UGDA: A Unified Graph-based Method with Domain-specific Adaptation for Multi-domain Recommendation

Bin Ruan[1], Hao Liu[2], Yitian Tu[1],
Zhiying Deng[1], Zhiqiang Guo[3], and Jianjun Li[1]✉

[1] Huazhong University of Science and Technology, Wuhan,China
{binruan,tuyt,dengzhiyingdd,jianjunli}@hust.edu.cn
[2] Wuhan Digital Engineering Institute, Wuhan, China
liuhao2020@sjtu.edu.cn
[3] Tsinghua University, Beijing, China
georgeguo.gzq.cn@gmail.com

Abstract. Multi-domain recommendation (MDR) aims to provide recommendations for different domains (e.g., product types) with overlapping users/items. MDR benefits from disentangling domain-shared and domain-specific user representations, but model-level disentangling methods suffer from the gradient conflict problem, which have spurred the development of new algorithms that explicitly separate global and local knowledge at both the model and embedding levels. Though effective, existing separation-based MDR methods still face three major challenges. Firstly, they are unable to capture the complete global structure information. Secondly, the learned global representation is directly applied to the recommendation of each domain, without considering its adaptability to each domain. Thirdly, the separation learning structure will lead to inconsistency between global and local representations. To address these problems, we propose UGDA, a Unified Graph-based method with Domain-specific Adaptation for MDR. Specifically, we construct a global unified graph from different domain graphs to directly learn the global representation, so as to capture more comprehensive global information. Moreover, we re-incorporate all interactions in each domain into the global unified graph, enabling the global representations to be domain-specific. Finally, we redesign the BPR loss and assign different weights to individual data sample to improve the consistency between global and local representations. Extensive experiments on representative public datasets demonstrate that UGDA consistently outperforms existing state-of-the-art MDR methods.

Keywords: Multi-domain recommendation · Domain-specific adaption · Unified graph, Global representation, Local representation

1 Introduction

Data sparsity has long been considered a prominent issue hindering the development of traditional single domain recommendation systems. As a promising

solution, cross-domain recommendation (CDR) has proven to be effective in alleviating the data sparsity issue, which aims to enhance recommendation in the target domain by utilizing auxiliary information learned from other domains. For example, using movie viewing records in *movie* domain can assist music recommendation in *music* domain. Most of the previous research on CDR focuses on either the single-target [29,23] or dual-target [27,22] scenarios, which involve only two domains. Specifically, single-target CDR tries to improve the recommendation performance exclusively in the target domain, while dual-target CDR attempts to improve the performance of both domains simultaneously. Multi-target cross-domain recommendation [5,20], also known as multi-domain recommendation (MDR), is a more general and challenging problem that seeks to boost the performance in multiple participating domains concurrently. Considering that existing methods typically address single- and dual-target CDR by modeling a pair-wise domain-domain relationship, directly applying them to the MDR scenario with n domains may involve handling at least C_n^2 pairs of relations, which is impractical when the number of domains is large.

Most existing MDR methods, like STAR[20], TreeMS[17] and MGFN[24], focus on disentangling user representations into domain-shared (global) and domain-specific (local) representations. However, if global and local representations have opposing gradient directions when disentangled from the original embedding, they may cancel each other out, causing a "gradient conflict" that prevents the original embedding from being updated. To address this problem, separation training architecture models like EDDA and TrineCDR were proposed to model and optimize domain-shared and domain-specific representations separately. Specifically, they initialize user local embeddings and train them with each domain's data, while initializing user global embedding and training them with all domains' data. This allows gradient information to be separated during training, effectively avoiding the "gradient conflict" problem.

Though encouraging, we find that separation training models still face some limitations. (1) Firstly, they learn global representation by aggregating local structure information from each domain or propagating information on single domain. However, the representations learned from local structure information of each domain cannot represent the complete global structure information. (2) Secondly, the global representation learned by most MDR methods is directly applied to the final recommendation of each domain, as illustrated in Figure 1(a). We argue that using global representation in this manner may not adapt well to all domains. This is because global information may not be necessary for every domain, and the same global representation can have different meanings across domains. For example, a user's preference for pleasant elements can manifest as upbeat songs in music and happy-ending comedies/movies in film. Thus, global context representations differ by domain. (3) Thirdly, although they avoid the "gradient conflict" problem by separating the training processes between global and local representations, it results in a new problem that a user's preference may become inconsistent when viewed from both global and local perspectives. This occurs because the separated training architecture permits the model to train

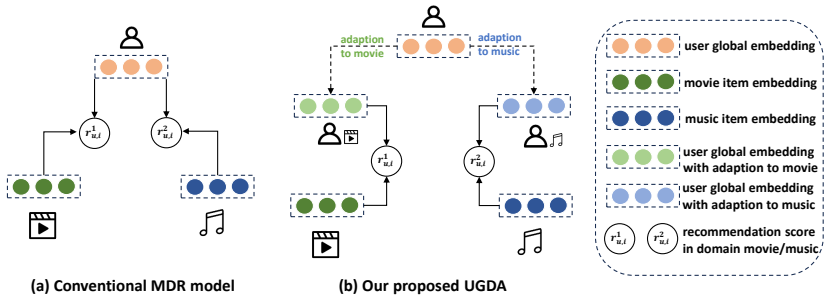


Fig. 1: (a) Conventional MDR, which directly uses global representation to recommend in each domain. (b) Our proposed UGDA, which makes global representation domain-specific.

the representations to fit the graph structure without imposing any constraints, potentially leading to inconsistencies.

To address these problems, we propose a novel Unified Graph-based method with Domain-specific Adaption for MDR, called UGDA. **Firstly**, by building a large unified graph containing the interactions of all the domains, the information can transfer among different domains directly. In this way, we do not need to explicitly aggregate the local representations, thus preventing any domain from dominating the final representation. Moreover, training on this unified graph yields more robust representations than on sparse individual domain graphs. **Secondly**, to adapt the global representation to each domain, we reintroduce domain-specific interactions into the unified graph, doubling the number of domain-specific edges. This allows for more extensive information propagation within each domain's subgraph, resulting in representations that combine global information with domain specificity, as illustrated in Figure 1(b). **Thirdly**, to enhance consistency between global and local representations, we redesign a weighted BPR loss, assigning weights based on the difference between global and local scores. By comparing the score differences between positive and negative samples based on global and local preferences, we can verify the model's consistency across different levels. In sum, our contributions are as follows:

- Instead of learning from each local structure and then performing aggregation, we propose UGDA to construct a unified graph to directly learn the global information, which is more robust. Moreover, by re-incorporating the interactions within a specific domain into the unified graph, UGDA can make the learned global representation domain-specific.
- We figure out the inconsistent representation problem due to the separation training strategy, and design a weighted BPR loss by assigning different weights to different data sample BPR loss items, which promotes consistency between global and local representations.
- We conduct extensive experiments on several large public datasets, and the results demonstrate the effectiveness and superiority of our UGDA model.

2 RELATED WORK

Cross-domain recommendation. Cross-domain models are basically based on mapping, with EMCDR[15] being a representative work. The later DCDCSR[28] maps data to a common standard domain. TMCDR[31] uses meta-learning in the process of mapping learning and considers each batch data as a task to learn better parameter with MAML [4]. SSCDR [9] introduces semi-supervised loss to eliminate biases of the mapping function towards overlapping users/items. [19] exerts VAE to learn the latent representation in domains.
Dual-domain recommendation. Dual-domain recommendation is a special case of MDR that only involves two domains. BiTGCF[13] uses a shared embedding space to learn the shared representation of users and items in two domains. CoNet[7] uses cross-stitch structure to share information in the middle layers. Later methods such as DisenCDR[2] focuses on disentangling embedding into domain-specific and domain-shared parts. DCCDR [25] uses contrastive learning to enhance the embedding between global and local information. DIDA [30] disentanglements the embedding into domain-specific, domain-shared and domain-invariant parts to prove not all the information is needed. UniCDR [1] proposes a unified framework to solve the dual-domain recommendation problem.
Multi-domain recommendation. Multi-domain recommendation requires participation of multiple domains, which is more challenging than dual-domain recommendation. [3] obtains shared and specific representations for recommendation. [12] design different methods to make adaptation to each domain. MetaDomain [26] focuses on sparse domain recommendation with meta-generator to enhance its performance. CAT-ART [11] designs an encoder to learn the global embedding. TrineCDR [21] and EDDA [16] are recent separation training structure models for MDR which initialize local and global representations and update them separately. Besides, multi-task learning (MTL) models [8,14] can also be used in MDR by considering each domain as a task with shared parameter to share information across different domains. Note in MDR, distributions across different domains are NOT the same, but MTL methods model inputs from different domains as having the same distribution, which limits their performance when applying to MDR.

3 PRELIMINARY

We consider a multi-domain recommendation scenario with K domains. The user set and item set in domain k are denoted as $\mathcal{U}_k$ and $\mathcal{I}_k$, respectively, and the corresponding user and item numbers are denoted as $n_k = |\mathcal{U}_k|$ and $m_k = |\mathcal{I}_k|$, respectively. The whole user and item sets can be represented as $\mathcal{U}_o = \mathcal{U}_1 \cup \mathcal{U}_2 \cup \cdots \cup \mathcal{U}_K$ and $\mathcal{I}_o = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_K$, respectively. We use n_o and m_o to denote the number of users and items in $\mathcal{U}_o$ and $\mathcal{I}_o$, respectively. There may exist some common users for two different domains k and k', i.e., $\mathcal{U}_k \cap \mathcal{U}_{k'} \neq \emptyset$, which allows to share knowledge across domains and achieve better recommendation performance than handling each domain individually. The interaction matrix in

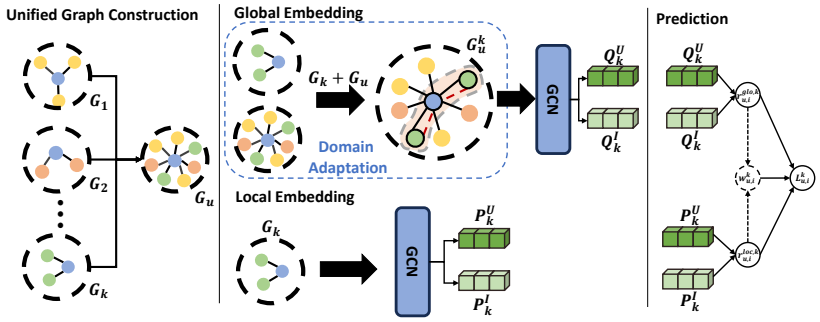


Fig. 2: Overall framework of UGDA that consists of three parts: (a) Unified graph construction module, which constructs a unified graph across all the domains. (b) Embedding learning module that learns domain-specific global embedding on global unified graph and local embedding on local graph. (c) Prediction module.

domain k is $\mathcal{R}_k \in \mathbb{R}^{n_k \times m_k}$, where $r_{ui} = 1$ implies user u has interacted with item i, and $r_{ui} = 0$ otherwise. The goal of the MDR task is to learn a score function $f(u, i|k, \boldsymbol{\Theta})$ to predict the preference of user u for item i in domain k, based on model parameters, denoted as $\boldsymbol{\Theta}$.

4 METHODOLOGY

4.1 Overall Framework

Figure 2 presents the overall framework of UGDA, which mainly consists of three components: (1) A unified graph construction module, which constructs a unified graph across all the domains. (2) An embedding learning module, which learns global embedding with domain adaptation and local embedding. (3) A prediction module, which uses the weighted loss to train the model and uses the learned global and local embeddings of each domain to make prediction.

4.2 Construction of Unified Graph

Different from existing model-level disentangling methods such as STAR and MGFN, separation-based models directly models global representations. Take EDDA as an example, to ensure that the global representation captures information from all domain, it first embeds the global representation into the graph of each domain for learning. Then, it sums up the representations obtained from each domain, treating it as the ultimate global representation. Formally,

$$\mathbf{E}_{glo} = \frac{1}{K} \sum_{k=1}^{K} GNN(\mathbf{E}_{glo}, \mathcal{G}_k) \tag{1}$$

where $\mathbf{E}_{glo}$ denotes global embedding for users and items, $\mathcal{G}_k = \{\mathcal{V}_k, \mathcal{E}_k\}$ represents the graph of domain k, with $\mathcal{V}_k$ being the node set and $\mathcal{E}_k$ being the

edge set of $\mathcal{G}_k$. We argue that the global embedding learned in this manner cannot capture the complete global structure. This is due to the independence of each domain's graph, where summing embeddings fails to fully transfer information between domains. In other words, it includes information from each domain without assessing its usefulness. TrineCDR propagates the global representation only within subgraphs, limiting its ability to learn global information.

To address this issue, we propose to build a large unified graph containing all the data to capture the complete global structure information. Specifically, let $\mathcal{V}_u = \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_K$ denote the set of all nodes, and $\mathcal{E}_u = \mathcal{E}_1 \oplus \mathcal{E}_2 \oplus \cdots \oplus \mathcal{E}_K$ denote the set of all edges in the unified graph, where $\oplus$ denotes a multiset addition operation that only appends elements into multiset but does not dismiss duplicated elements (e.g., $\{1, 2\} \oplus \{2, 3\} = \{1, 2, 2, 3\}$). In order to connect different domain graphs, we use overlapping users as joint nodes. In this way, we can construct a unified graph, denoted as $\mathcal{G}_u = \{\mathcal{V}_u, \mathcal{E}_u\}$, as shown in Figure 2 (a).

4.3 Embedding Learning

Global embedding learning with domain-specific adaptation The global representation learned by most existing model-level disentangling MDR methods is directly applied to the final recommendation of each domain, we argue that using global representation in this way may not adapt to all domains well. Therefore, we consider making some domain-specific modifications during the information propagation process on GNNs, which enables the global representation to be domain-specific when learning on the global graph. Consider the following classic information propagation structure of GNNs,

$$\mathbf{E}^{l+1} = \widetilde{\mathbf{A}}\mathbf{E}^{l} \tag{2}$$

where $\mathbf{E}^l$ is the embedding table obtained in layer l and $\widetilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the propagation matrix, with $\mathbf{A}$ being the adjacency matrix,

$$\mathbf{A} = \begin{pmatrix} 0 & \mathbf{R} \\ \mathbf{R}^T & 0 \end{pmatrix} \tag{3}$$

and $\mathbf{D}$ being the degree matrix of $\mathbf{A}$. We can see the propagation matrix $\widetilde{\mathbf{A}}$ is computed by interaction matrix $\mathbf{R}$, with $r_{ui} = 1$ implying user u has interacted with item i, and $r_{ui} = 0$ otherwise.

By performing the same procedure on unified graph $\mathcal{G}_u$, we can obtain the global embedding on the unified graph. This inspires us that if we could make some adjustments to the interaction matrix $\mathbf{R}_u$ of $\mathcal{G}_u$ to make it specific to a certain domain, then the propagation matrix would also be specific to that domain while retaining global information. From this perspective, we consider directly adding all interactions from a specific domain to the global interaction of the unified graph. By doubling the number of interactions in this specific domain, we effectively double the number of corresponding edges in the global unified graph. This adjustment enables information propagation to occur more extensively within the local subgraph of that domain. As a result, the resulting global

representation captures both the structural information of the entire graph and highlights the structural details of the local subgraph within the specific domain. We consider such a representation as the domain-specific global representation.

For the unified graph $\mathcal{G}_u$, its interaction matrix is $\mathbf{R}_u \in \mathbb{R}^{n_o \times m_o}$ and the interaction matrix of domain k is $\mathbf{R}_k \in \mathbb{R}^{n_k \times m_k}$. To facilitate matrix addition, we expand matrix $\mathbf{R}_k$ to the same dimensions as matrix $\mathbf{R}_u$. Specifically, for non-existent interactions, we set the corresponding values to 0. Then we can get interaction matrix $\mathbf{R}_{u,k}$ with adaptation to domain k by adding interaction matrix of unified graph and domain k together,

$$\mathbf{R}_{u,k} = \mathbf{R}_u + \mathbf{R}_k \tag{4}$$

With $\mathbf{R}_{u,k}$, we can get domain-specific global user embedding $\mathbf{Q}_k^U$ and item embeddings $\mathbf{Q}_k^I$ of domain k by the following formula,

$$\mathbf{Q}_k^U, \mathbf{Q}_k^I = GNN(\mathbf{Q}_k^U, \mathbf{Q}_k^I, \mathbf{R}_{u,k}) \tag{5}$$

where $\mathbf{Q}_k^U$ and $\mathbf{Q}_k^I$ denote global user representations and item representations specific to domain k, respectively.

Local embedding learning Similar to [16], we use the local embedding to model the local structure. Specifically, we use embedding trained on single domain model as local embedding. Formally,

$$\mathbf{P}_k^I, \mathbf{P}_k^U = GNN(\mathbf{P}_k^I, \mathbf{P}_k^U, \mathbf{R}_k) \tag{6}$$

where $\mathbf{P}_k^U$ and $\mathbf{P}_k^I$ are local user and item embeddings in domain k, respectively.

Note for the selection of the GNN model, we choose GRec [16] for both global and local embedding learning, as it has been shown to achieve good results in previous work. Consequently, the propagation formula can be defined as:

$$\begin{aligned} \mathbf{e}_u^{(l)} &= \alpha * \mathbf{e}_u^{(l-1)} + (1-\alpha) * \sum\nolimits_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(l-1)} \\ \mathbf{e}_i^{(l)} &= \alpha * \mathbf{e}_i^{(l-1)} + (1-\alpha) * \sum\nolimits_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(l-1)} \end{aligned} \tag{7}$$

where α is a weight parameter, $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ are the embeddings of user u and item i in layer l respectively, $\mathcal{N}_u$ and $\mathcal{N}_i$ are the set of neighbors of user u and item i, and $|\mathcal{N}_u|$ and $|\mathcal{N}_i|$ are the corresponding numbers. Note during the training of local representations, $\mathbf{e}^{(l)}$ corresponds to local embedding $\mathbf{p}^{(l)}$, and during the training of global representations, $\mathbf{e}^{(l)}$ corresponds to global embedding $\mathbf{q}^{(l)}$.

4.4 Prediction

UGDA adopts a separation-based structure by separating the training of global and local representations. For a separation-based structure, its BPR loss for a specific data sample in a domain can be represented as,

$$L_{bpr} = -ln\sigma_1(r_{u,i^+}^{loc} - r_{u,i^-}^{loc}) - ln\sigma_1(r_{u,i^+}^{glo} - r_{u,i^-}^{glo}) \tag{8}$$

where $r_{u,i}^{loc}$ and $r_{u,i}^{glo}$ are the recommendation score of local representation and global representation respectively, i^+ and i^- denote positive and negative samples, and σ_1 is softplus activation function. Due to the adoption of a separated design, the gradients of these two terms are independent of each other, which implies that there is no information exchange between global and local representations. Therefore, they are unable to ascertain the scoring differences between the same positive and negative samples. A user may encounter a significant scoring difference between positive and negative samples in the global representation, while experiencing a minor scoring difference in the local representation.

To bridge the gap between global and local scores and ensure consistency, we vary the weight of loss for different data samples. This weight is determined by the disparity between global and local scores for positive and negative sample preferences. Using the difference between the scores of the two preferences as the weight allows for a greater weight to be assigned when the global and local tendencies are inconsistent, leading to a larger loss. Conversely, when the two are consistent, a smaller weight is assigned, reducing the strength of the penalty. We design the weight as follows:

$$w_{u,i^+,i^-}^k = 2\sigma_2(|r_{u,i^+}^{glo,k} - r_{u,i^-}^{glo,k} - r_{u,i^+}^{loc,k} + r_{u,i^-}^{loc,k}|) \tag{9}$$

where w_{u,i^+,i^-} is a weight for the recommendation score of tuple (u, i^+, i^-) in domain k, $r_{u,i}^{loc,k} = \mathbf{p}_{k,u}^U \cdot \mathbf{p}_{k,i}^I$ and $r_{u,i}^{glo,k} = \mathbf{q}_{k,u}^U \cdot \mathbf{q}_{k,i}^I$ are the recommendation scores of local and global recommendations in domain k respectively, $\mathbf{p}_{k,u/i}^{U/I}$ and $\mathbf{q}_{k,u/i}^{U/I}$ are corresponding embeddings in $\mathbf{P}_k^{U/I}$ and $\mathbf{Q}_k^{U/I}$ respectively, and σ_2 is sigmoid activation function to map the score to $[0, 1]$. Because we aim to obtain the differences in scores between global and local preferences, we use the absolute value to eliminate negative values. Additionally, due to the mapping range of the sigmoid function, the mapping range at this point becomes $[0.5, 1)$. Therefore, we use a coefficient 2 to map the range of weights to $[1, 2)$, where the weight of the loss is 1 when the global and local preferences are exactly consistent; Otherwise, it is greater than 1. By doing so, user's global and local preferences for the same set of positive and negative samples should be as similar as possible. Consequently, we define the weighted BPR loss as,

$$\begin{aligned} L_{wbpr} = & -\sum_{k=1}^{K} \sum_{(u,i^+,i^-)\in\mathcal{D}_k} w_{u,i^+,i^-}^k l_{u,i^+,i^-} \\ l_{u,i^+,i^-}^k = & \, ln\sigma_1(r_{u,i^+}^{glo,k} - r_{u,i^-}^{glo,k}) + ln\sigma_1(r_{u,i^+}^{loc,k} - r_{u,i^-}^{loc,k}) \end{aligned} \tag{10}$$

where l_{u,i^+,i^-}^k is the corresponding sample loss for tuple (u, i^+, i^-), and $\mathcal{D}_k$ is dataset in domain k. By aligning global representations with local representations, we can maintain consistency in user preferences between global and local patterns. The final loss is

$$L = L_{wbpr} + \lambda||\Theta||_2^2, \tag{11}$$

where Θ is the trainable parameters of UGDA and λ is the weight of L_2 loss.

Table 1: Statistics of two datasets

Dataset	Domain	#Users	#Items	#Interactions	Density/%
Amazon	Arts	56,158	22,096	346,139	0.0279
	Inst	27,507	10,609	161,974	0.0555
	Luxury	3,819	1,580	23,994	0.3976
	Music	16,551	11,768	118,846	0.0610
	Pantry	14,168	4,967	96,451	0.1371
	Video	55,182	17,393	348,303	0.0362
AliAd	d1	7,647	2,084	21,698	0.1362
	d2	44,101	2,106	86,578	0.0932
	d3	5,265	1,745	10,000	0.1088
	d4	16,421	1,768	60,621	0.2088
	d5	12,900	1,818	23,592	0.1006
	d6	9,245	1,702	16,359	0.1040
	d7	8,799	1,679	15,891	0.1076
	d8	22,780	1,901	50,122	0.1157

5 EXPERIMENT

5.1 Experimental Settings

Datasets We adopt two commonly-used public datasets from [16] to evaluate our solution, and their statistics are listed in Table 1. Specifically,

- **Amazon** contains user ratings for items from Amazon. We use 6 categories of products in this dataset, including *Arts, Inst, Luxury, Music, Pantry* and *Video*. We regard each category as a domain, and regard each rating as a positive interaction.
- **AliAd** includes 8 days of ad display/click records from Alibaba. We choose 8 categories of ads as different domains and treat clicks as positive interactions.

Baselines We select several representative models for comparison, categorized into three types: Single domain methods (**SMF**, **LightGCN** and **GRec**); Multi-task methods (**MoE** and **MMoE**), where each domain recommendation is considered as a task; and Multi-domain methods (**CAT-ART**, **TrineCDR** and **EDDA**). We did not compare with STAR[20], TreeMS[17] and MGFN[24], as they are inferior to EDDA and their codes are not publicly available.

- SMF[18]: Single domain matrix factorization. We train a matrix factorization model for each domain.
- LightGCN[6]: A light yet effective GNN model, which performs well in recommendation task.
- GRec[16]: The variant of LightGCN, which assigns different weights to each layer with one hyperparameter α. Compared with LightGCN, GRec can control the information ratio of each layer.

- MoE[8]: A multi-task learning model with gating network to control the information flow within each task.
- MMoE[14]: A multi-task learning model with multi-gate to learn the importance of each domain to control the information flow.
- CAT-ART[11]: An MDR model uses auto-encoder to learn domain-shared representation and self-attention to transfer information from other domains.
- TrineCDR[21]: A recent MDR model, which mitigates negative migration at three different levels.
- EDDA[16]: A recent MDR model, which learns global representation by aggregating local structure information.

Evaluation Metrics We use Recall@K and NDCG@K to evaluate the test performance of different models. They are computed by the all-ranking protocol in which items not interacted by the user are treated as negative samples. Training/validation/test sets are split by the ratio of $7:1:2$. We set $K = 20$ in our experiments. For each positive sample, we randomly sample 2 negative samples.

Implementation Details For all the GNN-based models, we use the same embedding dimension $d = 32$ for convenience in training and set layer number as 3. α is set as 0.1 for GRec. We use Adam optimizer to train our model. Learning rate is set as 0.001 and batch size is 4096. The regularization coefficient is 0.001. For the CAT-ART model, we train an MF model on each domain and use the embeddings from the MF model to initialize its parameters. For TrineCDR, we set dropout ratio as 0.1 and soft threshold as 0.1. We train all the models for 1000 epochs until they converge, and test the models every 3 epochs.

5.2 Performance Comparison

Table 2 shows the main results, from which we have the following observations:

- UGDA exhibits the best performance among all the compared models, far exceeding all benchmark models in all domains and evaluation metrics. The improvements in the *Pantry* domain are the most significant, which are 10.361% and 16.917% on Recall@20 and NDCG@20 respectively. In the *Luxury* domain, the improvements are relatively small, which is mainly due to the relatively small domain scale that makes the model prone to overfitting. We attribute the performance improvement to UGDA's utilization of domain-adaptation learning on unified graph and the design of weighted BPR loss.
- Compared to single-domain, multi-task, and disentanglement-based methods, EDDA and UGDA generally perform better on Amazon with more interactions. This is because multi-task and disentanglement-based methods that rely more on overlapping users are hard to transfer information effectively when there are fewer overlapping users. TrineCDR performs worse than single-domain models in some cases, likely because its global representation learning fails to capture true global information. Additionally,

Table 2: Performance comparison of the baselines and our approach. The best results are in bold and the second best results are underlined. *Improvement* is the relative improvement of our model over the best baseline.

Dataset	Metric	Domain	SMF	LightGCN	GRec	MoE	MMoE	CAT-ART	TrineCDR	EDDA	UGDA	*Improvement*
Amazon	Recall@20	Arts	7.022	8.667	9.292	7.550	7.881	7.493	6.729	9.987	**10.703***	7.169%
		Inst	9.414	13.887	14.424	9.756	10.227	9.812	10.124	14.646	**15.413***	5.237%
		Luxury	25.391	28.383	29.185	25.965	25.802	24.891	25.473	29.425	**29.723***	1.013%
		Music	6.205	8.927	8.809	6.761	6.875	6.532	5.139	9.021	**9.512***	5.443%
		Pantry	3.965	6.347	7.673	4.530	4.729	4.326	4.623	6.937	**8.468***	10.361%
		Video	7.984	10.813	11.281	8.351	8.948	8.012	6.543	11.904	**12.990***	9.123%
	NDCG@20	Arts	4.102	4.911	5.572	4.383	4.907	4.303	4.357	6.310	**6.464***	2.441%
		Inst	5.026	8.478	9.470	5.745	5.991	6.271	6.941	9.553	**9.965***	4.312%
		Luxury	19.491	20.782	21.358	20.370	20.274	19.784	19.443	21.535	**21.974***	2.039%
		Music	3.241	4.844	4.824	3.521	3.754	3.633	2.562	4.866	**5.250***	7.891%
		Pantry	1.818	3.056	3.724	1.943	2.071	2.132	2.249	3.419	**4.354***	16.917%
		Video	3.909	5.363	6.209	4.294	4.440	3.186	4.192	6.110	**6.602***	6.330%
AliAd	Recall@20	d1	20.279	28.991	29.377	21.160	21.876	21.947	20.739	29.287	**30.600***	4.163%
		d2	34.772	42.291	40.395	35.540	36.164	34.502	37.571	40.762	**43.092***	1.894%
		d3	18.736	29.554	32.281	21.074	22.132	21.445	19.358	32.270	**34.754***	7.661%
		d4	30.783	38.026	41.173	30.954	31.357	34.378	35.238	41.258	**41.642***	0.931%
		d5	30.881	42.023	41.561	33.014	34.923	29.333	36.001	41.782	**43.133***	2.641%
		d6	34.077	44.155	45.524	35.801	35.397	34.185	30.979	45.511	**46.790***	2.781%
		d7	33.246	42.867	43.624	33.326	33.982	33.481	36.298	43.589	**44.987***	3.124%
		d8	29.641	36.572	36.748	30.062	30.860	27.255	31.626	36.991	**38.350***	3.674%
	NDCG@20	d1	9.250	15.079	15.871	10.253	10.419	10.671	9.742	15.788	**16.693***	5.809%
		d2	16.758	21.455	20.609	17.440	17.774	17.020	18.316	20.940	**22.596***	5.318%
		d3	9.333	16.137	17.125	11.805	11.801	11.050	8.473	17.058	**18.061***	5.466%
		d4	15.752	21.645	23.721	15.843	15.933	19.307	17.934	23.708	**24.185***	1.956%
		d5	15.709	23.147	22.514	17.122	17.840	15.767	19.089	22.657	**23.582***	1.879%
		d6	18.057	24.624	25.497	18.367	18.967	18.031	14.499	25.496	**26.106***	2.389%
		d7	16.892	24.054	24.279	17.375	17.661	18.178	18.762	24.251	**24.979***	2.883%
		d8	14.587	19.271	19.393	14.916	15.361	13.393	15.294	19.530	**20.814***	6.575%

* indicates significance level test $p < 0.05$ that compare UGDA over the best baseline.

disentanglement-based methods face parameter conflicts issues, leading to performance even inferior to single-domain models in some domains.

– On AliAd with fewer interactions, the performance of single-domain models surpasses that of all the other baseline models. This is because in scenarios with fewer interactions and sparser data, each domain alone cannot train robust representations. Cross-domain transfer becomes challenging, leading to negative transfer, as not all information is beneficial across domains. Consequently, the performance of CAT-ART, TrineCDR and EDDA may deteriorate due to negative transfer, even performing worse than single-domain models. However, UGDA still perform the best. This can be attributed to its direct construction of a large graph, which provides more robustness from a global perspective to counteract the impact of negative transfer.

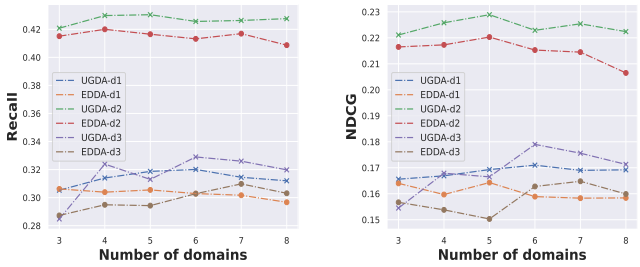
5.3 Ablation Study

To validate the effectiveness of each component of UGDA, we conducted ablation experiments and presented the results in Table 3. Due to the similarity of the experimental outcomes and space limitations, we only showcase the results for the Inst domain and the music domain. **w/o DA** means that domain-specific adaptation is not used, **w/o WL** means that weighted BPR loss is not used, and **w/o DA & WL** means that both of them are not used.

By comparing EDDA with w/o DA & WL, it can be found that using unified graphs to model global representations is more effective than summing subgraph representations of each domain in EDDA, because global unified graphs can di-

Table 3: Ablation Study

	Amazon-Inst		Amazon-Music	
	Recall@20	NDCG@20	Recall@20	NDCG@20
EDDA	14.867	9.707	8.862	4.850
w/o DA & WL	15.327	9.891	9.350	5.106
w/o DA	15.382	9.943	9.380	5.200
w/o WL	15.360	9.890	9.359	5.116
UGDA	**15.413**	**9.965**	**9.512**	**5.250**



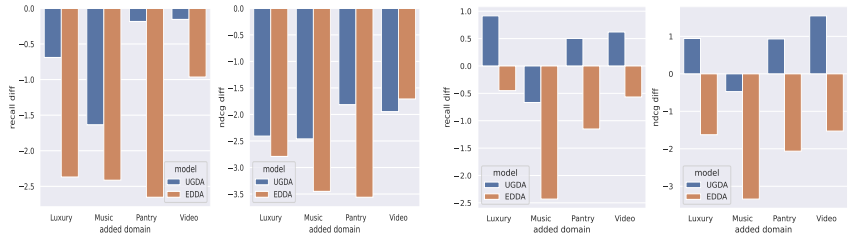
(a) (b)

Fig. 3: Performance with different domain number.

rectly capture the global graph structure, while the representation summation of each subgraph can only partially capture the global data structure. By comparing w/o DA and w/o WL with w/o DA & WL, we can see that adding both domain adaptation and weighted loss to the unified graph improves the model's performance, demonstrating the effectiveness of the domain-specific global representation and the weighted BPR loss. Furthermore, the full mode performs the best, demonstrating that the combination of all the modules is more effective.

5.4 Other Studies

Effect of domain number We evaluate UGDA's performance with different domain number, and the results are shown in Figure 3. By gradually increasing the domain number, we can find that the model performance gets improved at initial stage as more information is added. However, when the domain number exceeds a certain value, the model performance will decrease because of negative transfer, which implies not all the information from other domains is helpful for the current domain. For EDDA, we can observe that as the domain number increases, it is more severely affected by negative transfer. In certain domains, such as *d1*, the recall of EDDA even continuously decreases, indicating that it fails to learn sufficiently effective information to assist other domains. In most cases, adding more domains usually degrades performance, often worse than with



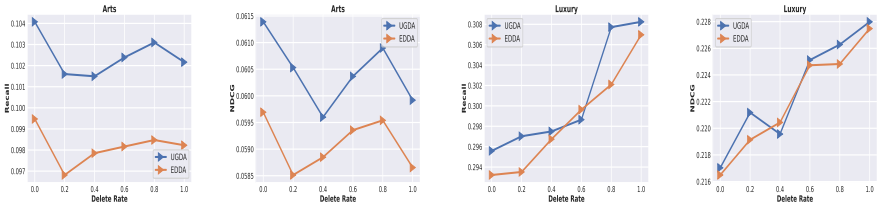
(a) Different Domain Effect on Arts (b) Different Domain Effect on Inst

Fig. 4: Performance with different domain

just three initial domains. In contrast, UGDA is more capable of overcoming the effects of negative transfer and continues to improve its performance as the number of domains increases, which can be attributed to the unified graph learning approach, which is able to learn more robust global representations, thereby resisting the effects of negative transfer.

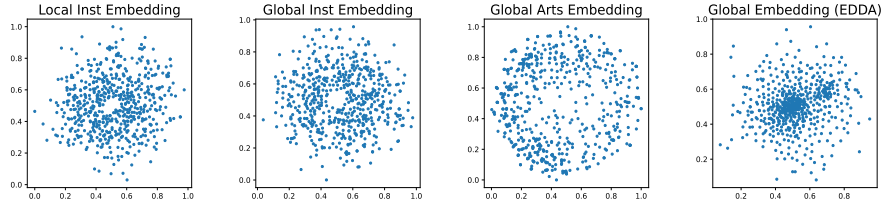
Effect of adding different domains We further conduct a study by incrementally adding different new domain x to the original <*Arts*, *Inst*> domain pair to form <*Arts*, *Inst*, *x*>. The results are presented in Figure 4. For *Arts*, the addition of a new domain consistently results in a performance decrease, indicating the presence of negative transfer. While both EDDA and UGDA experience a certain degree of performance drop, UGDA exhibits less degradation, showcasing greater robustness against adverse negative transfer effects. Notably, when *Pantry* is added, EDDA experiences a more significant performance drop compared to UGDA. For *Inst*, UGDA can even enhance performance when facing negative transfer effects across new domains such as *Luxury*, *Pantry*, and *Video*, suggesting that UGDA can effectively extract more valuable cross-domain information to improve performance. This effectiveness is attributed to our approach of modeling the system with a unified graph, facilitating the acquisition of robust global information.

Ratio of overlapping users. MDR relies on the existence of overlapping users, so we also evaluate the performance with different ratio of overlapping users. We choose <*Arts*, *Inst*> pair as base domains and add *Luxury* domain with different ratio of overlapping users. We first obtain the overlapping users between <*Arts*, *Inst*>, <*Arts*, *Luxury*>, and <*Inst*, *Luxury*>, and label them as u_{a-i}, u_{a-l}, and u_{i-l}, respectively. Then, we set $u_{a-l} = u_{a-l} - u_{a-i}$ and $u_{i-l} = u_{i-l} - u_{a-i}$. This is because we are only discussing the impact of overlapping users from the newly added domain. Next, following the specified deletion ratio, we sequentially remove the data of users from u_{a-l} and u_{i-l} according to the proportion and reinsert the remaining data into the dataset for training. The final results are shown in Figure 5, from which we can observe that, for the original *Arts* domain, the two



(a) (b) (c) (d)

Fig. 5: Performance with different deleting ratio of overlapping users



(a) (b) (c) (d)

Fig. 6: Visualization of the learned representations

models' performance generally decreases with the increase in removal ratio. This indicates that the presence of overlapping users can affect the transfer of cross-domain information. There is a slight improvement in model performance in the middle, which can be attribute to the removal of unnecessary information that does not need to be transferred. This improvement is particularly evident in the *Luxury* domain, where we can see that, as more users are removed, the model's performance improves. This demonstrates that the information from *Arts* and *Inst* has a negative transfer effect on *Luxury*, as *Luxury* itself is not sparse, and the transfer of information from other sparse domains affects its own information. Additionally, it can be observed that UGDA consistently outperforms EDDA in most cases, indicating that UGDA is more robust in transferring cross-domain information when faced with changes in overlapping user information.

5.5 Visualization

We use t-SNE[10] to visualize the local representations of UGDA in the *Inst* domain, the global representations in both *Arts* and *Inst* domains, and the global representations obtained by EDDA, as shown in Figure 6. We can see that the local representations in the *Inst* domain closely align with the distribution of global representations within the same domain. This underscores the effectiveness of our domain-specific globally structured graph in adapting to domain characteristics while preserving global information. Additionally, distinct distributions of global representations in the *Arts* and *Inst* domains indicate that rep-

resentations learned in globally structured graphs with domain-specific adaption carry unique domain-specific information. In visualizing global representations, EDDA's distributions concentrate in the central local region, whereas UGDA's representations are scattered across the entire space. This boils down to the fact that EDDA only uses simple addition to construct global representations, making them prone to being dominated by certain domains, resulting in the observed clustering phenomenon.

6 CONCLUSION

In this paper, we outline the shortcomings of previous MDR methods and propose our solution UGDA to overcome these limitations. We propose constructing a large unified graph to learn more comprehensive global structural information. Additionally, we introduce the idea of building domain-specific global graphs to ensure that global representations are tailored to each individual domain. Finally, we devise a novel BPR loss function that takes into account the consistency between users' global and local representations. Extensive experiments on two real-world datasets have demonstrated the effectiveness of UGDA, and further ablation studies have also verified the validity of each module of UGDA.

References

1. Cao, J., Li, S., Yu, B., Guo, X., Liu, T., Wang, B.: Towards universal cross-domain recommendation. In: WSDM. pp. 78–86 (2023)
2. Cao, J., Lin, X., Cong, X., Ya, J., Liu, T., Wang, B.: Disencdr: Learning disentangled representations for cross-domain recommendation. In: SIGIR. pp. 267–277 (2022)
3. Cui, Q., Wei, T., Zhang, Y., Zhang, Q.: Herograph: A heterogeneous graph framework for multi-target cross-domain recommendation. In: RecSys (2020)
4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. pp. 1126–1135. PMLR (2017)
5. Hao, X., Liu, Y., Xie, R., Ge, K., Tang, L., Zhang, X., Lin, L.: Adversarial feature translation for multi-domain recommendation. In: SIGKDD. pp. 2964–2973 (2021)
6. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR. pp. 639–648 (2020)
7. Hu, G., Zhang, Y., Yang, Q.: Conet: Collaborative cross networks for cross-domain recommendation. In: CIKM. pp. 667–676 (2018)
8. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991)
9. Kang, S., Hwang, J., Lee, D., Yu, H.: Semi-supervised learning for cross-domain recommendation to cold-start users. In: CIKM. pp. 1563–1572 (2019)
10. Kitazono, J., Grozavu, N., Rogovschi, N., Omori, T., Ozawa, S.: t-distributed stochastic neighbor embedding with inhomogeneous degrees of freedom. In: ICONIP. pp. 119–128. Springer (2016)
11. Li, C., Xie, Y., Yu, C., Hu, B., Li, Z., Shu, G., Qie, X., Niu, D.: One for all, all for one: Learning and transferring user embeddings for cross-domain recommendation. In: WSDM. pp. 366–374 (2023)

12. Li, X., Yan, F., Zhao, X., Wang, Y., Chen, B., Guo, H., Tang, R.: Hamur: Hyper adapter for multi-domain recommendation. In: CIKM. pp. 1268–1277 (2023)
13. Liu, M., Li, J., Li, G., Pan, P.: Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In: CIKM. pp. 885–894 (2020)
14. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: SIGKDD. pp. 1930–1939 (2018)
15. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: An embedding and mapping approach. In: IJCAI. vol. 17, pp. 2464–2470 (2017)
16. Ning, W., Yan, X., Liu, W., Cheng, R., Zhang, R., Tang, B.: Multi-domain recommendation with embedding disentangling and domain alignment. In: CIKM. pp. 1917–1927 (2023)
17. Niu, X., Li, B., Li, C., Tan, J., Xiao, R., Deng, H.: Heterogeneous graph augmented multi-scenario sharing recommendation with tree-guided expert networks. In: WSDM. pp. 1038–1046 (2021)
18. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
19. Salah, A., Tran, T.B., Lauw, H.: Towards source-aligned variational models for cross-domain recommendation. In: RecSys. pp. 176–186 (2021)
20. Sheng, X.R., Zhao, L., Zhou, G., Ding, X., Dai, B., Luo, Q., Yang, S., Lv, J., Zhang, C., Deng, H., et al.: One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In: CIKM. pp. 4104–4113 (2021)
21. Song, Z., Zhang, W., Deng, L., Zhang, J., Wu, Z., Bian, K., Cui, B.: Mitigating negative transfer in cross-domain recommendation via knowledge transferability enhancement. In: SIGKDD. pp. 2745–2754 (2024)
22. Wang, B., Zhang, C., Zhang, H., Lyu, X., Tang, Z.: Dual autoencoder network with swap reconstruction for cold-start recommendation. In: CIKM. pp. 2249–2252 (2020)
23. Zang, T., Zhu, Y., Liu, H., Zhang, R., Yu, J.: A survey on cross-domain recommendation: taxonomies, methods, and future directions. ACM Transactions on Information Systems **41**(2), 1–39 (2022)
24. Zhang, F., Peng, Q., Wu, Y., Pan, Z., Zeng, R., Lin, D., Qi, Y.: Multi-graph based multi-scenario recommendation in large-scale online video services. In: WWW. pp. 1167–1175 (2022)
25. Zhang, R., Zang, T., Zhu, Y., Wang, C., Wang, K., Yu, J.: Disentangled contrastive learning for cross-domain recommendation. In: DASFAA. pp. 163–178 (2023)
26. Zhang, Y., Li, X., Yu, Y., Tang, J., Deng, H., Lu, J., Zhang, Y., Jiang, Q., Xian, Y., Yu, L., et al.: Meta-generator enhanced multi-domain recommendation. In: WWW. pp. 485–489 (2023)
27. Zhao, C., Li, C., Fu, C.: Cross-domain recommendation via preference propagation graphnet. In: CIKM. pp. 2165–2168 (2019)
28. Zhu, F., Wang, Y., Chen, C., Liu, G., Orgun, M., Wu, J.: A deep framework for cross-domain and cross-system recommendations. In: IJCAI. pp. 3711–3717 (2018)
29. Zhu, F., Wang, Y., Chen, C., Zhou, J., Li, L., Liu, G.: Cross-domain recommendation: challenges, progress, and prospects. In: IJCAI. pp. 4721–4728 (2021)
30. Zhu, J., Wang, Y., Zhu, F., Sun, Z.: Domain disentanglement with interpolative data augmentation for dual-target cross-domain recommendation. In: RecSys. pp. 515–527 (2023)
31. Zhu, Y., Ge, K., Zhuang, F., Xie, R., Xi, D., Zhang, X., Lin, L., He, Q.: Transfer-meta framework for cross-domain recommendation to cold-start users. In: SIGIR. pp. 1813–1817 (2021)