matrices in domain A, domain B, and shared domain.
H^A_i, H^B_i	The initialized item embedding matrices in domain A and domain B.
$h^*_{u/i}$	The sampled embedding from corresponding embedding matrix $H^*_{u/i}$.
$z^A_{u/i}, z^B_{u/i}, z^S_{u/i}$	The user/item graph embeddings propagated by G^A, G^B, G^S.
$z^{A,S}_i, z^{B,S}_i$	The shared item embeddings in domain A and domain B split from z^S_i.
$e^A_{u/i}, e^B_{u/i}$	The integrated user/item embeddings in domain A and domain B.

indeed align with popular items, it is difficult to learn real interests from popular items. We assert an inverse correlation between interest embeddings and popularity weight to learn users' real interests.

3 Methods

3.1 Preliminaries

In this section, we first describe the notations used in our article. We list the important notations of this article in Table 1 for better readability. For dual-target CDR, there are two domains, A and B, with a shared user set $\mathcal{U}$. The item sets in domain A and domain B are denoted as $\mathcal{I}^A$ and $\mathcal{I}^B$, respectively. Additionally, two binary interaction matrices are defined as $\mathcal{R}^A \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}^A|}$ and $\mathcal{R}^B \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}^B|}$ in domain A and domain B, where $|\mathcal{U}|$ is the size of the user set and $|\mathcal{I}^A|/|\mathcal{I}^B|$ are the size of the item sets of domains A and B. We embed the user set $\mathcal{U}$ with $H^A_u \in \mathbb{R}^{|\mathcal{U}|\times d}$, $H^B_u \in \mathbb{R}^{|\mathcal{U}|\times d}$, and $H^S_u \in \mathbb{R}^{|\mathcal{U}|\times d}$ as domain-A-specific, domain-B-specific, and domain-shared initialized embedding matrices, and d denotes the embedding dimension. Meanwhile, we embed the item sets with $H^A_i \in \mathbb{R}^{|\mathcal{I}^A|\times d}$ and $H^B_i \in \mathbb{R}^{|\mathcal{I}^B|\times d}$ in domain A and domain B, respectively. Note that we use uppercase (e.g., H^A_u) to represent entire embedding matrices and lowercase letters (e.g., h^A_u) to represent sampled embeddings from embedding matrices.

The domain-based disentanglement and preference-based disentanglement are used in our method. For domain-based disentanglement, we disentangle user embeddings into domain-shared embeddings z^S_u and domain-specific embeddings z^A_u/z^B_u. Different from previous methods, we also disentangle item embeddings into domain-shared embeddings $z^{A,S}_i/z^{B,S}_i$ and domain-specific embeddings z^A_i/z^B_i. In our dual-target CDR task, the items in domains A and B do not overlap. Here, $z^{A,S}_i$ denotes the item embeddings within domain A, capturing shared attributes across both domains and the same for $z^{B,S}_i$. For preference-based disentanglement, we define user interest embeddings as z^{int}_u and user conformity embeddings as z^{conf}_u. Finally, we use e^A_u/e^B_u and e^A_i/e^B_i to represent the fused user and item embeddings in domain A of B.

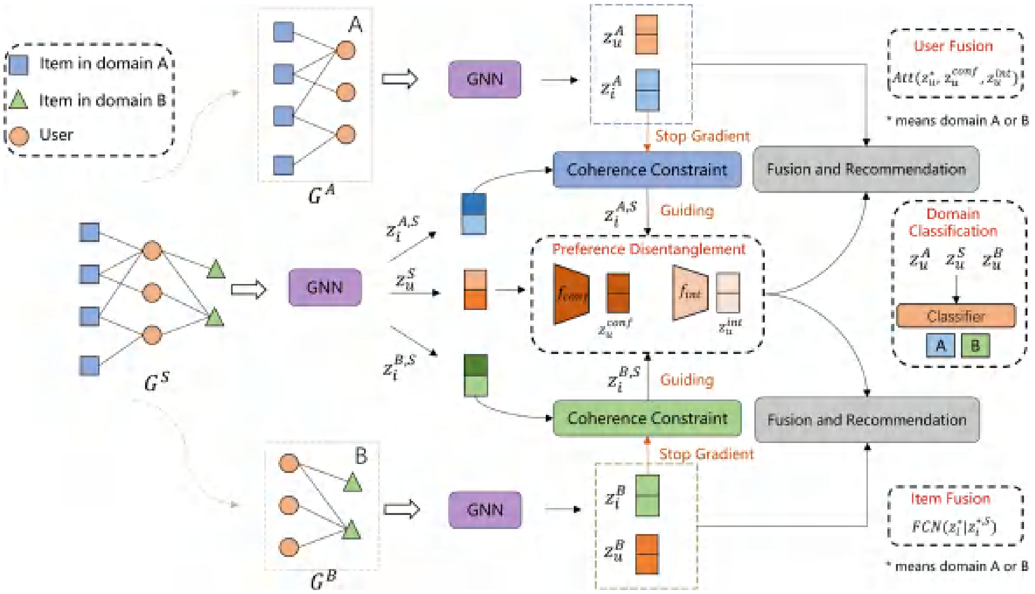


Fig. 3. The CoPD framework comprises three primary components: Coherence-guided Item Attribute Alignment, User Preference Disentanglement guided by Aligned Attribute, and Embedding Fusion and Recommendation. The process begins by generating graph-enhanced embeddings using three GNN models. Subsequently, we align the item attribute embeddings and disentangle user preference embeddings. Finally, we fuse the domain-shared user/item embeddings with their corresponding domain-specific embeddings to generate comprehensive recommendations.

3.2 Overall Framework

In this section, we present a novel approach called CoPD, designed for dual target CDRs, as depicted in Figure 3. Firstly, we introduce the graph collaborative filtering backbone in our CoPD to obtain graph embeddings for users and items. The details can be seen in Section 3.3. Next, we introduce CoPD's three components: Coherence-guided Item Attribute Alignment, User Preference Disentanglement, and Embedding Fusion and Recommendation:

(1) *Coherence-guided Item Attribute Alignment*. To acquire domain-shared embeddings for users and items, we create a global graph connecting domain A and domain B. This facilitates the propagation of feature information between the two domains, leveraging overlapping users as connectors. Furthermore, to ensure precise domain-shared item embeddings, we apply coherence constraints that align them with domain-specific item embeddings. Their coherence constraints assist in enabling domain-shared item embeddings to capture shared attributes while filtering out unnecessary ones comprehensively. The details can be seen in Section 3.4.

(2) *User Preference Disentanglement*. We disentangle domain-shared user embeddings into interest embeddings with an interest encoder and conformity embeddings with a conformity encoder. To disentangle user preference effectively, we utilize a popularity-weighted loss function to train interest and conformity encoders. Specifically, we establish a positive correlation between conformity preferences and popularity, and a negative correlation between interest preferences and popularity. In addition, we feed the learned domain-shared item embeddings

into the popularity-weighted loss to facilitate the disentanglement of user preferences. The details can be seen in Section 3.5.

(3) *Embedding Fusion and Recommendation*. Both domain-shared and domain-specific information are important for recommendations. To integrate this information for users, we employ an attention mechanism to fuse domain-shared interest and conformity embeddings with domain-specific embeddings. Similarly, for items, we concatenate the domain-shared embeddings with corresponding domain-specific embeddings and feed them into a fully connected network. Finally, we employ the inner product operation for integrated user and item embeddings to make recommendations. The details can be seen in Section 3.6.

3.3 Graph Collaborative Filtering Backbone

GNN-based collaborative filtering methods [14, 28, 45] fundamentally aggregate neighboring node features to enhance the overall representation for users and items within the graph. Generally, the aggregation scheme is formulated as follows:

$$h_u^{(l+1)} = Agg(h_u^{(l)}, h_i^{(l)}; i \in \mathcal{N}_u), \tag{1}$$

$$z_u = Readout([h_u^{(0)}, h_u^{(1)}, ..., h_u^{(L)}]), \tag{2}$$

where $Agg(\cdot)$ represents an aggregation function, such as a weighted sum aggregator or mean aggregator, and $\mathcal{N}_u$ denotes the neighbor set of user u in the interaction matrix. The $h_u^{(l)}$ and $h_i^{(l)}$ denote the refined embeddings for user u and item i after l layers of information propagation, where h_u^0 and h_i^0 are sampled from two learnable embedding matrices H_u and H_i. The L denotes the number of GNN layers and the readout function $Readout(\cdot)$ further encodes all representations of $[h_u^{(0)}, h_u^{(1)}, \ldots, h_u^{(L)}]$ to acquire the ultimate user u representations for recommendation. Similarly, the informative representations of items can be acquired analogously.

In CoPD, we produce user and item representations by applying GNN-based methods. Following LightGCN [14], we remove the nonlinear activation and feature transformation in Equation (1) as:

$$\begin{aligned} h_u^{(l+1)} &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} h_i^{(l)}, \\ h_i^{(l+1)} &= \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} h_u^{(l)}, \end{aligned} \tag{3}$$

where $\frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}$ serves as a symmetric normalization term, addressing the concern of embedding scale expansion resulting from graph convolution operations. After the propagation through L layers, we employ the average sum function in Equation (2) to encode the representations generated across all layers and obtain the conclusive representations as follows:

$$\begin{aligned} z_u &= \frac{1}{(L+1)}(h_u^{(0)} + h_u^{(1)} + ... + h_u^{(L)}), \\ z_i &= \frac{1}{(L+1)}(h_i^{(0)} + h_i^{(1)} + ... + h_i^{(L)}). \end{aligned} \tag{4}$$

3.4 Coherence-guided Item Attribute Alignment

To acquire domain-shared and domain-specific representations, we create global and local graphs using the corresponding user-item interaction matrices. Specifically, we create a global interaction matrix $\mathcal{R}^S$, where $\mathcal{R}^S \in \{0,1\}^{|\mathcal{U}| \times (|\mathcal{I}^A| + |\mathcal{I}^B|)}$ is formed by concatenating $\mathcal{R}^A$ and $\mathcal{R}^B$ ($\mathcal{R}^S$=$\mathcal{R}^A \parallel \mathcal{R}^B$).

The construction of global and local graphs can be written as follows:

$$G^* = \begin{pmatrix} 0 & \mathcal{R}^* \\ \mathcal{R}^{*\top} & 0 \end{pmatrix}, \tag{5}$$

where G^* (G^A, G^B, or G^S) is constructed by corresponding interaction matrices $\mathcal{R}^*$ ($\mathcal{R}^A, \mathcal{R}^B$, or $\mathcal{R}^S$).

After that, we denote the GNN backbone as f_{gnn} and domain-shared and domain-specific embeddings of user/item can be acquired by:

$$z_u^S, z_i^S = f_{gnn}(h_u^S, h_i^A \| h_i^B, G^S), \tag{6}$$

$$\begin{aligned} z_u^A, z_i^A &= f_{gnn}(h_u^A, h_i^A, G^A), \\ z_u^B, z_i^B &= f_{gnn}(h_u^B, h_i^B, G^B), \end{aligned} \tag{7}$$

and please note that we concatenate the origin item embeddings $h_i^A \| h_i^B \in \mathbb{R}^{(|\mathcal{I}^A|+|\mathcal{I}^B|)\times d}$ to facilitate enhanced information propagation for items across the two domains. After the information aggregation and readout of GNN, we obtain the domain-shared item embeddings z_i^S. We then split z_i^S into $z_i^{A,S}$ for domain A and $z_i^{B,S}$ for domain B.

The domain-shared attributes $z_i^{A,S}$ and $z_i^{B,S}$ can by propagated by the common user. However, the domain-shared item embeddings may incorporate redundant domain-specific attributes, which could be detrimental for recommendations in both domains. To solve this problem, our CoPD introduces coherence constraints to maintain alignment between the domain-shared item embeddings ($z_i^{A,S}$ and $z_i^{B,S}$) and the corresponding domain-specific item embeddings (z_i^A and z_i^B). Furthermore, taking domain A as an example, we freeze the parameters of z_i^A and introduce coherence constraints to $z_i^{A,S}$ and z_i^A. This process aims to eliminate irrelevant features from domain B in $z_i^{A,S}$. Specifically, we utilize cosine similarity to enforce coherence constraints between domain-shared and domain-specific embeddings. Cosine similarity is chosen because it can capture both magnitude and angle similarity. We consider the item i clicked by the user u ($i \in \mathcal{N}_u$) as the positive sample. In contrast, the item j not clicked by user u ($j \notin \mathcal{N}_u$) as the negative sample. These constraints are applied to both domain A and domain B. Taking domain A as an example, the proposed coherence constraints learning objective is to minimize the following function:

$$\mathcal{L}_{cc}^A = 1 - \frac{z_i^{A,S} \cdot SG(z_i^A)}{\|z_i^{A,S}\|\|z_i^A\|} + \max(0, \frac{z_i^{A,S} \cdot SG(z_j^A)}{\|z_i^{A,S}\|\|z_j^A\|}), \tag{8}$$

where $SG(\cdot)$ denotes the stop gradient operation. Specifically, the stop-gradient facilitates the independent learning of item attributes and prevents extraneous noise inherited from domain-shared item embedding. Similarly, the learning objective of domain B can be acquired analogously.

As mentioned above, we utilize coherence constraints to remove the redundant information in domain-shared item embeddings by aligning them with domain-specific item representations. Specifically, we designate the domain-specific embeddings as the anchor and adjust the domain-shared embeddings to align with this anchor. The objective of the coherence constraints is formally expressed as follows:

$$\mathcal{L}_{cc} = \mathcal{L}_{cc}^A + \mathcal{L}_{cc}^B. \tag{9}$$

3.5 User Preference Disentanglement Guided by Aligned Item Attributes

This section focuses on disentangling domain-shared user embeddings into interest and conformity embeddings with the assistance of domain-shared item embeddings. Firstly, motivated by [7, 24], we introduce a domain classifier denoted as $f_{dom}(\cdot)$ to ensure that the GNN backbone learns domain-shared and domain-specific user preferences. This classifier incorporates a single fully connected layer to predict the domain probability associated with user embeddings. Following Meta-FDMixup [7], we define the ground truth for domain-specific user embeddings as y_A and y_B for their respective domains. The ground truth for domain-shared user embeddings is denoted as y_S. We hope that the learned domain-specific user embeddings (e.g., z_u^A and z_u^B) can be easily categorized as a particular domain and the learned domain-shared user embeddings z_u^S cannot be classified into either domain A or domain B. Formally, let one and zero denote the label of domain A and domain B, respectively. All values in y_A are set as one, and all in y_B are set as zero. In particular, all entries in y_S are set as 0.5, because z_u^S contains common user preferences information across both domain A and domain B. The objective of the domain classifier is formulated as follows:

$$\begin{aligned}\mathcal{L}_{dom} = \frac{1}{3}\sum & KL(f_{dom}(z_u^S), y_S) \\ & + CE(f_{dom}(z_u^A), y_A) + CE(f_{dom}(z_u^B), y_B),\end{aligned}$$

where the $KL(\cdot,\cdot)$ and the $CE(\cdot,\cdot)$ denote the Kullback-Leibler divergence loss and cross-entropy loss, respectively. The domain classifier assists in capturing domain-shared user preferences for z_u^S.

Subsequently, the learned domain-shared user embedding is fed into a preference extractor to extract the domain-shared conformity and interest preferences. We implemented our preference extractor with two encoders, i.e., conformity encoder denoted as f_{conf} and interest encoder denoted as f_{int}. For simplicity, both of them are implemented as two-layer fully connected layers. We then disentangle the domain-shared user embeddings into two preference embeddings based on the outputs of these two preference encoders:

$$\begin{aligned} z_u^{int} &= f_{int}(z_u^S), \\ z_u^{conf} &= f_{conf}(z_u^S), \end{aligned} \tag{10}$$

where z_u^{int} representing interest embeddings and z_u^{conf} representing conformity embeddings.

After obtaining the disentangled preference embeddings, we hope z_u^{int} and z_u^{conf} reflect the preferences of user interest and conformity. We introduced the domain-shared item embeddings as mentioned in Section 3.4 (i.e., $z_i^{A,S}$ and $z_i^{B,S}$), which encompasses common attributes for domain A and domain B, to facilitate the representation learning for disentangled preference embeddings. **Bayesian Personalized Ranking (BPR)** loss [40] and Softmax loss [39] can be used to measure the quality of interaction predictions. We employ the Softmax loss here as it inherently facilitates the selection of challenging examples from multiple negative [27, 52, 54, 55], thus amplifying the process of learning particular user preference. There are two Softmax losses (i.e., conformity loss $\mathcal{L}_{conf}$ and interest loss $\mathcal{L}_{int}$) for learning the interest and conformity preferences. However, if the losses are simply summed without distinction for different items, the leaned representation z_u^{int} and z_u^{conf} cannot accurately reflect the preferences for user interest and conformity over items. Therefore, we integrate item popularity signal as a weight into the Softmax losses. Item popularity p_i is defined as the number of historical interactions that an item i has been involved in. A more popular item i has a higher item popularity p_i. We then normalize p_i to a range of 0 to 1 using min-max normalization. We aim to establish a positive correlation between the conformity loss $\mathcal{L}_{conf}$ and the popularity weight p_i. For highly popular items, the weight p_i in the conformity $\mathcal{L}_{conf}$ would be close to 1, ensuring that popular items are primarily attributed to user's conformity preference z_u^{conf}. Conversely, we aim to establish a negative correlation between the interest loss $\mathcal{L}_{int}$ and the popularity weight p_i. We use $1 - p_i$ as the weight of $\mathcal{L}_{int}$. For less popular items, the weight $1 - p_i$ would be close to 1, allowing the model to capture the users' interest preference z_u^{int}

for certain item attributes. For more popular items, the weight would be close to 0, thereby reducing the influence of item popularity and enabling the model to learn the user's genuine interests. Taking domain A as an example, popularity weight in Softmax loss can be defined as p_i^A. We minimize the popularity-weighted Softmax losses for representative learning:

$$\mathcal{L}_{conf}^{A} = -\sum_{(u,i)\in\mathcal{D}_{train}^{A}} p_i^A \log \frac{exp(z_u^{conf} \cdot z_i^{A,S}/\tau)}{\sum_{k\in\mathcal{I}^A} exp(z_u^{conf} \cdot z_k^{A,S}/\tau)}, \tag{11}$$

$$\mathcal{L}_{int}^{A} = -\sum_{(u,i)\in\mathcal{D}_{train}^{A}} (1-p_i^A) \log \frac{exp(z_u^{int} \cdot z_i^{A,S}/\tau)}{\sum_{k\in\mathcal{I}^A} exp(z_u^{int} \cdot z_k^{A,S}/\tau)}, \tag{12}$$

where τ represents the temperature hyper-parameter of Softmax [43] and $\mathcal{D}_{train}^{A}$ represents the training set in domain A. Similarly, we use p_i^B to represent item popularity in domain B and $\mathcal{D}_{train}^{B}$ represents the training set in domain B. The weighted Softmax losses in domain B can be calculated as follows:

$$\mathcal{L}_{conf}^{B} = -\sum_{(u,i)\in\mathcal{D}_{train}^{B}} p_i^B \log \frac{exp(z_u^{conf} \cdot z_i^{B,S}/\tau)}{\sum_{k\in\mathcal{I}^B} exp(z_u^{conf} \cdot z_k^{B,S}/\tau)}, \tag{13}$$

$$\mathcal{L}_{int}^{B} = -\sum_{(u,i)\in\mathcal{D}_{train}^{B}} (1-p_i^B) \log \frac{exp(z_u^{int} \cdot z_i^{B,S}/\tau)}{\sum_{k\in\mathcal{I}^B} exp(z_u^{int} \cdot z_k^{B,S}/\tau)}. \tag{14}$$

We continue to utilize z_u^{int} and z_u^{conf} in Equations (13) and (14) as user embeddings in domain B. We intend these embeddings to extract common interest and conformity preferences shared across domains A and B. Finally, the objective of User Preference Disentanglement is formulated as follows:

$$\mathcal{L}_{pd} = \mathcal{L}_{conf}^{A} + \mathcal{L}_{int}^{A} + \mathcal{L}_{conf}^{B} + \mathcal{L}_{int}^{B}. \tag{15}$$

3.6 Embedding Fusion and Recommendation

In this section, we introduce the fusion mechanism for combining domain-specific and domain-shared features. Additionally, we introduce the prediction layer to make recommendations based on the fused features. From Sections 3.4 and 3.5, we have obtained the learned domain-specific embeddings of users (i.e., z_u^A and z_u^B) and items (i.e., z_i^A and z_i^B), along with the learned domain-shared embeddings of the users (i.e., z_u^{conf} and z_u^{int}) and the items (i.e., $z_i^{A,S}$ and $z_i^{B,S}$). These embeddings enable us to capture both domain-specific and domain-shared preferences, facilitating more accurate and effective recommendations across domains. Taking domain A as an example, we emphasize the importance of both the domain-specific embeddings z_i^A and the domain-shared embeddings $z_i^{A,S}$. Each of these embeddings contains distinct yet complementary portions of item attributes. To aggregate the item attributes in domains A and B, we concatenate the domain-shared embeddings with the corresponding domain-specific embeddings. Subsequently, we use one-layer fully connected layers, denoted as FCN_A and FCN_B ($\mathcal{R}^{2d} \rightarrow \mathcal{R}^{d}$), to fuse the information of embeddings. This transformation allows for effectively integrating domain-specific and domain-shared attributes, resulting in a comprehensive representation of items in each domain. Taking domain A as an example:

$$e_i^A = FCN_A(z_i^A \| z_i^{A,S}), \tag{16}$$

where || denotes the concatenation operation. Users' preferences for interest and conformity can vary across different scenarios and domains. For example, a user might be open to all popular movies in the Movie domain but have a unique interest in music in the Music domain. Additionally, users may exhibit preferences for items based on either learned domain-shared preferences or learned domain-specific preferences. To adapt users' multiple preferences towards different domains, we employ domain-aware attention functions [10, 63] denoted as Att_A and Att_B ($\mathcal{R}^{3d} \rightarrow \mathcal{R}^d$). The attention function aggregates individual preference representations into comprehensive user representations, enabling the incorporation of diverse user preferences. By attentively weighing the contributions of domain-specific, conformity, and interest embeddings, the function effectively captures the refinement preferences of users across different domains and scenarios. The attention function in domain A can be written as follows:

$$e_u^A = Att_A(z_u^A, z_u^{conf}, z_u^{int}). \tag{17}$$

After obtaining the final embeddings of the user and item, we employ the inner product operation to predict the probability that a user u interacts with an item i:

$$\hat{y}_{ui}^A = (e_u^A)^T \cdot e_i^A. \tag{18}$$

To learn representations directly from interactions, we employ the BPR loss [40] for the recommendation task. The BPR loss is a pairwise loss function that encourages the model to predict observed user-item interactions to have higher scores compared to unobserved interactions:

$$\mathcal{L}_{rec}^A = -\sum_{u=1}^{|\mathcal{U}|} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma(\hat{y}_{ui}^A - \hat{y}_{uj}^A). \tag{19}$$

By minimizing the BPR loss in Equation (19), the positive pairs (u and i) have higher rating scores than the negative pairs (u and j). Similarly, we perform the same optimization function in domain B. The recommendation loss can be summarized as follows:

$$\mathcal{L}_{rec} = \mathcal{L}_{rec}^A + \mathcal{L}_{rec}^B + \beta||\Theta||^2, \tag{20}$$

where β controls the L_2 regularization of the GNN backbone parameters Θ. Finally, we utilize a multi-task learning mechanism to train our CoPD by minimizing the following cross-domain objectives:

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{dom} + \lambda_1 \mathcal{L}_{cc} + \lambda_2 \mathcal{L}_{pd}, \tag{21}$$

where λ_1 and λ_2 are hyper-parameters to balance the weights assigned to the coherence constraints loss and preference disentanglement loss, respectively. The complete workflow of our CoPD is shown in Algorithm 1.

4 Experiments

In this work, we claim that (1) the global graph encoder and coherence constraints can help extract domain-shared attributes for item embedding and (2) our CoPD can disentangle users' interest and conformity embeddings guided by domain-shared item embedding. To verify the effectiveness of our CoPD, we conduct extensive experiments on six CDR tasks. Specifically, we provide comprehensive analyses from the subsequent perspectives:

- *RQ1*: Does our method outperform the state-of-the-art cross-domain baselines?
- *RQ2*: Are the aligned item embeddings learned by the coherence constraints effective for recommendation performance?

Algorithm 1: The CoPD Algorithm

```
Input: Training interaction data $\mathcal{D}^A_{train}$ and $\mathcal{D}^B_{train}$, test interaction data $\mathcal{D}^A_{test}$ and $\mathcal{D}^B_{test}$;
       Constructed graph $G^A, G^B, G^S$ of domain A, domain B and shared domain; The
       number of epochs, batch size, learning rate, and the hyper-parameters of CoPD.
Output: The learned user/item embedding matrices, the parameters of two encoders for
        disentangling; The predicted score between user and item for recommendation.
Initialize the embedding matrices of users ($H^A_u, H^B_u, H^S_u$) and items ($H^A_i, H^B_i$);
// Training Procedure
for each epoch do
    for $(u, i^A, i^B)$ in ($\mathcal{D}^A_{train}$ and $\mathcal{D}^B_{train}$) do
        Sample the origin embeddings from initialized embedding matrices;
        Obtain the graph embeddings in domain A and domain B according to Equation (7);
        Obtain the graph embeddings in shared domain according to Equation (6), and split
          the domain-shared item embeddings;
        Enforce coherence constraints on domain-shared item embedding according to
          Equation (9);
        Disentangle domain-shared user embedding according to Equation (10) and calculate
          the preference disentanglement loss according to Equation (15);
        Generate the final representation according to Equation (16) and (17);
        Calculate the BPR loss according to Equation (20);
        Calculate the overall loss and optimize the parameters of the entire model according
          to Equation (21).
    end
end
// Inference Procedure
for $u$={1,2,3,...,$\mathcal{U}$} do
    for $(u, i^A, i^B)$ in ($\mathcal{D}^A_{test}$ and $\mathcal{D}^B_{test}$) do
        Sample the origin embeddings from initialized embedding matrices;
        Obtain the graph embeddings in domain A and domain B according to Equation (7);
        Obtain the graph embeddings in shared domain according to Equation (6), and split
          the domain-shared item embeddings;
        Disentangle domain-shared user embedding according to Equation (10);
        Generate the final representation according to Equation (16) and (17);
        Calculate the recommendation score according to Equation (18).
    end
end
```

- —*RQ3*: Does disentangling user interest and conformity embeddings enhance recommendation performance compared to solely disentangling user-shared embedding? Moreover, to what degree does the disentanglement of user interest and conformity embeddings contribute to overall performance?
- —*RQ4*: Does our proposed CoPD successfully achieve the desired disentanglement for user preferences?
- —*RQ5*: How do different settings of hyper-parameters influence the performance of CoPD?

Table 2. Statistics of Six Dual-target CDR Tasks

Tasks	#Users	#Items	#Interactions	#Density (%)
Amazon-Elec	3,325	38,706	118,114	0.091
Amazon-Phone	3,325	17,709	52,966	0.089
Amazon-Sport	9,928	30,796	100,938	0.033
Amazon-Cloth	9,928	39,008	95,369	0.024
Amazon-Sport	4,998	20,845	54,256	0.052
Amazon-Phone	4,998	13,655	46,445	0.068
Amazon-Elec	15,761	51,447	224,689	0.027
Amazon-Cloth	15,761	48,781	133,609	0.017
Douban-Movie	2,106	9,555	907,219	4.508
Douban-Book	2,106	6,777	95,974	0.672
Douban-Movie	1,666	9,555	781,288	4.908
Douban-Music	1,666	5,567	69,681	0.751

4.1 Datasets

To evaluate the recommendation performance of our proposed CoPD and ensure a fair comparison with previous methods, we evaluate our model on four publicly available Amazon datasets[1] [13, 34] and three Douban datasets [63]. The four Amazon datasets are Electronics (*Elec*); Cell Phones (*Phone*); Accessories, Sports, and Outdoors (*Sport*); and Clothing, Shoes, and Jewelry (*Cloth*). Following the method introduced in [30], we convert the explicit ratings into implicit feedback, representing ratings as 0 and 1 to indicate whether the user has rated the item. Next, we filter out users with fewer than 5 interactions and items with fewer than 10 interactions. The users in our dual-target CDR tasks are all overlapped, meaning the users at least click one item in each domain; thus, we extract overlapping users in both domains. Following the experimental setting in DisenCDR [2], we exclude cold-start items from the test set. Finally, we divide the aforementioned four Amazon datasets into four dual-target CDR tasks, comprising two tasks with high domain similarity (Elec & Phone, Sport & Cloth) and two tasks with lower domain similarity (Sport & Phone, Elec & Cloth). The three Douban datasets are Douban-Movie, Douban-Book, and Douban-Music. Following the method introduced in GA-DTCDR [63] and DIDA-CDR [66], we convert the explicit ratings into implicit feedback. Next, we filter out users and items with fewer than five interactions. We also filter out of the cold-start item entry in the test set. Finally, we construct the three Douban datasets for two dual-target CDR tasks. The details of these six dual-target CDR tasks are presented in Table 2.

4.2 Evaluation Metrics

In item recommendation, the **Leave-One-Out (LOO)** evaluation [16] is commonly employed to demonstrate performance [2, 15, 18, 30, 47]. To compare the performance of CoPD and other baseline models, we adopt the LOO evaluation method. To construct a test set of each user, one item is preserved as ground truth, and we randomly sample 999 items that the user has not interacted

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

with as negative items. We utilize two widely adopted evaluation metrics: **Hit Ratio (HR)** and **Normalized Discounted Cumulative Gain (NDCG)** [20]. The predicted rank list is cut off at $K = 10$. The HR measures whether the test items are on the top-10 list, and the NDCG estimates ranking quality by assigning higher scores to hits at the top ranks. In both metrics, a higher value indicates superior recommendation performance.

4.3 Baseline Methods

To demonstrate effectiveness, we choose 14 state-of-the-art baseline models for comparison with our proposed CoPD. These 14 baseline models can be divided into four categories, i.e., (1) Single-domain Recommendation (NeuMF, NGCF, LightGCN), (2) Disentanglement-based Single-domain Recommendation (DGCF, DICE), (3) CDR (CDFM, CoNet, DDTCDR, PPGN, BiTGCF), and (4) Disentanglement-based CDR (DisenCDR, DIDA-CDR, ETL, UniCDR).

1. Single-domain Approaches
   - NeuMF [15]: NeuMF develops techniques based on neural networks to tackle the collaborative filtering problem in recommendation.
   - NGCF [45]: NGCF stacks multiple user-item graph propagation layers to aggregate the high-order neighboring information and learn a better recommendation representation.
   - LightGCN [14]: LightGCN is a simple design of NGCF; it removes the non-linear activation and the transformation in NGCF, exhibiting substantial improvements over NGCF.
2. Disentanglement-based Single-domain Approaches
   - DGCF [46]: DGCF disentangles user intents on adopting the items by modeling the intent-aware interaction graphs and encouraging independence of different intents to yield disentangled representations.
   - DICE [60]: DICE presents a general framework that learns structurally disentangled interest and conformity representations. We use LightGCN as the backbone of DICE in our comparison.
3. Cross-domain Approaches
   - CDFM [31]: CDFM enhances recommendations in a target domain by applying factorization machines, utilizing user interaction data from auxiliary domains as contextual information.
   - CoNet [18]: CoNet employs a customized cross-stitch neural network. Specifically, it builds deep cross-connections to enhance the prediction of user-item interactions across diverse domains.
   - DDTCDR [25]: DDTCDR incorporates dual learning into CDR and puts forth a deep dual transfer network. This model needs user features and item features of identical dimensions as input, acquired through matrix factorization.
   - PPGN [56]: PPGN formulates a Cross-domain Preference Matrix to comprehensively represent interactions across diverse domains.
   - BiTGCF [30]: BiTGCF builds upon the Graph Collaborative Filtering network as its foundation. It enables a two-way knowledge transfer across two domains by utilizing common users as bridges.
4. Disentanglement-based Cross-domain Approaches
   - DisenCDR [2]: DisenCDR introduces two mutual-information-based disentanglement regularizers to separate domain-shared and domain-specific information.
   - DIDA-CDR [66]: DIDA-CDR introduces an interpolative data augmentation approach to generate augmented user representations and proposes an effective method to disentangle domain-specific and domain-independent information.

- ETL [3]: ETL models the joint distribution of user behaviors across domains to capture both overlapping and domain-specific features, enhancing CDR accuracy.
- UniCDR [1]: UniCDR employs domain-shared and domain-specific representations enhanced by masking mechanisms and contrastive learning to improve CDR performance.

4.4 Implementation Details

We implement our CoPD method on the PyTorch platform with Python 3.9.0. For all baseline methods and our CoPD, we use the common hyper-parameters settings reported in DisenCDR [2]: The initializing embedding dimension is fixed as 128, the embedding parameters are initialized using the Xavier method [8], the mini-batch size is fixed as 1,024, the learning rate is fixed as 0.001, the dropout rate is fixed as 0.3, and the negative sampling number is fixed as 1. Additionally, we select the best hyper-parameters of the L2 regularization coefficient from {1e-5, 1e-4} and the number of graph encoder layers from 1 to 4. Since the experimental settings of our CoPD are the same as DisenCDR, we directly use best results reported in DisenCDR for single-domain methods (NeuMF, NGCF, and LightGCN), cross-domain methods (CDFM, CoNet, DDTCDR, PPGN, and BiTGCF), and DisenCDR. For disentanglement-based single-domain approaches (DGCF and DICE), we refer to the implementation in official code and RecBole [49, 57, 58] and adjust hyper-parameters according to the optimal configurations outlined in their original papers. For DIDA-CDR, we implement it by ourselves and adjust the parameters μ_1 and μ_2 and other parameters align with the settings in DIDA-CDR. But during training, we randomly sampled one unseen interaction as a negative sample following DisenCDR [2], in contrast to the seven negative samples mentioned in DIDA-CDR. For ETL and UniCDR, we use the source code and fine-tuned the parameters. All baseline methods and our CoPD are trained with 100 epochs for convergence, and all parameters are optimized by Adam optimizer [21].

For the hyper-parameters in our model, we set the number of graph encoder layers to 3 and the L2 regularization coefficient to 1e-4. To determine the optimal hyper-parameters λ_1 and λ_2 in Equation (21), we perform a grid search within the range of $\{0.01, 0.1, 1, 10\} \bigotimes \{0.01, 0.1, 1, 10\}$. The temperature τ is searched within the range of $\{0.01, 0.05, 0.1, 0.15, 0.2\}$. After the search, we set $\lambda_1 = 1$ and $\lambda_2 = 1$ for Elec & Phone, Sport & Cloth, Douban Movie & Book, and Douban Movie & Music, and $\lambda_1 = 0.01$ and $\lambda_2 = 1$ for Elec & Cloth and Sport & Phone. Additionally, we set $\tau = 0.05$ for all datasets.

4.5 Overall Performance Comparison with Baseline Models (RQ1)

In Tables 3 and 4, we provide a performance comparison of CoPD with various approaches, using metrics such as HR@10 and NDCG@10. The results indicate that the performance of CoPD significantly outperforms all competing baselines, demonstrating the effectiveness of our CoPD in solving the data sparsity problem in both domains. Based on the experimental results, we have the following observations: (1) Compared with single-domain methods, CoPD demonstrates superior performance. This indicates that our proposed global-local graph encoder framework, comprising a global graph encoder and two domain-specific local graph encoders, effectively transfers shared information across the two domains. (2) Compared with disentangle-based single-domain methods, although DGCF and DICE demonstrate outstanding performance in certain domains, outperforming several CDR approaches, they often yield inferior results in other domains. Our CoPD outperforms DGCF and DICE and achieves significant improvement in both domains. This indicates that CoPD produces more effective domain-shared disentanglement representations. These representations capture user-shared interest and conformity preferences and make better recommendations. (3) Compared with cross-domain methods, DisenCDR, DIDA-CDR, ETL, UniCDR, and our CoPD outperform all other cross-domain methods, indicating that disentanglement-based

Table 3. The Overall Performance Comparison of CoPD and Baselines on Amazon CDR Tasks

Similar tasks	Amazon-Elec		Amazon-Phone		Amazon-Sport		Amazon-Cloth	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
NeuMF	0.1617	0.0924	0.1584	0.0802	0.1074	0.0546	0.1118	0.0602
NGCF	0.1855	0.1087	0.2279	0.1238	0.1313	0.0687	0.1322	0.0697
LightGCN	0.1917	0.1028	0.2325	0.1272	0.1319	0.0694	0.1358	0.0729
DGCF	0.1969	0.1169	0.2489	0.1488	0.1477	0.0887	0.1114	0.0680
DICE	0.1012	0.0619	0.2200	0.1309	0.1451	0.0862	0.1072	0.0631
CDFM	0.1824	0.1092	0.1797	0.0972	0.1161	0.0633	0.1232	0.0705
CoNet	0.1722	0.0986	0.1766	0.0930	0.1209	0.0641	0.1240	0.0662
DDTCDR	0.1847	0.1108	0.1723	0.0858	0.1186	0.0637	0.1254	0.0713
PPGN	0.2168	0.1163	0.2454	0.1334	0.1510	0.0803	0.1423	0.0768
BiTGCF	0.2214	0.1220	0.2571	0.1393	0.1483	0.0795	0.1468	0.0793
DisenCDR	0.2457	0.1451	0.2876	0.1613	0.1755	0.0946	0.1631	0.0903
DIDA-CDR	0.2121	0.1252	0.2517	0.1394	0.1890	0.1080	0.2001	0.1148
ETL	0.2172	0.1302	0.2712	0.1496	0.1611	0.0939	0.1471	0.0885
UniCDR	0.2292	0.1383	0.2472	0.1377	0.1837	0.1098	0.1785	0.1120
CoPD	**0.2742**	**0.1653**	**0.3275**	**0.1998**	**0.2594**	**0.1649**	**0.2247**	**0.1495**

Dissimilar tasks	Amazon-Sport		Amazon-Phone		Amazon-Elec		Amazon-Cloth	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
NeuMF	0.1011	0.0519	0.1467	0.0780	0.2008	0.1179	0.1084	0.0580
NGCF	0.1606	0.0853	0.1707	0.0922	0.2020	0.1174	0.1086	0.0580
LightGCN	0.1633	0.0916	0.1647	0.0895	0.1997	0.1073	0.1124	0.0611
DGCF	0.1882	0.1147	0.1990	0.1182	0.1645	0.0998	0.0924	0.0555
DICE	0.1531	0.0901	0.1980	0.1131	0.1814	0.1145	0.0980	0.0554
CDFM	0.1197	0.0655	0.1632	0.0901	0.2109	0.1189	0.1037	0.0563
CoNet	0.1288	0.0691	0.1660	0.0915	0.2126	0.1261	0.1135	0.0619
DDTCDR	0.1214	0.0647	0.1617	0.0898	0.2170	0.1310	0.1147	0.0638
PPGN	0.1800	0.1054	0.2040	0.1109	0.2185	0.1236	0.1298	0.0688
BiTGCF	0.1863	0.1011	0.2110	0.1125	0.2161	0.1225	0.1311	0.0680
DisenCDR	0.2017	0.1180	0.2355	0.1297	0.2371	0.1356	**0.1513**	**0.0837**
DIDA-CDR	0.1890	0.1080	0.2021	0.1148	0.2253	0.1336	0.1342	0.0736
ETL	0.1771	0.1006	0.2113	0.1162	0.2339	0.1390	0.1407	0.0777
UniCDR	0.1655	0.1013	0.1933	0.1100	0.2092	0.1215	0.1105	0.0603
CoPD	**0.2182**	**0.1283**	**0.2396**	**0.1342**	**0.2606**	**0.1577**	0.1390	0.0790

Best performance is highlighted in bold and the second-best performance is underlined.

Table 4. The Overall Performance Comparison of CoPD and Baselines on Douban CDR Tasks

Douban tasks	Douban-Movie		Douban-Book		Douban-Movie		Douban-Music	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
NGCF	0.1026	0.0537	0.0731	0.0408	0.0916	0.0423	0.0611	0.0387
LightGCN	0.1053	0.0549	0.0735	0.0415	0.0924	0.0425	0.0636	0.0399
DGCF	0.1567	0.0774	0.1227	0.0682	0.1510	0.0792	0.0655	0.0317
DICE	0.1463	0.0785	0.1227	0.0665	0.1781	0.0939	0.1252	0.0642
CoNet	0.1505	0.0742	0.1023	0.0447	0.1360	0.0686	0.0726	0.0356
PPGN	0.1203	0.0642	0.1052	0.0478	0.1009	0.0435	0.0724	0.0403
BiTGCF	0.1211	0.0646	0.1058	0.0493	0.1014	0.0441	0.0732	0.0410
DisenCDR	0.1639	0.0870	0.1456	0.0820	0.1631	0.0848	0.0927	0.0458
DIDA-CDR	0.1815	0.0949	0.1373	0.0763	0.1883	0.1008	0.0947	0.0500
ETL	0.1572	0.0716	0.1284	0.0748	0.1396	0.0742	0.0830	0.0413
UniCDR	0.2187	0.1191	0.1711	0.1019	0.2022	0.1118	0.1286	0.0664
CoPD	**0.2206**	**0.1194**	**0.1991**	**0.1152**	**0.2220**	**0.1165**	**0.1418**	**0.0785**

Best performance is highlighted in bold and the second-best performance is underlined.

CDR methods can effectively extract users' shared information and mitigate the negative transfer problem [61, 65]. Disentanglement-based methods achieve excellent recommendation results by transferring this shared information across different domains. (4) Compared with disentanglement-based CDR, for DisenCDR and ETL, our proposed model not only employs domain-shared user embedding to derive shared user preference but also introduces the coherence constraints to capture the domain-shared item attributes, resulting in improved recommendations. Particularly in CDR tasks with high domain similarity in Table 3, we achieve significant performance improvement compared with DisenCDR and ETL. For DIDA-CDR, which disentangles user-shared embedding into domain-shared and domain-independent embeddings, our model adopts a different disentanglement approach. CoPD disentangles user-shared embedding into conformity and interest embeddings and outperforms DIDA-CDR significantly. UniCDR employs contrastive learning to maximize mutual information between specific and shared representations, which may disrupt the original distribution of domain-specific features. CoPD applies a stop gradient mechanism when maximizing mutual information between specific and shared representations, which freezes the parameters of domain-specific embeddings. This ensures that domain-specific features remain unaffected and retain their uniqueness. Therefore, CoPD achieves significant performance improvement compared with UniCDR in all CDR tasks.

4.6 Ablation Study (RQ2 and RQ3)

In this subsection, we examine the effectiveness of the two modules: Coherence-guided Item Attribute Alignment (introduced in Section 3.4) and User Preference Disentanglement with Aligned Item Attribute (introduced in Section 3.5). We conduct the experiments on four variants of CoPD and show the results in Figures 4 and 5.

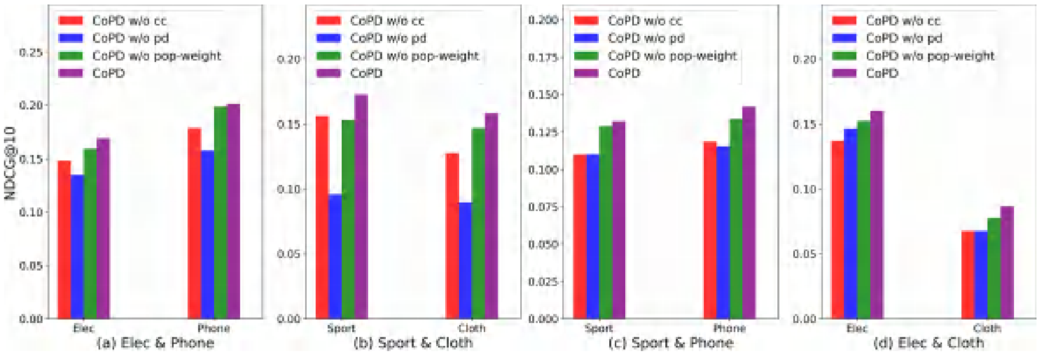


Fig. 4. Performance comparison between CoPD and its two variants CoPD w/o pd and CoPD w/o cc.

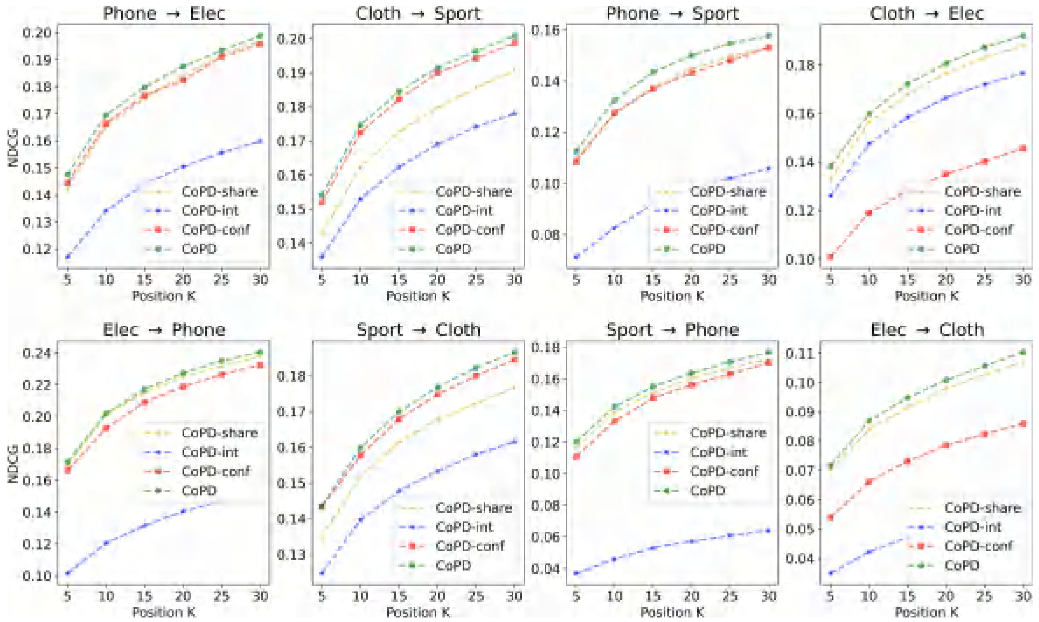


Fig. 5. Effectiveness of interest embedding CoPD-int and conformity embedding CoPD-conf.

From Figure 4, we investigate the effectiveness of the two modules: Coherence-guided Item Attribute Alignment and User Preference Disentanglement guided by Aligned Item Attribute.

- CoPD w/o cc: The CoPD model without Coherence-guided Item Attribute Alignment module mentioned in Section 3.4. Specifically, we don't generate the domain-shared embeddings to facilitate the learning of disentangled user preference embeddings.
- CoPD w/o pd: The CoPD model without preference disentanglement module. Specifically, we don't disentangle the domain-shared user embedding into conformity and interest embeddings. Instead, we directly fuse the domain-shared user embedding with domain-specific user embedding for recommendation.
- CoPD w/o pop-weight: We train preference embeddings with non-popularity weighted Softmax loss.

(1) The performance of the CoPD w/o pd variant significantly decreases. This observation emphasizes the significance of the User Preference Disentanglement with the Aligned Item Attribute module in our model. Our CoPD significantly enhances by generating diverse and robust disentangled user preference embeddings from the domain-shared user embeddings for recommendation by incorporating User Preference Disentanglement. Additionally, CoPD outperforms CoPD w/o pop-weight, highlighting the importance of preference disentanglement with popularity-weighted Softmax loss.

(2) The CoPD w/o cc variant exhibits a smaller performance decrease than the CoPD w/o pd variant on Elec & Phone and Sport & Cloth (two high domain similarity CDR tasks). This suggests that aligning item attributes is less significant than disentangling user preferences. This observation indicates that it is possible for conventional models to capture domain-shared item attributes when users interact with items with two similar domains. In CDR tasks with lower domain similarity, such as Sport & Phone and Elec & Cloth, most item attributes in these domains are distinct. It becomes challenging for recommendation models to extract domain-shared item attributes. Therefore, by employing item alignment with coherence constraints in these recommendation tasks, CoPD filters out domain-specific information and extracts more domain-shared item attributes for domain-shared item embeddings, leading to superior improvements.

Next, we investigate the influence of user interest and conformity embeddings on CoPD performance and the results are depicted in Figure 5.

- CoPD-share: A baseline model that lacks both interest and conformity mechanism and uses domain-shared embeddings for recommendation. We use non-weighted Softmax loss for training.
- CoPD-int: We don't use conformity embedding. Instead, we directly fuse the interest embedding with domain-specific user embedding for recommendation.
- CoPD-conf: We don't use interest embedding. Instead, we directly fuse the conformity embedding with domain-specific user embedding for recommendation.

(1) For these four recommendation tasks, users tend to click and give high ratings to items based on their conformity embeddings rather than interest embeddings. This indicates that conformity plays a significant role in CDR. Additionally, users tend to prefer popular items because a substantial portion of popular items is associated with high quality [59]. Moreover, many users do not exhibit specific interest preferences in certain domains. For instance, when browsing items in the Sports domain (Accessories, Sports, and Outdoors), users prefer items with good quality and reputation but don't care about particular attributes.

(2) The user interest embedding significantly impacts the Elec domain of the Elec & Cloth task. This illustrates that user interest preferences are also crucial, particularly when users click on items with their real interests. Additionally, we find that the Coherence-guided Item Attribute Alignment module is more effective in the Elec domain of Elec & Cloth task, as indicated in Figure 4. It indicates that extracting common item attributes aids CoPD in learning the interest embeddings of users, especially when users interact with items with their interests.

(3) CoPD-share outperforms CoPD-conf and CoPD-int in most CDR tasks, indicating that incorporating both conformity and interest information enhances the recommendation performance. However, CoPD surpasses CoPD-share in all tasks, underscoring the effectiveness of preference disentanglement for interest and conformity.

4.7 Analysis of Preference Disentanglement (RQ4)

This section confirms that our CoPD can learn domain-shared user interest and conformity embeddings. We use $Conf-emb$ (blue) and $Int-emb$ (orange) to represent the learned shared conformity

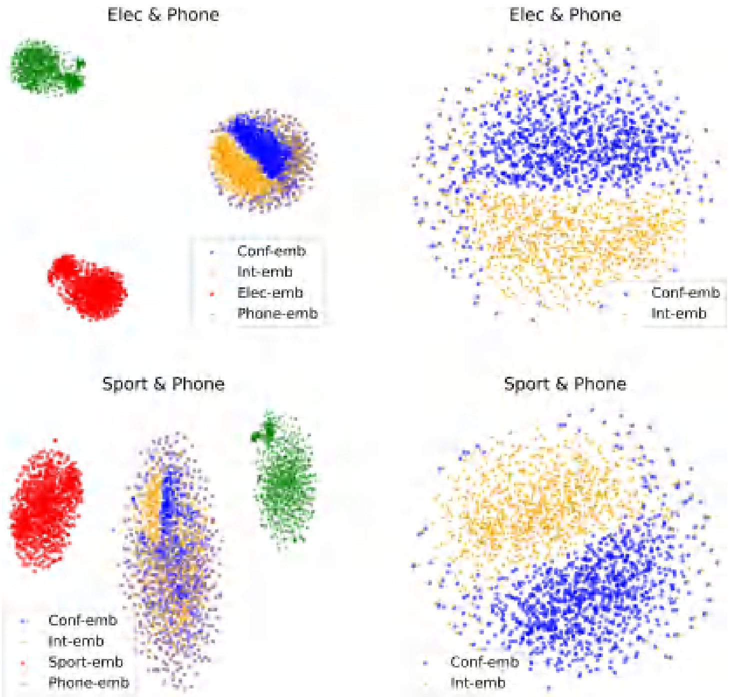


Fig. 6. Visualization of conformity embeddings (blue), interest embeddings (orange), domain-A-specific embeddings (red), and domain-B-specific embeddings (green).

and interest embeddings, respectively. We use $A - emb$ (red) and $B - emb$ (green) to represent the domain-specific embeddings in domains A and B, respectively.

We visualize the learned embeddings using t-SNE [42], as shown in Figure 6. In the left columns, the two-dimensional embeddings of domain-shared and domain-specific are distinctly separated, indicating that our CoPD effectively disentangles user preference embeddings into domain-shared and domain-specific embeddings. However, interest and conformity embeddings are not clearly distinguishable due to the confounding influence of domain-specific embeddings. In the right columns, we exclusively train domain-shared interest and conformity embeddings in t-SNE. The conformity embeddings and interest embeddings are distinguishable, demonstrating the effectiveness of the disentanglement by popularity-weighted Softmax loss. In conclusion, unlike previous approaches that only rely on domain-shared user embeddings, our disentanglement method disentangles the domain-shared user embeddings into interest and conformity embeddings, resulting in improved performance.

4.8 Case Study (RQ4)

In this section, we present a case study to show the effectiveness of our model in capturing user interest and conformity preferences, as illustrated in Figure 7. We select the top-4 items in each domain with different preference recommendations in the Sport & Cloth task. In observing the recommendation made by the user conformity embedding (red dotted circle), a clear trend emerges: Items with high ratings also exhibit high popularity. This implies that the user #22, with his conformity preference, tends to click on popular items. For example, in the Sport domain, item #384 has a popularity of 0.4589; in the Cloth domain, item #286 has a popularity of 0.4034.

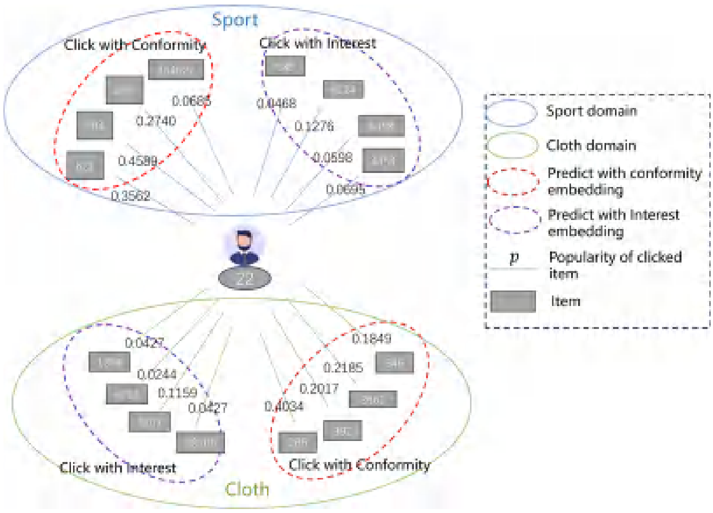
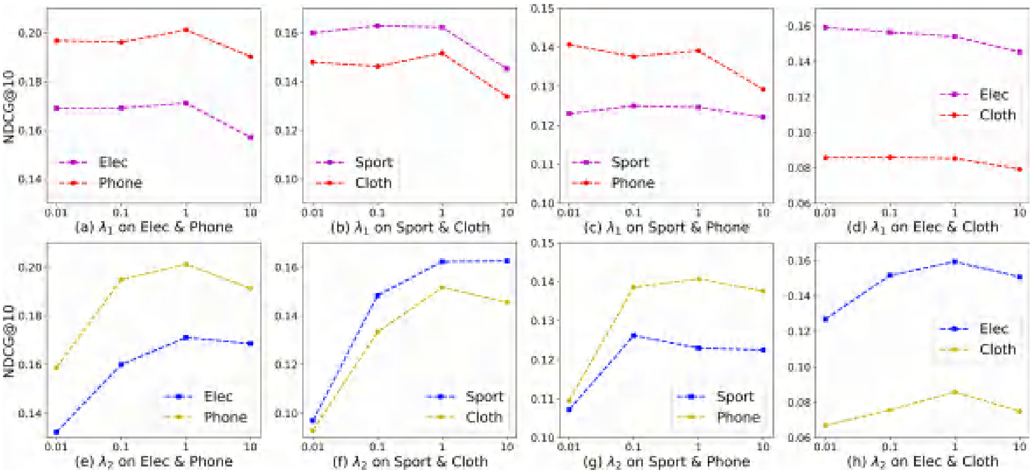


Fig. 7. The case study of user #22 on Sport & Cloth task.



(a) λ_1 on Elec & Phone (b) λ_1 on Sport & Cloth (c) λ_1 on Sport & Phone (d) λ_1 on Elec & Cloth

(e) λ_2 on Elec & Phone (f) λ_2 on Sport & Cloth (g) λ_2 on Sport & Phone (h) λ_2 on Elec & Cloth

Fig. 8. (a)–(d): Impact of λ_1. (e)–(h): Impact of λ_2.

Furthermore, in observing the recommendation made by the user interest embedding (purple dotted circle), the popularity of clicked items remains consistently low. This indicates that these items are popularity-irrelevant and reflect the user's real interest. For instance, in the Sport domain, item #538 has a popularity of 0.0468, and in the Cloth domain, item #4734 has a popularity of 0.0244. When disentangling user preferences using popularity-weighted loss, it's worth noting that our CoPD successfully achieves the desired separation between user conformity and interest preferences.

4.9 The Impact of Hyper-parameters (RQ5)

In this subsection, (1) we initially test the sensitivity of two hyper-parameters λ_1 and λ_2 in Equation (21). According to Figure 8, we conclude that λ_2 is more sensitive than λ_1. λ_1 represents a tradeoff of coherence constraints to synchronize domain-shared embeddings with domain-specific ones.

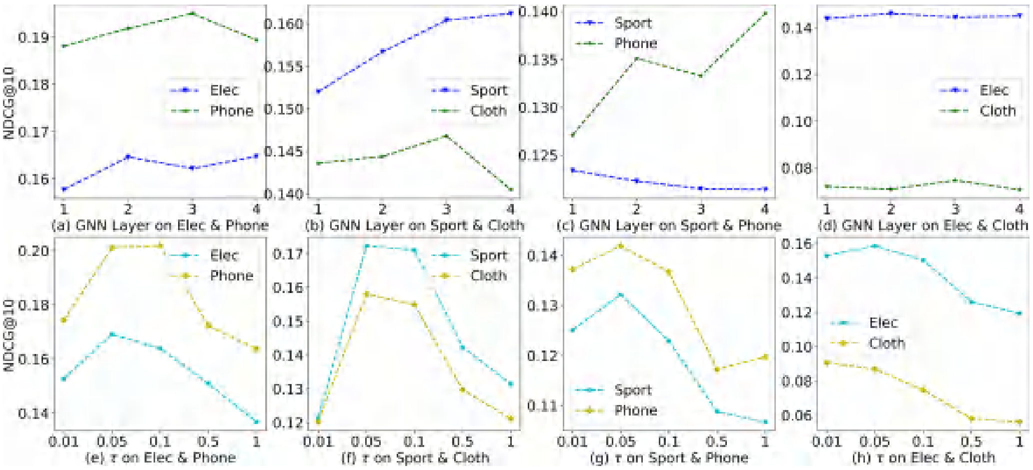


Fig. 9. (a)–(d): Impact of the number of GNN layers. (e)–(h): Impact of τ.

A higher λ_1 reduces redundant information in domain-shared embeddings but risks losing valuable domain-shared information. Conversely, a lower λ_1 preserves domain-shared information more effectively, although it may not eliminate redundancy as efficiently. From Figure 8(a)–(d), we observe that a slightly higher λ_1 (e.g., 1) improves the performance in tasks with high domain similarity by reducing redundant information (Figure 8(a) and (b)), while a lower λ_1 (e.g., 0.01) preserves the domain-shared information and performs better in tasks with low domain similarity (Figure 8(c) and (d)). However, when λ_1 is set too high (e.g., 10), performance declines sharply due to the excessive loss of shared information and the increased influence of domain-specific information. For λ_2, a slightly larger value, such as 1, yields the best performance. Therefore, with a larger value of λ_2, CoPD can pay more attention to disentangling user interest and conformity embeddings. However, a larger value of λ_2 reduces performance, as it decreases the influence of domain-specific information. (2) Next, we conducted experiments to investigate how the number of GNN layers influences performance and report the results in Figure 9(a)–(d). The 1-layer and 4-layer variants achieved the worst results because the 1-layer GNN can't aggregate enough information, and the 4-layer variant suffers from the over-smoothing problem. Specifically, the best number of GNN layers on Sport is 1 in Figure 9(c). This is due to the fact that the number of GNN layers influences both the global graph G^S and single-domain graph G^A (the Sport graph). As the GNN layers increase, the Sport domain receives more redundant information from Phone domain via the global graph G^S. (3) Finally, as illustrated in Figure 9(e)–(h), the temperature τ is crucial for model performance. We observe that a large value of τ leads to poor performance, aligning with experimental results reported in [28].

5 Conclusion

This article introduces a CoPD method to tackle challenges arising from data sparsity in recommendation tasks. The CoPD framework comprises three components. The first component is Coherence-Guided Item Attribute Alignment, where a global-local graph encoder framework and coherence constraints are introduced to acquire common item attributes. The global-local graph encoder framework facilitates the effective learning of domain-shared and domain-specific embeddings, and coherence constraints preserve common and essential attributes in domain-shared embeddings while eliminating redundant attributes from another domain.

The second component is User Preference Disentanglement with aligned item attributes. For learning disentangled user preferences, we utilize the popularity-weighted Softmax loss between aligned item embeddings and domain-shared user embeddings, guiding the model to emphasize the user's conformity preferences towards common attributes of popular items and directing the user's interest preferences towards the common attributes of popular-irrelevant items. The third component is the fusion of domain-specific and domain-shared embeddings, aimed at integrating these embeddings into comprehensive user preferences and item attributes. Experimental results demonstrate the superior performance of CoPD over state-of-the-art baselines.

In the future, we plan to enhance our disentanglement techniques by integrating side information, such as user reviews and item category data, to more effectively capture users' shared preferences across domains and enhance CDRs' performance. Furthermore, we plan to devise specific coherence constraints for different CDR tasks, such as cold-start user and cold-start item recommendations. Finally, we will investigate the preference disentanglement in group-based recommendations. Motivated by the insights of this article, we hold that users within the same group share common preferences, referred to as conformity and interest studied in this article. By disentangling users' group preferences into conformity and interest, we anticipate enhancing the performance of group-based recommendations.

References

[1] Jiangxia Cao, Shaoshuai Li, Bowen Yu, Xiaobo Guo, Tingwen Liu, and Bin Wang. 2023. Towards universal cross-domain recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*, 78–86.

[2] Jiangxia Cao, Xixun Lin, Xin Cong, Jing Ya, Tingwen Liu, and Bin Wang. 2022. Disencdr: Learning disentangled representations for cross-domain recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM*, 267–277.

[3] Xu Chen, Ya Zhang, Ivor W. Tsang, Yuangang Pan, and Jingchao Su. 2023. Toward equivalent transformation of user preferences in cross domain recommendation. *ACM Transactions on Information Systems* 41, 1 (2023), 1–31.

[4] Zhihong Chen, Jiawei Wu, Chenliang Li, Jingxu Chen, Rong Xiao, and Binqiang Zhao. 2022. Co-training disentangled domain adaptation network for leveraging popularity bias in recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 60–69.

[5] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A heterogeneous graph framework for multi-target cross-domain recommendation. In *Proceedings of the 3rd Workshop on Online Recommender Systems and User Modeling (ORSUM '20), in Conjunction with the 14th ACM Conference on Recommender Systems (ORSUM@ RecSys)*.

[6] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Proceedings of the Spanish Conference on Information Retrieval*, Vol. 24.

[7] Yu Fu, Yanwei Fu, Jingjing Chen, and Yu-Gang Jiang. 2021. Generalized meta-FDMixup: Cross-domain few-shot learning guided by labeled target data. *IEEE Transactions on Image Processing* 31 (2021), 7078–7090.

[8] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. JMLR.org, 249–256.

[9] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. 2018. Image-to-image translation for cross-domain disentanglement. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 31.

[10] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI '21)*. Zhi-Hua Zhou (Ed.), ijcai.org, 2483–2489.

[11] Lei Guo, Jinyu Zhang, Tong Chen, Xinhua Wang, and Hongzhi Yin. 2022. Reinforcement learning-enhanced shared-account cross-domain sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering 35*, 7 (2022), 7397–7411.

[12] Lei Guo, Jinyu Zhang, Li Tang, Tong Chen, Lei Zhu, and Hongzhi Yin. 2024. Time interval-enhanced graph neural network for shared-account cross-domain sequential recommendation. *IEEE Transactions on Neural Networks and Learning Systems* 35, 3 (2024), 4002–4016.

[13] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, 507–517.

[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 639–648.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182.

[16] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 549–558.

[17] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net.

[18] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 667–676.

[19] Lin Hu, Junxiang Peng, Yan Zhang, Hong Wen, Shuai Tan, and Jiabing Fan. 2022. Artificial noise assisted interference alignment for physical layer security enhancement. In *Proceedings of the 2022 IEEE Global Communications Conference (GLOBECOM '22)*. IEEE, 4148–4153.

[20] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[21] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. ICLR.

[22] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. 2020. Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1081–1090.

[23] Ansong Li, Zhiyong Cheng, Fan Liu, Zan Gao, Weili Guan, and Yuxin Peng. 2022. Disentangled graph neural networks for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 7870–7882.

[24] Chenglin Li, Mingjun Zhao, Huanming Zhang, Chenyun Yu, Lei Cheng, Guoqiang Shu, Beibei Kong, and Di Niu. 2022. RecGURU: Adversarial learning of generalized user representations for cross-domain recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. ACM, 571–581.

[25] P. Li and Alexander Tuzhilin. 2019. DDTCDR: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 331–339.

[26] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 305–314.

[27] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized ranking with importance sampling. In *Proceedings of the Web Conference 2020*. ACM/IW3C2, 1093–1103.

[28] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*. ACM, 2320–2329.

[29] Hao Liu, Lei Guo, Lei Zhu, Yongqiang Jiang, Min Gao, and Hongzhi Yin. 2024. MCRPL: A pretrain, prompt, and fine-tune paradigm for non-overlapping many-to-one cross-domain recommendation. *ACM Transactions on Information Systems* 42, 4 (2024), 1–24.

[30] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. 2020. Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. ACM, 885–894.

[31] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *Proceedings of the European Conference on Information Retrieval*. Springer, 656–661.

[32] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. 2023. Exploring false hard negative sample in cross-domain recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.), ACM, 502–514.

[33] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach. In *Proceedings of the International Joint Conference on Artificial Intelligence. ijcai.org*, 2464–2470.

[34] Julian McAuley, Christopher Targett, Javen Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 43–52.

[35] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling preference representations for recommendation critiquing with ß-VAE. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1356–1365.

[36] Wentao Ning, Xiao Yan, Weiwen Liu, Reynold Cheng, Rui Zhang, and Bo Tang. 2023. Multi-domain recommendation with embedding disentangling and domain alignment. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. ACM, 1917–1927.

[37] Tieyun Qian, Yile Liang, Qing Li, and Xuan Ma, Ke Sun, and Zhiyong Peng. 2022. Intent disentanglement and feature self-supervision for novel recommendation. *IEEE Transactions on Knowledge and Data Engineering 35, 10* (2022), 9864–9877.

[38] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, 995–1000.

[39] Steffen Rendle. 2021. Item recommendation from implicit feedback. In *Recommender Systems Handbook*. Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.), Springer, 143–171.

[40] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618. Retrieved from https://arxiv.org/abs/1205.2618.

[41] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1285–1293.

[42] Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.

[43] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* Computer Vision Foundation/IEEE, 2495–2504.

[44] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. 2022. Disentangled representation learning for recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2022), 408–424.

[45] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165–174.

[46] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1001–1010.

[47] Bin Wu, Tianren Shi, Lihong Zhong, Yan Zhang, and Yangdong Ye. 2023. Graph-coupled time interval network for sequential recommendation. *Information Sciences* 648 (2023), 119510.

[48] Hui Xu, Changyu Li, Yan Zhang, Lixin Duan, Ivor W. Tsang, and Jie Shao. 2022. Metacar: Cross-domain meta-augmentation for content-aware recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 8199–8212.

[49] Lanling Xu, Zhen Tian, Gaowei Zhang, Junjie Zhang, Lei Wang, Bowen Zheng, Yifan Li, Jiakai Tang, Zeyu Zhang, Yupeng Hou, et al. 2023. Towards a more user-friendly and easy-to-use benchmark library for recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2837–2847.

[50] Zitao Xu, Weike Pan, and Zhong Ming. 2023. A multi-view graph contrastive learning framework for cross-domain sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM, 491–501.

[51] Xiaoxin Ye, Yun Li, and Lina Yao. 2023. DREAM: Decoupled representation via extraction attention module and supervised contrastive learning for cross-domain sequential recommender. In *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM, 479–490.

[52] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 269–277.

[53] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. 2022. A survey on cross-domain recommendation: Taxonomies, methods, and future directions. *ACM Transactions on Information Systems* 41, 2 (2022), 1–39.

[54] An Zhang, Wenchang Ma, Xiang Wang, and Tat-Seng Chua. 2022. Incorporating bias-aware margins into contrastive loss for collaborative filtering. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 35, 7866–7878.

[55] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. 2023. Invariant collaborative filtering to popularity distribution shift. In *Proceedings of the ACM Web Conference 2023*, 1240–1251.

[56] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-domain recommendation via preference propagation GraphNet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2165–2168.

[57] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. RecBole 2.0: Towards a more up-to-date recommendation library. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*. ACM, 4722–4726.

[58] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. RecBole: Towards a unified, comprehensive and efficient framework for recommendation

algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*. ACM, 4653–4664.

[59] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2022. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Transactions on Knowledge and Data Engineering 35, no. 10* (2022), 9920–9931.

[60] Y. Zheng, Chen Gao, Xiang Li, Xiangnan He, Depeng Jin, and Yong Li. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of the Web Conference 2021*, 2980–2991.

[61] Jie Zhou, Qian Yu, Chuanxian Luo, and Jing Zhang. 2023. Feature decomposition for reducing negative transfer: A novel multi-task learning method for recommender system. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 16390–16391.

[62] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. DTCDR: A framework for dual-target cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 1533–1542.

[63] Feng Zhu, Yan Wang, and Chaochao Chen, Guanfeng Liu, and Xiaolin Zheng. 2020. A graphical and attentional framework for dual-target cross-domain recommendation. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence (IJCAI)*, Vol. 21. ijcai.org, 39.

[64] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: Challenges, progress, and prospects. arXiv:2103.01696. Retrieved from https://arxiv.org/abs/2103.01696

[65] Feng Zhu, Yan Wang, Jun Zhou, Chaochao Chen, Longfei Li, and Guanfeng Liu. 2021. A unified framework for cross-domain and cross-system recommendations. *IEEE Transactions on Knowledge and Data Engineering* 35 (2021), 1171–1184.

[66] Jiajie Zhu, Yan Wang, Feng Zhu, and Zhu Sun. 2023. Domain disentanglement with interpolative data augmentation for dual-target cross-domain recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 515–527.

[67] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2021. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, 1507–1515.

Received 23 April 2024; revised 20 October 2024; accepted 20 February 2025