

Integrating Group-based Preferences from Coarse to Fine for Cold-start Users Recommendation

Siyu Wang^{1†}, Jianhui Jiang^{1†}, Jiangtao Qiu², Shengran Dai^{1*}

¹Gusu Laboratory of Materials, Suzhou, China

²Southwestern University of Finance and Economics, Chengdu, China

{wangsiyu2022, jiangjianhui2021, daishengran2021}@gusulab.ac.cn

qiuajt_t@swufe.edu.cn

Abstract

Recent studies have demonstrated that cross-domain recommendation (CDR) effectively addresses the cold-start problem. Most approaches rely on transfer functions to generate user representations from the source to the target domain. Although these methods substantially enhance recommendation performance, they exhibit certain limitations, notably the frequent oversight of similarities in user preferences, which can offer critical insights for training transfer functions. Moreover, existing methods typically derive user preferences from historical purchase records or reviews, without considering that preferences operate at three distinct levels: category, brand, and aspect, each influencing decision-making differently. This paper proposes a model that integrates the preferences from coarse to fine levels to improve recommendations for cold-start users. The model leverages historical data from the source domain and external memory networks to generate user representations across different preference levels. A meta-network then transfers these representations to the target domain, where user-item ratings are predicted by aggregating the diverse representations. Experimental results demonstrate that our model outperforms state-of-the-art approaches in addressing the cold-start problem on three CDR tasks.

1 Introduction

Recommender systems are crucial for e-commerce platforms and have garnered significant attention from both industry and academia. Despite extensive research, recommending content to users without historical records remains a major challenge, particularly for cold-start users. Recent studies (Zhu et al., 2022; Man et al., 2017; Zhu et al., 2021a; Zhao et al., 2020; Wang et al., 2020; Vartak

et al., 2017; Sun et al., 2023; Li et al., 2024) demonstrate the effectiveness of CDR systems in addressing this challenge. These works primarily focus on transferring user representations from a source domain with abundant historical data to a target domain with minimal or no historical data. Building on these researches, this paper also employs CDR methods to tackle the cold-start problem.

The key of CDR for cold-start users is to bridge user representations in the source domain and the target domain. Previous studies mainly utilize neural networks to extract user’s item-level preference representation from historical purchased items in the source domain (Man et al., 2017; Wang et al., 2018; Zhu et al., 2022; Sun et al., 2023; Li et al., 2024). Additionally, a common transfer function, as shown in Figure 1 (a) was designed for all users to facilitate the migration of representation from the source domain to the target domain. During training, overlapping users are employed to optimize the models, resulting in the successful implementation of a cross-domain recommendation system for cold-start users. However, users’ preferences vary, necessitating personalized transfer approaches for each individual. Then, Zhu et al. (2022) utilized a meta-network to generate personalized transfer functions for each user, as shown in Figure 1 (b). On the other hand, a unified user representation cannot reflect the user’s multiple preferences in the source domain. Thus, Sun et al. (2023) proposed a novel reinforced multiple preferences transfer framework for CDR.

Although these methods substantially enhance recommendation performance, they exhibit certain limitations, notably the frequent oversight of similarities in user preferences, which can offer critical insights for training transfer functions, as shown in Figure 1 (c). Moreover, existing methods typically derive user preferences from historical purchase records or reviews, without considering that preferences operate at three distinct levels: cate-

[†]Equal Contribution

*Corresponding Author

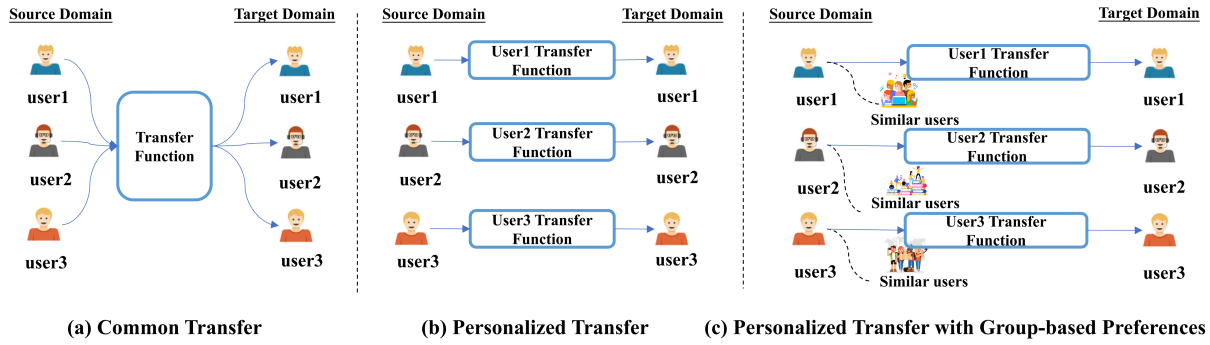


Figure 1: An illustration of common transfer, personalized transfer, and personalized transfer with similar users preferences (group-based preferences).

gory, brand, and aspect, each influencing decision-making differently. For example, some users prioritize brands, while others focus on specific aspects of items. Therefore, effectively capturing and aggregating representations across different preference levels is crucial.

This paper integrates **GR**oup-bAseD **P**reference**E**s from coarse to fine for cold-start users recommendation (**GRAPECDR**). First, we use historical data to generate user representations at category, brand, and aspect levels in the source domain. External memory networks create group-based representations for each level, which are then transferred to the target domain by a meta-network. In the target domain, we predict user-item ratings by combining: (1) category preference from the user’s representation and item’s category embedding, (2) brand preference from the user’s representation and item’s brand embedding, and (3) aspect preference from review aspects and polarities matched with the user’s representation. The final rating is an aggregate of these three ratings. Experiments on three cross-domain tasks show our model outperforms many state-of-the-art approaches.

2 Related work

2.1 Cold-start Recommendation

Previous works have focused on the following aspects of research: (1) How to efficiently use limited interactive information. In certain cold-start scenarios, where only a limited amount of interaction data is available, efficiently leveraging this constrained data becomes crucial. Current research (Lu et al., 2020; Vartak et al., 2017; Lee et al., 2019) primarily employed meta-learning approaches to address such situations, yielding promising results. (2) How to use policy for recommendation. For ex-

ample, Pan et al. (2019a); Li et al. (2010) adopted the method of reinforcement learning to adjust their product selection strategies according to user click feedback to maximize the total number of user clicks. (3) How to efficiently use auxiliary information. The user’s auxiliary information includes rich content, which can greatly improve the performance of cold-start recommendations. Generally speaking, this auxiliary information includes user attributes (Man et al., 2017; Pan et al., 2019b; Seroussi et al., 2011; Volkovs et al., 2019; Zhu et al., 2021b), knowledge graph (Wang et al., 2019), and cross-domain information (Man et al., 2017; Zhu et al., 2021a; Wang et al., 2018; Zhu et al., 2022; Zhao et al., 2020; Sun et al., 2023; Li et al., 2024). Overall, the cross-domain recommendation (CDR) used the data of the auxiliary domain to help the recommendation in the target domain achieved better results than other methods (Zhu et al., 2022). Therefore, in this paper, we focus on CDR for cold-start users.

2.2 Cross-domain Recommendation for Cold-start Users

The core task of CDR for cold-start users is to bridge user preference representations between the source and target domains. Current studies derive these representations mainly from users’ interaction history (item-level) or reviews (aspect-level) in the source domains.

Methods based on user interaction history primarily used matrix factorization (MF) models to generate latent factors or rating patterns, then transferred them across domains. Pan et al. (2010) integrated user and item knowledge from auxiliary data sources using a matrix-based transfer learning framework. EMCDR (Man et al., 2017) developed an embedding-and-mapping framework for

cross-domain recommendation. [Zhu et al. \(2018\)](#) combined MF models and a Deep Neural Network (DNN), leveraging user and item sparsity in both domains to guide DNN training. [Wang et al. \(2018\)](#) introduced Cross-Domain Latent Feature Mapping (CDLFM) with user rating behavior and neighborhood-based feature mapping. Previous models overlooked bidirectional latent connections between users and items, which [Li and Tuzhilin \(2020\)](#) addressed with a dual learning approach for iterative information transfer. [Zhu et al. \(2022\)](#) proposed PTUPCDR, a framework for personalized transfer of user preferences, but it neglected preference correlations among users. To address this, [Li et al. \(2024\)](#) proposed a novel CDR framework with neural processes. Latest, [Zhu et al. \(2024\)](#) proposed a multi-interest meta-network to decouple users’ multiple interests, and generate multi-interest bridges to transfer user embeddings from the source domain to the target domain.

Considering user reviews contain rich preference information, [Zhao et al. \(2020\)](#) extracted aspect information from reviews to generate user aspect-level representation and transferred it from the source to the target domain. [Fu et al. \(2019\)](#) used review text and item content as supplementary data, integrating multiple variations of Stacked Denoising Autoencoders (SDAE) to combine this information with the rating matrix in both domains. However, a unified user representation cannot capture a user’s multiple interests. To address this, [Sun et al. \(2023\)](#) proposed a reinforced multi-interest transfer framework for CDR.

3 The GRAPECDR Framework

3.1 Overview of GRAPECDR

We use $\mathcal{D}^s = \{\mathcal{I}^s, \mathcal{U}^s\}$ and $\mathcal{D}^t = \{\mathcal{I}^t, \mathcal{U}^t\}$ to denote the source and target domains, where $\mathcal{I}^{s/t}$ and $\mathcal{U}^{s/t}$ are item and user set in $\mathcal{D}^{s/t}$ respectively. Let \mathcal{U}^o be an overlapping user set in two domains, that is, the user has purchased items in both domains. \mathcal{U}^c is a cold-start user that only has purchased items in \mathcal{D}^s domain. Given a user $u \in \mathcal{U}^c$ and an item $i \in \mathcal{I}^t$, predict the user-item rating $\hat{r}_{u,i}$.

The architecture of GRAPECDR is shown in Figure 2. It mainly consists of three parts: **User Group-based Representations Generation**, **Representations Transfer**, and **Rating Prediction**. Similar to previous studies ([Zhao et al., 2020](#); [Zhu et al., 2022](#); [Sun et al., 2023](#); [Li et al., 2024](#)), we only select overlapping users \mathcal{U}^o for our experi-

ments.

3.2 User Group-based Representations Generation

3.2.1 Intrinsic Representations Generation

As mentioned earlier, the user’s preferences include three levels from coarse to fine: category-level, brand-level, and aspect-level. In this paper, we first generate three levels of user intrinsic representations in \mathcal{D}^s .

Each item can belong to one or more categories. For example, the book (The Progressive Era) belongs to ‘*Historical Novel*’ and ‘*United States History*’ categories. In \mathcal{D}^s , the u ’s category-level intrinsic representation can be computed by:

$$\mathbf{e}_u^s = \text{softmax}(\mathbf{n}_{e_u^s})\mathbf{E}^s, \quad (1)$$

where $\mathbf{n}_{e_u^s} \in \mathbb{R}^{1 \times |\mathcal{E}^s|}$ is number of occurrences for each category in u ’s historical purchased items (If the category does not appear, the number will be 0), \mathcal{E}^s is category vocabulary in \mathcal{D}^s , and $\mathbf{E}^s = [\mathbf{e}_1^s, \mathbf{e}_2^s, \dots, \mathbf{e}_{|\mathcal{E}^s|}^s]$ are embedding matrix of \mathcal{E}^s .

Similarly, the user’s brand-level intrinsic representation can be obtained by:

$$\mathbf{i}_u^s = \text{softmax}(\mathbf{n}_{i_u^s})\mathbf{I}^s, \quad (2)$$

where $\mathbf{I}^s = [\mathbf{i}_1^s, \mathbf{i}_2^s, \dots, \mathbf{i}_{|\mathcal{I}^s|}^s]$ is the brand embedding matrix, and $\mathbf{n}_{i_u^s} \in \mathbb{R}^{1 \times |\mathcal{I}^s|}$ is number of occurrences for each brand in u ’s historical purchased items and \mathcal{I}^s is brand vocabulary in \mathcal{D}^s .

Further, we extract aspect occurrence list and the number of occurrences from u ’s reviews, as elaborated in Appendix A. $\mathbf{n}_{v_u^s} \in \mathbb{R}^{1 \times |\mathcal{V}^s|}$ is u ’s number of occurrences for each aspect, \mathcal{V}^s is aspect vocabulary in \mathcal{D}^s . Finally, the u ’s aspect-level intrinsic representation can be obtained by:

$$\mathbf{v}_u^s = \text{softmax}(\mathbf{n}_{v_u^s})\mathbf{V}^s, \quad (3)$$

where $\mathbf{V}^s = [\mathbf{v}_1^s, \mathbf{v}_2^s, \dots, \mathbf{v}_{|\mathcal{V}^s|}^s]$ is the embedding matrix of \mathcal{V}^s .

3.2.2 Group-based Representations Generation

At this stage, the three levels of u ’s intrinsic representations in \mathcal{D}^s have been obtained. The category-level intrinsic representation \mathbf{e}_u^s will be used as an example to illustrate the process of generating the group-based representation in the source domain.

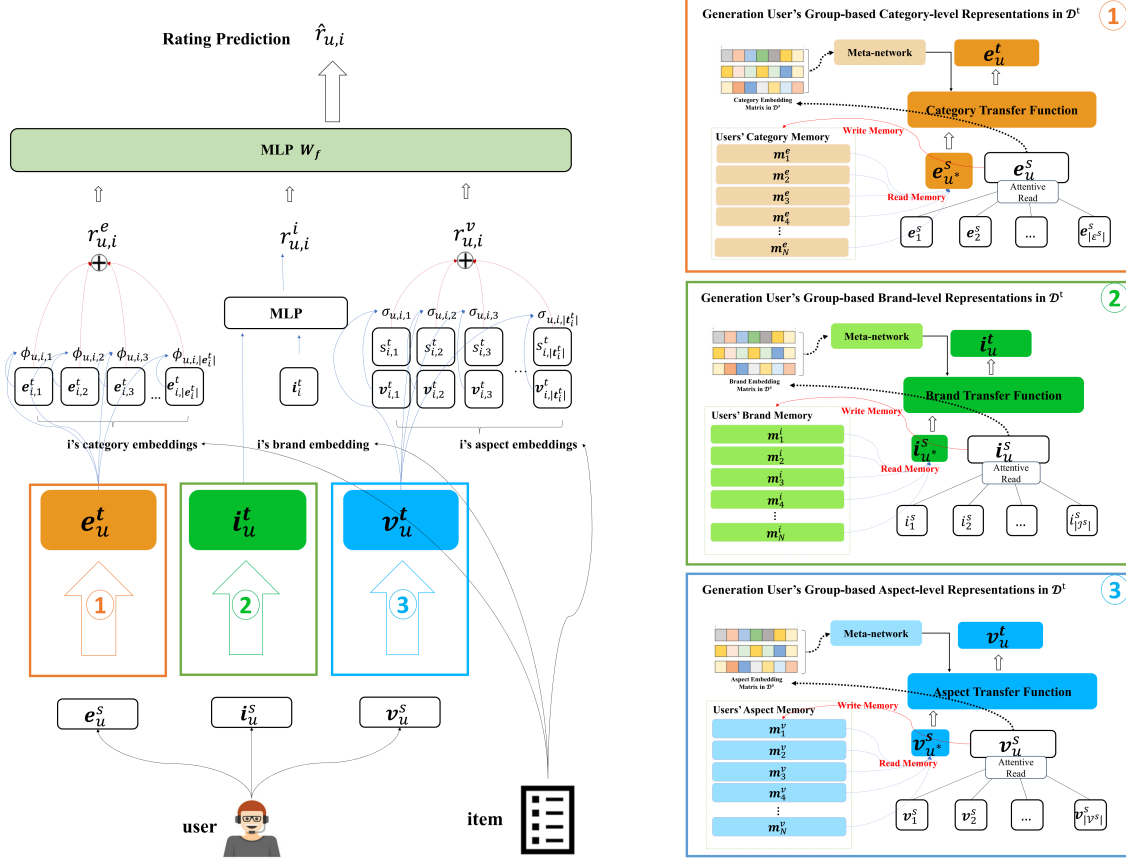


Figure 2: The architecture of GRAPECDR.

Firstly, we introduce an **External Memory Network (EMN)** (Graves et al., 2014) to store the typical features of overlapping users at the category-level. Suppose there are N overlapping users in our system, their category memory matrix is $\mathbf{M}^e = [\mathbf{m}_1^e, \mathbf{m}_2^e, \dots, \mathbf{m}_N^e]$, where \mathbf{m}_k^e is the memory slot of k^{th} user. Then, a soft-attention mechanism is adopted to read \mathbf{M}^e for obtaining u 's group-based representation at category-level, denoted as $\mathbf{e}_{u^*}^s$:

$$\mathbf{e}_{u^*}^s = \sum_{k=1}^N z_{u,k} \cdot \mathbf{m}_k^e, \quad (4)$$

$$z_{u,k} = \frac{\exp w_{u,k}}{\sum_{i=1}^N \exp w_{u,i}}, w_{u,k} = \mathbf{e}_u^s \cdot (\mathbf{m}_k^e)^T, \quad (5)$$

where \mathbf{e}_u^s is computed by Equation 1, and $z_{u,k}$ is similarity weight. The memory slot in \mathbf{M}^e corresponding to each user is expected to store as many of the user's typical features as possible. The group-based representation for the user is then derived through a weighted summation of the typical features of similar users. Consequently, the memory

slot for each user must be updated during training (Chen et al., 2018). Drawing inspiration from Neural Turing Machine (Graves et al., 2014), our approach involves clearing \mathbf{M}^e before incorporating fresh information. To elaborate, we commence by computing the D -dimensional erase vector \mathbf{erase}_u from \mathbf{e}_u^s by:

$$\mathbf{erase}_u = \sigma(\mathbf{W}_{erase} \mathbf{e}_u^s + \mathbf{b}_{erase}), \quad (6)$$

where σ is the element-wise sigmoid function, and \mathbf{W}_{erase} and \mathbf{b}_{erase} are the erase parameters to be learned. Given the erase vector, the u 's memory slot \mathbf{m}_u^e is updated by:

$$\mathbf{m}_u^e \leftarrow \mathbf{m}_u^e \odot (1 - \mathbf{erase}_u), \quad (7)$$

where \odot is element-wise product. The primary purpose of the erase operation is to remove user memories that are deemed unimportant, as acquired through the neural control network. After erasing, an add vector \mathbf{add}_u is used to update the u 's memory by:

$$\mathbf{add}_u = \tanh(\mathbf{W}_{add} \mathbf{e}_u^s + \mathbf{b}_{add}), \mathbf{m}_u^e \leftarrow \mathbf{m}_u^e + \mathbf{add}_u, \quad (8)$$

where \mathbf{W}_{add} , \mathbf{b}_{add} are the add parameters to be learned. The erase-add update strategy facilitates both the forgetting and reinforcement of user representations throughout the learning process (Chen et al., 2018). By automatically adjusting the erase and add parameters, the model can discern which signals should be attenuated and which should be augmented.

3.3 Representations Transfer

Zhu et al. (2022) argued that representation transfer should be personalized. Building on this approach, a meta-network is proposed to generate a personalized transfer function from users' group-based representations. A similarity weight vector \mathbf{a}_u^e is computed by taking the inner product of \mathbf{e}_u^s with each category in \mathcal{E}^s , which is then used to calculate a weighted sum of the category embeddings.

$$\mathbf{a}_u^e = \text{softmax}(\mathbf{e}_u^s (\mathbf{E}^s)^T), \mathbf{e}_{\hat{u}}^s = \mathbf{a}_u^e \mathbf{E}^s. \quad (9)$$

The proposed meta-network is formulated as:

$$\mathbf{W}_u^e = g(\mathbf{e}_{\hat{u}}^s; \theta), \quad (10)$$

where $g(\cdot)$ is the meta-network, which is parameterized by θ . In this paper, the meta-network is a two-layer feed-forward network. The \mathbf{W}_u^e is a weight matrix whose size depends on the structure of the transfer function. The personalized category transfer function is formulated as:

$$f_u^e = (\cdot; \mathbf{W}_u^e). \quad (11)$$

Note that the \mathbf{W}_u^e is used as the parameters of the transfer function rather than input. And the generated transfer function depends on group-based representation. By the transfer function, we can obtain the transformed user's group-based category-level representation $\mathbf{e}_{u^*}^s$ (Computed by Equation 4) in the target domain:

$$\mathbf{e}_u^t = f_u^e(\mathbf{e}_{u^*}^s). \quad (12)$$

Similarly, we can acquire the user's group-based brand-level and aspect-level representations $\mathbf{i}_u^t, \mathbf{v}_u^t$ in the target domain. It is important to highlight that each level of preference corresponds to distinct transfer functions f_u^i, f_u^v and external memory networks $\mathbf{M}^i, \mathbf{M}^v$ (as shown in Figure 2).

3.4 Rating Prediction

u 's rating for $i \in \mathcal{I}^t$ is mainly composed of three parts: **category preference rating** $r_{u,i}^e$, **brand**

preference rating $r_{u,i}^i$, **aspect preference rating** $r_{u,i}^v$.

To calculate category preference rating, we need to extract categories from the i 's metadata, denoted as $\mathbf{e}_i^t = [e_{i,1}^t, \dots, e_{i,|\mathcal{E}_i^t|}^t]$. Since u has different weights on different categories, which can be represented by:

$$\mathbf{e}_{u,i}^t = \sum_{k=1}^{|\mathcal{E}_i^t|} \phi_{u,i,k} \mathbf{e}_{i,k}^t, \quad (13)$$

$$\phi_{u,i,k} = \frac{\exp(\mathbf{h}_e^T \tanh(\mathbf{W}_e \mathbf{e}_u^t + \mathbf{U}_e \mathbf{e}_{i,k}^t))}{\sum_{j=1}^{|\mathcal{E}_i^t|} \exp(\mathbf{h}_e^T \tanh(\mathbf{W}_e \mathbf{e}_u^t + \mathbf{U}_e \mathbf{e}_{i,j}^t))}, \quad (14)$$

where $\mathbf{e}_{i,k}^t$ is embeddings of $e_{i,k}^t$. Finally, category preference rating $r_{u,i}^e$ is computed by:

$$r_{u,i}^e = \mathbf{W}_{e2}(\mathbf{W}_{e1}[\mathbf{e}_u^t \oplus \mathbf{e}_{u,i}^t] + \mathbf{b}_{e1}) + b_{e2}. \quad (15)$$

Then we compute the brand preference rating by:

$$r_{u,i}^i = \mathbf{W}_{i2}(\mathbf{W}_{i1}[\mathbf{i}_u^t \oplus \mathbf{i}^t] + \mathbf{b}_{i1}) + b_{i2}, \quad (16)$$

where \mathbf{i}^t is the brand embedding of i in \mathcal{D}^t .

To calculate aspect preference rating, we need to extract aspects and their sentiment polarities from the i 's reviews (as elaborated in Appendix A), denoted as $\mathbf{v}_i^t = [v_{i,1}^t, \dots, v_{i,|\mathcal{V}_i^t|}^t]$ and $\mathbf{s}_i^t = [s_{i,1}^t, \dots, s_{i,|\mathcal{S}_i^t|}^t]$. Note that if multiple reviews all relate to the same aspect, the sentiment polarity is the average of them. Intuitively, u has different preferences on different aspects of i , so the aspect preference probability is calculated as follows:

$$\sigma_{u,i,k} = \frac{\exp(\mathbf{h}_v^T \tanh(\mathbf{W}_v \mathbf{v}_u^t + \mathbf{U}_v \mathbf{v}_{i,k}^t))}{\sum_{j=1}^{|\mathcal{V}_i^t|} \exp(\mathbf{h}_v^T \tanh(\mathbf{W}_v \mathbf{v}_u^t + \mathbf{U}_v \mathbf{v}_{i,j}^t))}, \quad (17)$$

where $\mathbf{v}_{i,k}^t$ is the embedding of $v_{i,k}^t$. Moreover, when a user makes a purchase, if the aspects highlighted in the reviews align with the user's interests and exhibit a positive polarity, it is likely to bolster the user's inclination to make the purchase. In view of this, we calculate the weighted sum of the aspect polarity scores to generate aspect preference rating:

$$r_{u,i}^v = \sum_{k=1}^{|\mathcal{V}_i^t|} \sigma_{u,i,k} s_{i,k}^t, \forall s_{i,k}^t \in (-1, 1). \quad (18)$$

In practice, users assign varying weights to different preference ratings. For instance, some users

prioritize aspect ratings, whereas others prioritize category ratings. Consequently, we derive the user’s rating weight from their representations, which is computed by:

$$\mathbf{w}_f = \text{sigmoid}(\mathbf{w}_w[\mathbf{e}_u^t \oplus \mathbf{i}_u^t \oplus \mathbf{v}_u^t] + \mathbf{b}_w), \quad (19)$$

and the finally rating is defined as:

$$\hat{r}_{u,i} = \mathbf{w}_f[r_{u,i}^e \oplus r_{u,i}^i \oplus r_{u,i}^v]. \quad (20)$$

3.5 Optimization Strategy

During Training, the historical data generated by overlapping users is leveraged to optimize the model parameters with SmoothL1(Girshick, 2015) loss, which is defined as follows:

$$\mathcal{L}_{main} = - \sum_{u \in \mathcal{U}^O, i \in \mathcal{I}^t} l(y_{u,i}, \hat{r}_{u,i}), \quad (21)$$

$$l(a, b) = \begin{cases} \frac{1}{2}(a - b)^2 & \text{if } |a - b| < 1 \\ |a - b| - \frac{1}{2} & \text{otherwise} \end{cases}, \quad (22)$$

where $y_{u,i}$ is ground-truth rating.

4 Experiments

4.1 Datasets

Three popular categories: Books, CDs & Vinyl (Music), and Movies & TV (Movies) on the Amazon reviews dataset (He and McAuley, 2016; McAuley et al., 2015) are selected for the experiments. Table 1 shows the detailed statistics of the dataset.

4.2 Baseline Methods

In the experiment, we choose the following baselines for comparison: **TGT** is a naive target model trained only with data in the target domain. **CMF** (Singh and Gordon, 2008) is a simple CDR method extending MF, where user embeddings are shared between source and target domains. **SSCDR** (Kang et al., 2019) trained the bridge function in a semi-supervised manner. **EMCDR** (Man et al., 2017) is an embedding-and-mapping framework that learns a mapping function to capture coordinate relationships between domains. **PTUPCDR** (Zhu et al., 2022) is a framework for personalized transfer of user preferences, aiming to address a single shared bridge function. **REMIT** (Sun et al., 2023) is a novel framework called reinforced multi-interest transfer for CDR. **MIMNET** (Zhu et al.,

2024) employed multi-interest meta-network with multi-granularity target-guided attention for cross-domain recommendation.

Following Zhao et al. (2020); Zhu et al. (2022), to compare the performance of the models more comprehensively, we use Mean Absolute Error (MAE), and Rooted Mean Square Error (RMSE) as metrics.

4.3 Implementation Details

We use PyTorch (Paszke et al., 2019) to develop both our model¹. Model parameters are initially randomized according to a uniform distribution as described in (Glorot and Bengio, 2010). These parameters are then updated using Adam (Kingma and Ba, 2014) with a learning rate of 1×10^{-4} . Both the embedding dimension (300) and the batch size (256) are consistently set across all models. An open-source large language model (LLM) with prompt-based extraction is utilized to directly extract aspects and their associated sentiments (see Appendix A for details)

Following the methods described in (Sun et al., 2023; Zhu et al., 2024), the effectiveness of GRAPECDR is evaluated by randomly removing all ratings from a subset of overlapping users in the target domain, who are then treated as test users. The remaining overlapping users are used to train the bridge function. In the experiments, the proportions of test (cold-start) users β are set to 80%, 50%, and 20% of the total overlapping users. For each task, the average results are reported over three random runs.

4.4 Cold-start Recommendation Experiments

Multiple experiments are conducted on three tasks, and the results are presented in Table 2. TGT performs the worst because it trains directly on target domain data. CMF performs better by combining data from different domains into one. SSCDR and EMCDR outperform CMF by using a common bridge to transfer user embeddings from the source to the target domain. PTUPCDR further improves performance by learning personalized bridges for each user. Among baseline methods, REMIT is the second best due to its use of multiple personalized bridges and an RL-based selector for interests.

Our method, GRAPECDR, consistently performs the best across all tasks and cold-start settings. It also shows greater robustness. As

¹Our source code is available at <https://github.com/syogogo/GRAPECDR>.

	Domain	Users		items			
		#users	#overlap	#items	#categories	#brands	#aspects
Task 1	Movie	123,960	18,031	50,052	398	12366	1537
	Music	75,258		64,443	425	13787	3769
Task 2	Book	603,668	37,388	367,982	638	123510	3375
	Movie	123,960		50,052	398	12366	5725
Task 3	Book	603,668	16,738	367,982	638	123510	1917
	Music	75,258		64,443	425	13787	4635

Table 1: Statistics of the three tasks. #**overlap** denotes number of overlapping users.

Tasks	β	Metric	TGT [†]	CMF [†]	SSCDR [†]	EMCDR [†]	PTUPCDR [†]	REMIT [†]	MIMNet [*]	Ours
1	20%	MAE	4.4730	1.4128	1.2151	1.1963	1.0051	0.9393	0.8027	0.7419
		RMSE	5.1615	1.8725	1.4946	1.4803	1.3498	1.2709	1.1509	1.0296
	50%	MAE	4.5064	1.5179	1.3235	1.4570	1.1464	1.0437	0.8729	0.7441
		RMSE	5.1774	2.0225	1.6794	1.8086	1.5987	1.4580	1.2244	1.0263
	80%	MAE	4.5129	1.8609	1.6382	1.9397	1.4245	1.2181	1.0614	0.7534
		RMSE	5.1983	2.4507	2.1576	2.3290	2.0338	1.6601	1.4721	1.0406
2	20%	MAE	4.2026	1.4857	1.2631	0.9834	0.9093	0.8759	0.8718	0.8531
		RMSE	4.7677	1.9308	1.4700	1.2295	1.1914	1.1650	1.1430	1.1442
	50%	MAE	4.2225	1.5974	1.3407	1.0556	0.9567	0.9172	0.9025	0.8538
		RMSE	4.7890	2.0636	1.5607	1.3238	1.2712	1.2379	1.1983	1.1433
	80%	MAE	4.2405	2.3416	1.2495	1.2249	1.0519	1.0055	0.9710	0.8584
		RMSE	4.8201	2.8561	1.5572	1.5334	1.4338	1.3772	1.2910	1.1495
3	20%	MAE	4.4516	1.7873	1.5988	1.6121	1.4701	1.3749	0.8107	0.7228
		RMSE	5.1455	2.3316	2.1146	2.1638	2.0707	1.9940	1.1711	0.9928
	50%	MAE	4.4825	1.9348	1.8166	1.9050	1.5872	1.4401	0.9348	0.7222
		RMSE	5.1585	2.5232	2.2718	2.3289	2.2279	2.0495	1.3009	1.0078
	80%	MAE	4.5188	2.3989	2.1749	2.2192	1.8344	1.6396	1.1167	0.7326
		RMSE	5.2001	3.0838	2.5652	2.6331	2.5235	2.2653	1.5178	1.0100

Table 2: Comparisons of baselines performances. And the baselines marked [†] are taken from Sun et al. (2023), and marked ^{*} is taken from Zhu et al. (2024), we used the same training and test sets as these papers. Note that a lower MAE and RMSE value indicates a better model performance. For each task, the averaged results over three random runs are reported.

β increases, fewer training users, making cross-domain mapping less effective due to fewer training instances. Therefore, the baseline methods all show varying degrees of performance degradation. GRAPECDR addresses this by focusing on transferring categories, brands, and aspects across domains, rather than user representations, thus reducing the impact of β .

4.5 Ablation Study

This section presents an analysis of the proposed model to assess the contribution of each component.

4.5.1 Impact of different levels of preference on model performance

First, we observe a significant performance decline in Table 3, when the aspect preferences (denoted as *category + brand*) are removed. The MAE increases by approximately 0.04 across Tasks 1, 2, and 3, indicating that fine-grained preferences are crucial to the model’s performance. Additionally, when brand preference is removed (denoted as *category*), the model shows a slight decrease in

performance, with the MAE increasing by about 0.01 across the same tasks, this result suggests that while brand preference does influence the model, its impact is less pronounced than that of aspect preference. Overall, the ablation experiment confirms that each preference level significantly affects the model’s performance.

4.5.2 Impact of integrating group-based preference on model performance

Furthermore, we construct a variant model for the transfer function, wherein each user possesses a personalized transfer function (as shown in Equation 12), but the input of this function is user intrinsic representation (denoted as *w/o group-based transfer preference*). As illustrated in Table 3, upon exclusion of group-based preference, the model’s performance experiences a notable decline across all tasks. In Task 3, MAE increased by about 0.06, and in the other tasks, it increased by about 0.03. This result shows that adding the preference of similar users to the representations can significantly improve the performance of the model.

Models	Task 1		Task 2		Task 3	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
<i>category</i>	0.7912	1.0767	0.9216	1.2134	0.7639	1.0299
<i>category+brand</i>	0.7744	1.0723	0.9025	1.1916	0.7500	1.0232
GRAPECDR(<i>category+brand+aspect</i>)	0.7419	1.0296	0.8531	1.1442	0.7228	0.9928
<i>w/o group-based transfer function</i>	0.7872	1.0647	0.8957	1.1796	0.7867	1.0531

Table 3: Performance comparison of different levels of preference. Note we set $\beta = 20\%$ in these experiments. *category* denotes that the model only contains category preferences, and *category+brand* denotes that the model contains category and brand preferences. *w/o group-based transfer function* denotes GRAPECDR do not use group-based transfer function.

Source Domain	Target Domain
book,dialogue,storytelling,value,writing style	movie,acting,performance,story,film
planet,story	cartoons,dvd,animation,ship,acting
soundtrack,plot,movie release,waiting time,movie longevity	performance,special,music,staging,production
plot,film,acting,mood	story,musicians,special,music,concert

Table 4: Case Study.

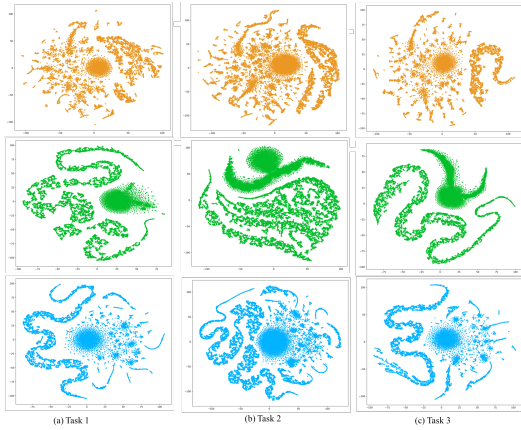


Figure 3: Visualization for category, brand, and aspect memory network (M^e , M^i and M^v) in three tasks. The orange color represents the category memory network, the green color denotes the brand memory network, and the blue color represents the aspect memory network.

4.6 Visualization Analysis

Three external memory networks are employed to store user features corresponding to category-level, brand-level, and aspect-level preferences. T-SNE (Van der Maaten and Hinton, 2008) visualizes these networks across three tasks, showing that similar user features cluster into distinct groups. Figure 3 illustrates that the category memory network (orange) and the brand and aspect networks (green and blue) effectively capture and store typical user features, aligning with our goal of grouping users by shared preferences.

4.7 Case Study

This section presents a case study focusing on aspect-level preferences as an illustrative exam-

ple. Results are shown in Table 4. The first two cases come from Task 2 (Books->Movies), and the remaining cases come from Task 1 (Movies->Musics).

We first obtain the aspects mentioned in the user’s reviews, as shown in the ‘Source Domain’ column of Table 4. Then, the attention weights from Equation 17 are used to identify the top five aspects the user focuses on in the target domain. In the first case, the aspects mentioned by users in the source domain (Books) mainly include “*storytelling, writing style*”, which may mean that the reader pays more attention to the plot. In the target domain (Movies), users pay more attention to “*story, acting, performance*”. Obviously, this preference is similar to the preference in the source domain. In the second case, users prefer “*planet, story*” in the source domain, the reader may be a child or a woman/man with children. In the target domain, they pay more attention to the “*cartoons*” and “*animation*” aspects of movies. Similarly, cases 3 and 4 yield analogous outcomes. The case study highlights not only the efficacy but also the explanation of our method.

5 Conclusion

This paper addresses the cold-start problem in cross-domain recommendation by integrating group-based preferences from coarse to fine levels. A model is designed to generate coarse-to-fine preference representations using historical data from the source domain. Firstly, external memory networks are employed to capture group-based representations of similar users, which are then transferred to the target domain via a meta-network.

Aggregating ratings from various preference representations then predict user-item ratings. Empirical results demonstrate that the proposed model outperforms several state-of-the-art methods. Furthermore, ablation experiments further reveal that preferences at each level significantly influence performance, and incorporating similar users' preferences enhances the model's effectiveness.

Acknowledgments

This study was supported by the National Key Research and Development Program of China (2022YFF0707103).

References

- Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiayi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 108–116.
- Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *AAAI*, volume 33, pages 94–101.
- Ross Girshick. 2015. Fast r-cnn. In *ICCV*, pages 1440–1448.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *PMLR*, pages 249–256.
- Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.
- Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517.
- SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1563–1572.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.
- Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *SIGKDD*, pages 1073–1082.
- Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670.
- Pan Li and Alexander Tuzhilin. 2020. Dtdcdr: Deep dual transfer cross domain recommendation. In *WSDM*, pages 331–339.
- Xiaodong Li, Jiawei Sheng, Jiangxia Cao, Wenyan Zhang, Quangang Li, and Tingwen Liu. 2024. *Cdrnp: Cross-domain recommendation to cold-start users via neural process*. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24*, page 378–386, New York, NY, USA. Association for Computing Machinery.
- Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *SIGKDD*, pages 1563–1573.
- Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: an embedding and mapping approach. In *IJCAI*, pages 2464–2470.
- Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52.
- Feiyang Pan, Qingpeng Cai, Pingzhong Tang, Fuzhen Zhuang, and Qing He. 2019a. Policy gradients for contextual recommendations. In *WWW*, pages 1421–1431.
- Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019b. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *SIGIR*, pages 695–704.
- Weike Pan, Evan Xiang, Nathan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, volume 24, pages 230–235.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32.
- Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. 2011. Personalised rating prediction for new users using latent factor models. In *HT*, pages 47–56.
- Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *SIGKDD*, pages 650–658.

- Caiqi Sun, Jiewei Gu, BinBin Hu, Xin Dong, Hai Li, Lei Cheng, and Linjian Mo. 2023. Remit: reinforced multi-interest transfer for cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9900–9908.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. *NeurIPS*, 30.
- Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2019. Dropoutnet: Addressing cold start in recommender systems. In *NeurIPS*, pages 8026–8037.
- Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *SIGKDD*, pages 968–977.
- Xinghua Wang, Zhaohui Peng, Senzhang Wang, Philip S Yu, Wenjing Fu, and Xiaoguang Hong. 2018. Cross-domain recommendation for cold-start users via neighborhood based feature mapping. In *DAS-FAA*, pages 158–165.
- Xinghua Wang, Zhaohui Peng, Senzhang Wang, Philip S Yu, Wenjing Fu, Xiaokang Xu, and Xiaoguang Hong. 2020. Cdlfm: cross-domain recommendation for cold-start users via latent feature mapping. *Knowledge and Information Systems*, 62:1723–1750.
- Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. Catn: Cross-domain recommendation for cold-start users via aspect transfer network. In *SIGIR*, pages 229–238.
- Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet Orgun, and Jia Wu. 2018. A deep framework for cross-domain and cross-system recommendations. In *IJCAI*, pages 3711–3717.
- Xiaofei Zhu, Yabo Yin, and Li Wang. 2024. Mimnet: Multi-interest meta network with multi-granularity target-guided attention for cross-domain recommendation. *arXiv preprint arXiv:2408.00038*.
- Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021a. Transfer-meta framework for cross-domain recommendation to cold-start users. In *SIGIR*, pages 1813–1817.
- Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *WSDM*, pages 1507–1515.
- Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021b. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *SIGIR*, pages 1167–1176.

A Appendix

A.1 Aspect-Polarity Extraction with LLM

we use Mistral-Nemo-2407-Instruction-quantized.w4a16², a quantized version of Mistral-Nemo-2407-Instruction, as the base model to extract the aspect and related polarity of the given sentences. Due to the quantized version model and batch reference, the speed of inference to extract expected information is greatly improved. Then, we deploy the model with Vllm(Kwon et al., 2023) on a server machine of NVIDIA A800.

The system prompt of extraction is as follows: *Now you are an aspect category and sentiment polarity extractor. Your work is to extract aspect category and sentiment polarity pairs from the given sentences. If you could not detect any aspect category and sentiment polarity information from the provided sentences, please just return a empty list:[] . Remember the polarity should be one of 'positive', 'negative' and 'neutral'. Note that if the aspect category and sentiment polarity pairs exist, your answer should be a json list. Each json in the list has two keys: 'aspect' and 'polarity', for example: [{"aspect": "food", "polarity": "negative"}]. Do not contain explain or any other unrelated content in your answer!*

The user prompt of extraction is as follows: *Given a sentence as follows: [TARGET_SENTENCE] Please extract the aspect and polarity pairs in a json list.*

The target sentence is inserted in place of the placeholder [TARGET_SENTENCE] to generate the user prompt, which is then appended to the system prompt. This combined prompt is fed into the model to obtain a response. A JSON output parser is used to extract the expected JSON-formatted answer. If the parser fails, a retry mechanism is triggered until a correct answer is produced or the retry limit is reached.

²<https://huggingface.co/nvidia/Mistral-NeMo-12B-Instruct>