Debiasing Graph Transfer Learning via Item Semantic Clustering for Cross-Domain Recommendations

Zhi Li
Osaka University
li.zhi@ist.osaka-u.ac.jp

Daichi Amagata
Osaka University
amagata.daichi@ist.osaka-u.ac.jp

Yihong Zhang
Osaka University
yhzhang7@gmail.com

Takahiro Hara
Osaka University
hara@ist.osaka-u.ac.jp

Shuichiro Haruta
KDDI Research, Inc.
sh-haruta@kddi-research.jp

Kei Yonekawa
KDDI Research, Inc.
ke-yonekawa@kddi-research.jp

Mori Kurokawa
KDDI Research, Inc.
mo-kurokawa@kddi-research.jp

Abstract—Deep learning-based recommender systems may lead to over-fitting when lacking training interaction data. This over-fitting significantly degrades recommendation performances. To address this data sparsity problem, cross-domain recommender systems (CDRSs) exploit the data from an auxiliary source domain to facilitate the recommendation on the sparse target domain. Most existing CDRSs rely on overlapping users or items to connect domains and transfer knowledge. However, matching users is an arduous task and may involve privacy issues when data comes from different companies, resulting in a limited application for the above CDRSs. Some studies develop CDRSs that require no overlapping users and items by transferring learned user interaction patterns. However, they ignore the bias in user interaction patterns between domains and hence suffer from an inferior performance compared with single-domain recommender systems. In this paper, based on the above findings, we propose a novel CDRS, namely semantic clustering enhanced debiasing graph neural recommender system (SCDGN), that requires no overlapping users and items and can handle the domain bias. More precisely, SCDGN semantically clusters items from both domains and constructs a cross-domain bipartite graph generated from item clusters and users. Then, the knowledge is transferred via this cross-domain user-cluster graph from source to the target. Furthermore, we design a debiasing graph convolutional layer for SCDGN to extract unbiased structural knowledge from the cross-domain user-cluster graph. Our Experimental results on three public datasets and a pair of proprietary datasets verify the effectiveness of SCDGN over state-of-the-art models in terms of cross-domain recommendations.

Index Terms—recommender system, cross-domain recommendations, graph convolutional network, debiasing learning

I. INTRODUCTION

Recommender systems (RSs) predict users’ future item-interacted behaviors, e.g., purchasing in e-commerce and rating in the case of online movies, where the predictions are inferred by models learned from their past interaction behaviors [1], [2]. Deep learning has been employed in RSs, with its well-designed structures and the large number of learnable parameters, to better model users’ complex interaction patterns. In real-world services, most users interact with only a few items, particularly in start-up companies and when companies develop new services. In such scenarios, deep learning-based RSs may lead to over-fitting because of the sparse interaction data [3], [4], which significantly degrades the recommendation performance.

Motivation. To address the aforementioned data sparsity problem, cross-domain recommender systems (CDRSs) have been developed. CDRSs exploit the data from an auxiliary domain (i.e., a source domain) to facilitate the inference process in a target domain. CDRSs are categorized into two approaches: multi-task learning and transfer learning [5], [6]. In multi-task learning-based CDRSs, some neural layers or user (item) embedding are shared between domains [5]. These shared layers or embedding are optimized by fitting the recommendation tasks to both source and target domains, and hence they can learn knowledge from both domains to provide accurate recommendations compared to single-domain recommendations. In contrast, transfer learning-based CDRSs focus on recommendations in the target domain [6], [7]. These CDRSs extract knowledge from the source domain and use the learned knowledge to improve the target recommendations.

Most existing CDRSs, including the above ones, bridge domains by matching user information and transferring knowledge across domains via overlapping users. These overlapping users are, however, not always available in real-world services. In addition, user matching is an arduous task and may involve privacy issues, particularly when data come from different companies. Some CDRSs, such as RecSysDAN [8] and CFAA [7], therefore extract user interaction patterns from the source domain and transfer the learned knowledge to the target domain. However, these methods ignore the bias in user interaction patterns between domains and hence suffer from an inferior performance compared with single-domain recommenders.

To address the above findings, we propose a novel CDRS, namely semantic clustering enhanced debiasing graph neural recommender system (SCDGN), that requires no overlapping users and items and can handle the domain bias. More precisely, SCDGN semantically clusters items from both domains and constructs a cross-domain bipartite graph generated from item clusters and users. Then, the knowledge is transferred via this cross-domain user-cluster graph from source to the target. Furthermore, we design a debiasing graph convolutional layer for SCDGN to extract unbiased structural knowledge from the cross-domain user-cluster graph. Our Experimental results on three public datasets and a pair of proprietary datasets verify the effectiveness of SCDGN over state-of-the-art models in terms of cross-domain recommendations.
interaction patterns to the target domain, where the interaction patterns are defined as learned user (item) embedding or the distribution of predictions. They define domains as different categories in a Web service, such as Amazon Book and Amazon Movie, or different places in a real-world service, such as the restaurant visit records in different cities. By doing so, they can merge the source and target domains with domain-shared side information, such as user profiles and item contents. Unfortunately, user interaction patterns have been observed to be strongly domain-dependent, particularly when these domains come from different services [9]. These CDRSs hence suffer from the domain bias in user interaction patterns.

**Contribution.** From the above observations, it can be seen that a technique which demands no user matching and can alleviate the bias of interaction patterns is required. Motivated by this, we develop a new CDRS, namely semantic clustering enhanced debiasing graph neural recommender system (SCDGN), that is applicable to different services sharing no entities, including users and items. To achieve this, we generate a cross-domain user-cluster graph to bridge two domains, where the graph consists of users and item clusters. Since item clusters are generated from textual information on items (e.g., movie titles and web page contents), user-cluster graph can merge the semantic interaction information of different domains and user matching becomes unnecessary. Then, we extract knowledge from both target item-level and cross-domain cluster-level interaction graphs by devising a CDRS variant of LightGCN [10]. This variant is inspired by the success of LightGCN in extracting complex high-hop neighbor information from graph structures. To handle the bias of user interaction patterns in different domains, we develop a novel debiasing graph convolutional layer to learn unbiased knowledge from the cross-domain user-cluster graph. More precisely, we design adaptive debiasing vectors for users and item clusters to weight edges in the cross-domain user-cluster graph. Moreover, inspired by the effectiveness of debiasing learning [9], we develop two-level restrictions to learn the above debiasing vectors.

To summarize, our contributions are as follows:

- We propose a novel semantic cluster-based domain merge approach to make the interaction information transferable at an item-cluster level. By doing so, Our CDRS does not require user matching.
- We develop a debiasing cluster-enhanced cross-domain graph convolutional model to transfer knowledge and alleviate the bias in interaction patterns between domains.
- Our experimental results on three public datasets and two private datasets demonstrate that our proposal significantly outperforms state-of-the-art methods.

A full version of this paper is available at [11].

**II. RELATED WORK**

**Cross-domain Recommender Systems.** To mitigate the data sparsity problem, CDRSs leverage data from an auxiliary source domain to facilitate recommendations in the sparse target domain. Some existing CDRSs require overlapping users to bridge domains and to transfer individual-level knowledge [5], [12], [13]. However, matching users is arduous and may involve privacy issues in most real-world applications. Considering privacy and the scalability of methods, some studies avoid user alignment and transfer distribution-level knowledge from the source domain to the target domain. For example, ESAM [14] and CFAA [7] aligned the attribution distribution and correlation between source and target embedding spaces to transfer knowledge. RecSys-DAN [8] proposed a novel discriminator and minimized the divergence of the predictions between the source domain and the target domain for knowledge transformation. Unfortunately, bias in interaction patterns between domains may degrade the recommendation performances of CDRSs. The methods mentioned above do not consider this domain bias issue.

**Graph Convolution in Recommendations.** Recently, graph neural networks (GNNs) have been employed in RSs to guide the embedding learning by exploiting user-item graph structures [10], [15], [16]. NGCF [15] defined the information propagation as aggregation of the embeddings of neighbors to enhance the target node’s (i.e., users’ or items’) embedding. Considering that recommender systems often use one-hot embedding (i.e., less information than images and text), LightGCN [10] further simplified and customized graph models to avoid over-fitting. In addition, some GNN-based CDRSs also alleviate the sparse problem by combining the complex high-order graph structural information from the source to the target [6], [16]. However, the above-mentioned GNN-based CDRSs require overlapping users to connect domains and ignore the domain bias in user preferences patterns. BitGCF [6] developed a domain-specific feature propagation layer to handle the domain bias, but it still requires overlapping users to fuse domain information.

**III. METHOD**

**A. Problem Formulation**

In this work, we define the top-K recommendations in a sparse domain as our recommendation task. We consider an auxiliary source domain $D_s$ and a sparse target domain $D_t$. $D_s$ contains $U_t$, $V_t$, and $R_t$, where $U_t$ ($V_t$) denotes the user (item) set, and $R_t$ is the interaction set between them. Similarly, $D_s$ contains $U_s$, $V_s$, and $R_s$. There is no overlap between the user and item sets of $D_s$ and $D_t$.

To address the data sparsity problem, we consider the semantic clustering information of items $C$ extracted from both the source and target domains, because this enhances the sparse interactions in the target domain. As a result, each interaction $r \in R_t$ is a tuple $r = (u, v, c)$, where $u \in U_t$, $v \in V_t$, and $c \in C$. For each user $u \in U_t$, we predict a preference score $\hat{y}_{u,v}$ for each item $v \in V_u = \{v \in V_t, v \notin V_u\}$, where $V_u$ is the items that interacted with $u$. We then rank the items in $V_u$ according to their preference scores and recommend the top-K items with the largest scores to $u$.  

763
1. Item Embedding

- Source domain item
- Target domain item

Text corpus

BERT

Item embedding space (semantic feature)

2. Semantic Clustering

- k-means

3. Graph Constructing

- Source domain
- Target domain

User-cluster graph \(G_{cross}\)

4. Predicting

msg. propagation on \(G_{target}\)

5. Recommending

Prediction model

Recommendation list

0.7

0.5

0.4

0.2

Fig. 1. Overview of our proposed CDRS. (1) The item embedding learns semantic embeddings for all items in source and target domains. (2) All items in the source and target domains are then clustered by their semantic embeddings. (3) A cross-domain user-cluster graph is constructed to merge the two domains’ interaction information at a semantic cluster level. (4) A debiasing graph convolutional neural network makes predictions by leveraging the interaction information from the target user-item and cross-domain user-cluster graphs. (5) Finally, recommendations are produced by the prediction results.

### TABLE I

| Notation | Description |
|----------|-------------|
| \(u\)    | a user      |
| \(v\)    | an item     |
| \(c\)    | a cluster (a set of items) |
| \(G_{cross}\) | cross-domain user-cluster graph |
| \(G_{target}\) | target user-item graph |
| \(N_u\)  | neighbor cluster set of \(u\) in \(G_{cross}\) |
| \(N_c\)  | neighbor user set of \(c\) in \(G_{cross}\) |
| \(M_u\)  | neighbor item set of \(u\) in \(G_{target}\) |
| \(M_c\)  | neighbor user set of \(v\) in \(G_{target}\) |
| \(v_{txt}, e_{txt}\) | semantic vectors of \(v\) and \(c\) |
| \(e_u, e_v, e_c\) | embedding of \(u\), \(v\), and \(c\) |
| \(a_u, a_c\) | debiasing vectors of \(u\) and \(c\) |
| \(\vec{e}_u, \vec{e}_v, \vec{e}_c\) | unbiased finet embedding of \(u\), \(v\), and \(c\) |
| \(\vec{e}_u, \vec{e}_v, \vec{e}_c\) | biased finet embedding of \(u\), \(v\), and \(c\) |
| \(h_u^{(l)}, h_v^{(l)}\) | \(G_{target}\)’s \(l\)-th graph conv. layer outputs |
| \(h_c^{(l)}\) | \(G_{cross}\)’s \(l\)-th debiasing graph conv. layer outputs |
| \(g_u^{(l)}, g_v^{(l)}\) | \(G_{cross}\)’s \(l\)-th graph conv. layer outputs |

### C. Semantic Domain Fusing

To semantically fuse the source and target domains, we first embed all items in the source and target domains into a domain-shared embedding space. Then, we cluster items based on this embedding space and construct a cross-domain user-cluster graph to enhance the interaction information.

1) **Semantic item embedding**: Given textual information on items, such as a description of a product, we extract semantic features from the text information to represent items in the source and target domains. We apply the token embeddings from a pre-trained BERT [17] to represent tokens in item text, because this model is learned by sufficient Wikipedia data and hence contains semantic information. The text of item \(v\) is denoted by \(text(v)\), and an semantic embedding of item \(v\) is obtained by

\[
v_{txt} = \sum_{w \in text(v)} \phi_{tf-idf}(w) \cdot \phi_{BERT}(w),
\]

where \(\phi_{BERT}(w)\) and \(\phi_{tf-idf}(w)\) are respectively the embedding and the tf-idf score of token \(w \in text(v)\). Note that
\( \phi_{tf-idf}(w) \) is calculated based on the text corpus collected from both domains.

2) User-cluster graph construction: We next construct a user-cluster graph. This aims at merging the source and target domains without user, item, and side information alignments. In addition, high-hop neighbors in this user-cluster graph can yield useful knowledge to improve the recommendation accuracy [10]. In Section III-D, we leverage this observation through modeling such structures from this graph, which also motivates building this user-cluster graph.

To construct the user-cluster graph, we first run the semantic clustering in Figure 1, that is, we cluster all items from the source and target domains in the learned semantic embedding space. We employ the empirically effective k-means [19] method and leave the discussion of more clustering methods as a future work. After that, we construct the cross-domain user-cluster graph \( G_{cross} \) to merge the two domains’ semantic-level interaction information, where \( \mathcal{U} = \{ U_s, U_t \} \) and \( \mathcal{C} \) respectively denote the user and cluster sets. The link \( r_{u,c} = 1 \) indicates that there is an interaction between \( u \) and any item belonging to \( c \); otherwise \( r_{u,c} = 0 \).

D. Debiasing Graph Convolutional Predictor

We here develop a cluster-enhanced debiasing graph convolutional model for recommendations in the sparse target domain. Different from existing CDRSs that transfer item interaction patterns directly, our model transfers the semantic clustering interaction patterns via the cross-domain user-cluster graph \( G_{cross} \). To achieve this, our model fuses \( G_{cross} \) and the target user-item graph \( G_{target} \) to refines the user and item embeddings with structural knowledge from graphs, where \( G_{target} = \{(u, r_{u,v}, v) | u \in \mathcal{U}, v \in \mathcal{V}_t, r_{u,v} \in \mathcal{R}_t \} \). This model consists of three main components: (i) an embedding layer, which learns latent vectors for users and items, (ii) debiasing graph convolutional layers, which recursively propagate unbiased high-hop neighbor information to refine the user and item vectors, and (iii) a prediction layer, which aggregates the user and item representations from all propagation layers and outputs the predictions.

1) Embedding layer: To alleviate the data sparsity problem, we propose a novel approach that projects users into the item embedding space learned in Section III-C. Furthermore, we design a metric-invariant dimension reduction approach to control the scale of parameters according to the difficulty of the recommendation task and the sparsity of the training data. The item embedding is calculated by a dimension compression layer: \( e_v = W_{v_{ext}} + b \), where \( W \) and \( b \) are the parameters of this layer. The dimension of \( e_v \) is much smaller than that of \( v_{ext} \), in order to adapt to the sparse target domain. The cluster embedding is computed by the same layer: \( e_c = W_{c_{ext}} + b \), where \( c_{ext} \) is the semantic embedding of cluster \( c \). \( c_{ext} \) is defined as the mean pooling of all item semantic embeddings in this cluster and formulated by

\[
e_{c_{ext}} = \frac{1}{|\mathcal{V}_c|} \sum_{v \in \mathcal{V}_c} v_{ext},
\]

where \( \mathcal{V}_c \) is the item set in cluster \( c \). The user embedding is defined as the ID embedding \( e_u \), which has the same dimension as that of \( e_v \). We measure the cosine similarities \( S_r \) between items and clusters and minimize the mean squared error of the cosine similarities calculated before and after dimension reduction to ensure the metric invariance, where the error is defined as

\[
L_{dr} = \frac{1}{|\mathcal{R}_t|} \sum_{(u,v,c) \in \mathcal{R}_t} \left( (S_c(e_v, e_{v^-}) - S_c(v_{ext}, v_{ext}^-))^2 + (S_u(e_c, e_{c^-}) - S_c(v_{ext}, c_{ext}^-))^2 \right) .
\]

In this equation, \( v^- \) is a negative item randomly sampled from \( \mathcal{V}_u \) and \( c^- \) is the cluster to which \( v^- \) belongs. This approach adjusts the embedding dimension and maintains a consistent spatial relationship with the original embedding space.

2) Debiasing graph convolutional layers: Because of the superiority of graph convolutional networks in capturing and modeling structural information from graphs, we develop graph convolutional modules for extracting structural information from the target user-item graph \( G_{target} \) and the cross-domain user-cluster graph \( G_{cross} \). More precisely, we employ the state-of-the-art “light graph convolution” layer [10] to propagate graph information because of its effectiveness in alleviating overfitting for our sparse target domain. To identify the domain bias in user preference patterns and extract unbiased knowledge from \( G_{cross} \), we propose a novel debiasing graph convolutional layer. For each user \( u \in \mathcal{U} \), we set an adaptive debiasing vector \( a_u \) to represent her individual domain bias. For each cluster \( c \in \mathcal{C} \), we also set an adaptive debiasing vector \( a_c \). By doing so, the debiasing factor of user-cluster interaction \( r_{u,c} \) can be defined as: \( a_{uc} = a_u^T a_c \). The \( l \)-th debiasing graph convolutional layer for \( G_{cross} \) is formulated as:

\[
\begin{align*}
g_{u}^{(l+1)} &= \sum_{c \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|} \sqrt{|\mathcal{C}|}} a_{uc} \cdot g_{c}^{(l)}, \\
g_{c}^{(l+1)} &= \sum_{u \in \mathcal{N}_c} \frac{1}{\sqrt{|\mathcal{N}_c|} \sqrt{|\mathcal{U}|}} a_{uc} \cdot g_{u}^{(l)},
\end{align*}
\]

where \( \mathcal{N}_u = \{ c | r_{u,c} = 1, r_{u,v} \in \mathcal{R}_{target} \} \) is the neighbor cluster set of user \( u \) and \( \mathcal{N}_c \) is the neighbor user set of cluster \( c \). We define \( g_u^{(0)} = e_u \) and \( g_c^{(0)} = e_c \). It is worth mentioning that we detach the gradient computation of the debiasing vectors \( a_u \) and \( a_c \) here for computational efficiency. The learning of \( a_u \) and \( a_c \) is left to the proposed restrictions in Section III-E. For \( G_{target} \), we adopt the standard “light graph convolution” layer [10], where the \( l \)-th graph convolutional layer is formulated as:

\[
\begin{align*}
h_{u}^{(l+1)} &= \sum_{v \in \mathcal{M}_u} \frac{1}{\sqrt{|\mathcal{M}_u|} \sqrt{|\mathcal{V}|}} h_{v}^{(l)}, \\
h_{v}^{(l+1)} &= \sum_{u \in \mathcal{M}_v} \frac{1}{\sqrt{|\mathcal{M}_v|} \sqrt{|\mathcal{U}|}} h_{u}^{(l)},
\end{align*}
\]

where \( \mathcal{M}_u = \{ v | r_{u,v} = 1, r_{u,v} \in \mathcal{R}_{target} \} \) is the neighbor item set of user \( u \) and \( \mathcal{M}_v \) is the neighbor user set of item \( v \). Similarly, we define \( h_u^{(0)} = e_u \) and \( h_v^{(0)} = e_v \).
3) Prediction layer: We next refine $e_u$, $e_v$, and $e_c$ by using the extracted graph structure information. The final representation is produced by aggregating the embeddings obtained at each graph convolutional layer:

$$
\bar{e}_u = \sum_{l=0}^{P} g_u^{(l)} + \sum_{l=0}^{Q} h_u^{(l)},
$$

$$
\bar{e}_v = \sum_{l=0}^{Q} h_v^{(l)};
\bar{e}_c = \sum_{l=0}^{P} g_c^{(l)},
$$

where $P$ and $Q$ are the numbers of debiasing graph convolutional layers for $G_{cross}$ and graph convolutional layers for $G_{target}$, respectively. It is worth mentioning that $e_u$ is refined by the structural information from both the target and cross-domain graphs, i.e., the knowledge from both the item and cluster level interactions.

Finally, the preference score is defined as the inner product of the user and item final representation:

$$
y_{uv} = \bar{e}_u^T \bar{e}_v
$$

E. Restrictions for Debiasing Learning

The previous debiasing learning [9] designs their restrictions via overlapping users and domain-shared item attributes, e.g., category, seller, brand, and price, resulting in a limited application. Besides, it directly sets adaptive debiasing factors for each user-item interaction and optimizes them separately. In other words, the learning of a debiasing factor only relies on the corresponding interaction and thus suffers from a severe overfitting issue. Based on these findings, we get hints from the matrix factorization algorithm and re-define the debiasing factor $a_{uc}$ as the inner product of the corresponding user debiasing vector $a_u$ and cluster debiasing vector $a_c$. By doing so, the number of debiasing parameters is reduced from $|U| \times |C|$ to $|U| + |C|$, which alleviates the overfitting issue. Our approach learns $a_u$ and $a_c$ via the restriction losses at both prediction and individual levels.

1) Restriction in prediction level: As a debiasing factor, $a_{uc}$ is demanded to produce unbiased prediction $\hat{y}_{uc}$ from the biased version $\hat{y}_{uc}$. Formally, we define the debiasing operation as $a_{uc} \cdot \hat{y}_{uc}$. Therefore, we devise a restriction loss $L_{rsp}$ that measures the mean squared error between $\hat{y}_{uc}$ and $a_{uc} \cdot \hat{y}_{uc}$.

$$
L_{rsp} = \frac{1}{|R_s|} \sum_{(u,v,c) \in R_s} (\hat{y}_{uc} - a_{uc} \cdot \hat{y}_{uc})^2,
$$

where $\hat{y}_{uc} = \bar{e}_u^T \bar{e}_c$. Here, $\bar{e}_u$ and $\bar{e}_c$ denote the biased user embedding and the biased cluster embedding, respectively, and are formulated by the aggregating outputs of every graph convolutional layers:

$$
\bar{e}_u = \sum_{l=0}^{P} g_u^{(l)};
\bar{e}_c = \sum_{l=0}^{P} g_c^{(l)},
$$

where $g_u^{(l)}$ and $g_c^{(l)}$ are, respectively, the user and cluster aggregation result of the $l$-th graph convolutional layer that can be computed by Equation 4 without the debasing factor. By minimizing $L_{rsp}$, we can ensure a consistent result between the prediction debiased by the debiasing factor and the prediction produced by the debiasing graph convolutional layers. As a result, $L_{rsp}$ constrains the learning space of $a_u$ and $a_c$ and hence can alleviate overfitting.

2) Restriction in individual level: The individual level, $a_u$ and $a_c$, are required to generate unbiased $\bar{e}_u$ and $\bar{e}_c$ from the biased $\bar{e}_u'$ and $\bar{e}_c'$, respectively. To meet this requirement, we introduce a user restriction loss $L_{rsu}$ and a cluster restriction loss $L_{rsc}$. $L_{rsu}$ measures the Euclidean distance between $\bar{e}_u$ and $a_u \circ \bar{e}_u'$, where $\circ$ is Hadamard product. Analogously, $L_{rsc}$ measures the Euclidean distance between $\bar{e}_c$ and $a_c \circ \bar{e}_c'$.

$$
L_{rsu} = \frac{1}{|U|} \sum_{u \in U} \| \bar{e}_u - a_u \circ \bar{e}_u' \|_2^2,
$$

$$
L_{rsc} = \frac{1}{|C|} \sum_{c \in C} \| \bar{e}_c - a_c \circ \bar{e}_c' \|_2^2.
$$

Taking $L_{rsu}$ as an example, minimizing $L_{rsu}$ forces $a_u$ to mitigate the domain bias for user $u$, i.e., this user’s all interactions are leveraged to learn the debiasing vector $a_u$. $L_{rsu}$ hence provides a stronger constrain compared to the existing method [9] and can alleviate the overfitting issue, as does $L_{rsc}$.

F. Model Optimization

We adopt the Bayesian Personalized Ranking (BPR) loss [10] to measure the loss of predicted preference scores. The BPR loss is obtained by:

$$
L_{bpr} = - \sum_{(u,v,c) \in R_s} \ln \sigma (y_{uv} - y_{uv^-}),
$$

where $v^-$ is a negative item randomly sampled from $\hat{Y}_u$.

The total loss is measured by combining the dimension reduction loss $L_{dr}$, the restriction loss $L_{rs}$, and the BPR loss $L_{bpr}$, that is,

$$
L = L_{bpr} + \lambda_1 L_{rs} + \lambda_2 L_{dr} + \lambda_3 ||\theta||^2,
$$

where $L_{rs} = L_{rsp} + L_{rsu} + L_{rsc}$. $\lambda_1$, $\lambda_2$, and $\lambda_3$ are hyperparameters used to balance the weight between losses. We employ a gradient descent algorithm to optimize our model by minimizing $L$.

IV. EXPERIMENTS

A. Experiment Setting

Dataset. We conducted experiments on two proprietary datasets and three widely used public datasets to investigate the recommendation performance of SCDGN in practical applications and for benchmarking purposes.

The public datasets contain a subset of MovieLens25M\(^1\) and two subsets of Amazon\(^2\). The subset of MovieLens25M (ML) contains movie ratings from 30/9/2016 to 1/10/2018,}

\(^1\)grouplens.org/datasets/movielens/25m/

\(^2\)jmcauley.ucsd.edu/data/amazon/
where the movie descriptions in ML were collected from the public API of TMDB\(^3\). The two subsets of Amazon include an AmazonBook (AB) dataset and an AmazonMovie (AM) dataset. AB and AM contain book and movie ratings from 30/9/2016 to 3/10/2018, respectively, as well as textual descriptions of the books and movies.

The private datasets have an online advertisement dataset (ADs) [20] and an e-commerce dataset (E-com). ADs contains web browsing records from 1/8/2017 to 31/8/2017 on an ads platform and the textual content of Web pages. E-com provides purchase records from an e-commerce platform and the textual descriptions of products, where the purchase records in E-com have the same period as that of ADs.

We measured three cross-domain recommendation tasks, where each recommendation task contains an auxiliary source domain and a relatively sparse target domain. We defined A→B as a cross-domain recommendation task, where A is the source domain, and B is the target domain. The recommendation tasks include: (1) ADs→E-com and com→ADs; (2) ML→AM and AM→ML; (3) ML→AB and AB→ML. For each source domain, we selected users who have 3 to 10 interaction records and items that have 10 to 15 interaction records to fit a dense setting. Inversely, for each target domain, we selected users who have 3 to 5 interactions and items that have 5 to 15 interactions to form a relatively sparse environment. Some basic information about the pre-processed datasets is summarized in Table II.

| Dataset | Users | Items | Interactions | Int./U |
|---------|-------|-------|--------------|-------|
| AB      | 13,350| 10,477| 61,004       | 4.57  |
| AM      | 22,046| 7,814 | 104,216      | 4.73  |
| ML      | 18,232| 14,435| 421,803      | 4.99  |
| E-com   | 17,418| 6,142 | 81,499       | 4.68  |

### Evaluation criteria.
For each user in target domains, we took this user’s last and second-last interactions to form the test and validation sets, respectively. The remaining interactions were used as the training set. Then, we randomly sampled 99 items that had no interaction with this user and ranked the target item among the 100 items. The result for the top-K recommendations was measured by the widely used Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG).

### Evaluated methods.
To measure the validity of the semantic information coming from the source data, we compared our method with the following state-of-the-art methods, including (1) single-domain methods: NeuCF [21] and LightGCN [10], (2) cross-domain methods: s\(^2\)-Meta [22], RecSys-DAN [8], ESAM [14], and CFAA [7].

For fair comparisons, we aligned the base model for all cross-domain methods with LightGCN, and this base model is equal to our SCDGN without the cross-domain user-cluster graph part. Moreover, we replaced the randomly initialized item embedding with our pre-trained semantic item embedding in Section III-C for all cross-domain comparisons and LightGCN, and these methods are identified with (wt).

### Implementation details.
The codes of NeuCF\(^4\), LightGCN\(^5\), and s\(^2\)-Meta\(^6\) were obtained from the corresponding GitHub repositories. Our SCDGB, ESAM, and CFAA were implemented using PyTorch framework and can be found in a GitHub repository\(^7\).

We used Adam to optimize the model parameters and speed up the training process with the mini-batch trick. For hyper-parameters, the learning rate was 0.001 for the recommendation tasks on private datasets and 0.01 for the cases on public datasets. The cluster number was 200. The embedding size of \(e_u\) was 32. The mini-batch size was 1024. The restriction loss balance factor \(\lambda_1\) was set to 1, 0.001, and 0.0001 for the recommendation task on ML→AM, E-com→ADs, and ADs→E-com, respectively. \(\lambda_1\) was set to 0.01 for the recommendation task on AM→ML, ML→AB, and AB→ML. The dimension reduction loss balance factor \(\lambda_2\) was set to 1 for the recommendation task on private datasets, ML→AM, and ML→AB, where it was set to 10 for the recommendation on AM→ML and AB→ML. The weight of the regularization term \(\lambda_3\) was set to 0.01 for the recommendation task on private datasets and 0.1 for the case on public datasets. The user-cluster graph convolutional layer number \(P\) was set to 2 for recommendations on public datasets and 1 for the private datasets. For fair comparisons, we set the same user-item graph convolutional layer number \(Q\) as 3 for all comparisons except NeuCF. All these hyper-parameters were tuned on the validation set.

### B. Performance Comparison.
We report the average recommendation performances on the test set of each target domain. The comparison results are listed in Table III. This table shows that SCDGN outperforms other competitors on HR@5, NDCG@5, and HR@1 (in most cases). Besides, SCDGN achieves a remarkable improvement on four public recommendation tasks. This observation empirically demonstrates that our SCDGN effectively leverages the semantic information on the source domains to improve the recommendations in the target domains. For single-domain RSs, we find that LightGCN (wt) achieves the best performance. LightGCN the second best, and NeuCF the worst. This is because the target semantic information and the graph convolutional network yield a better performance. For cross-domain RSs, although they transfer interaction patterns or

---

\(^3\)www.themoviedb.org/documentation/api

\(^4\)github.com/yihong-chen/neural-collaborative-filtering

\(^5\)github.com/gusye1234/LightGCN-PyTorch

\(^6\)github.com/THUDM/ScenarioMeta

\(^7\)github.com/ZL6298/SCDGN
align embedding space from the source domain to the target domain, they perform worse than the single-domain method, i.e., LightGCN, in most cases. This result indicates that domain bias in interaction patterns causes the negative transfer issue and an inferior performance.

C. Vs. CDRS with Overlapping Users

To further investigate the effectiveness of the proposed method, we identified overlapping users between AM and AB and conducted experiments to compare our SCDGN with the state-of-the-art CDRSs that require overlapping users, i.e., CGN [12] and BiTGCF [6]. Some basic information of the datasets used in this experiment is summarized in Table IV.

Table V shows the comparison results on HR@1 and HR@5. We observe that our SCDGN remarkably outperforms the state-of-the-art CDRSs that require overlapping users, i.e., LightGCN, in most cases. This result indicates that domain bias in interaction patterns causes the negative transfer issue and thus outperform CGN. In addition, it is worth mentioning that SCDGN involves no user assignment, suggesting that SCDGN has a broader application than CGN and BiTGCF.

---

TABLE III
Comparison between our proposal and state-of-the-art by using HR@K and NDCG@K. Performances ± 95% confidence intervals are reported. Bold shows the winner.

| Method       | HR@1    | NDCG@5 | HR@1    | NDCG@5 |
|--------------|---------|--------|---------|--------|
| NeuCF        | 0.323 ± 0.024 | 0.442 ± 0.025 | 0.348 ± 0.022 |
| LightGCN     | 0.368 ± 0.005 | 0.435 ± 0.005 | 0.384 ± 0.006 |
| LightGCN (wt)| 0.383 ± 0.007 | 0.466 ± 0.005 | 0.400 ± 0.008 |
| BiTGCF       | 0.022 ± 0.001 | 0.169 ± 0.006 | 0.194 ± 0.008 |
| CGN          | 0.144 ± 0.009 | 0.274 ± 0.015 | 0.200 ± 0.017 |
| SCDGN (ours) | 0.321 ± 0.006  | 0.435 ± 0.006  | 0.367 ± 0.008  |

TABLE IV
Basic information on the datasets with only the overlapping users. #Int./U is the average number of interactions per user.

| Dataset | #Users | #Items | #Interactions | #Int./U |
|---------|--------|--------|---------------|--------|
| AM      | 1,315  | 5,488  | 15,169        | 11.54  |
| AB      | 722    | 2,894  | 6,485         | 8.99   |
| Target  | 1,315  | 4,246  | 7,458         | 5.67   |

TABLE V
Comparison between our proposal and CDRSs that require overlapping users.

| Scenario      | Method     | HR@1    | HR@5    |
|---------------|------------|---------|---------|
| AM → AB       | CGN        | 0.036 ± 0.001 | 0.131 ± 0.002 |
|               | BiTGCF     | 0.059 ± 0.002 | 0.217 ± 0.002 |
|               | SCDGN (ours) | 0.094 ± 0.003 | 0.171 ± 0.003 |
| AB → AM       | CGN        | 0.032 ± 0.001 | 0.173 ± 0.001 |
|               | BiTGCF     | 0.087 ± 0.003 | 0.266 ± 0.003 |
|               | SCDGN (ours) | 0.129 ± 0.002 | 0.209 ± 0.003 |

TABLE VI
Performances of variants of SCDGN.

| Dataset | Method     | HR@5    | NDCG@5 |
|---------|------------|---------|--------|
| ML → AM | w/o SI     | 0.044 ± 0.003 | 0.189 ± 0.002 |
|         | w/o DRloss | 0.240 ± 0.003 | 0.186 ± 0.002 |
|         | w/o DB     | 0.229 ± 0.003 | 0.177 ± 0.002 |
|         | SCDGN      | 0.260 ± 0.006 | 0.200 ± 0.005 |
|         | w/o SI     | 0.278 ± 0.006 | 0.199 ± 0.006 |
|         | w/o DRloss | 0.314 ± 0.008 | 0.223 ± 0.010 |
|         | w/o DB     | 0.253 ± 0.006 | 0.183 ± 0.007 |
|         | SCDGN      | 0.321 ± 0.008 | 0.228 ± 0.012 |
D. Ablation Study

To study the impact of different components of SCDGN, we conducted ablation studies on ML→AM and ML→AB with some variants of SCDGN, including (1) w/o SI: SCDGN without user-cluster graph information, which is equal to LightGCN (wt), (2) w/o DRloss: SCDGN without the dimension reduction loss $\mathcal{L}_{dr}$, and (3) w/o DB: SCDGN without debiasing learning mechanism. Table VI shows HR@5 and NDCG@5 of SCDGN and its variants. From this table, we can see that all the information from the user-cluster graph, the metric-invariant dimension reduction, and the debiasing learning boost recommendation accuracy. Specifically, the results decrease the most without the proposed debiasing learning approach. This observation demonstrates that it is necessary to handle domain bias even when transferring the semantic cluster-level interaction information. Besides, the decrement of results on w/o DRloss indicates the effectiveness of constraining the metric relationship when reducing dimension in a sparse domain.

E. Impact of Loss Balance Factors $\lambda_1$ and $\lambda_3$

In this part, we conducted experiments on ML→AM to discuss the impact of the hyper-parameter $\lambda_1$ and $\lambda_3$, where $\lambda_1$ and $\lambda_3$ are the factors to balance the restriction loss and the regularization term, respectively. Figure 2 reports the results on NDCG@K with varying $\lambda_1$ and $\lambda_3$. From this figure, we found that SCDGN achieves the best performance when $\lambda_1 = 1$ and $\lambda_3 = 0.001$. A small $\lambda_1$ produces an under-fitting issue when learning user and item debiasing vector, resulting in an inferior performance. Inversely, a large $\lambda_1$ may introduce noise information from the source domain to mislead the user preference prediction of the target domain. Therefore, a proper $\lambda_3$ is necessary.

V. CONCLUSION

In this work, we proposed a novel semantic clustering enhanced debiasing graph neural recommender system (SCDGN) for cross-domain recommendations with no overlapping user and item between source and target domains. SCDGN exploits semantic features as transferable knowledge to bridge domains and enrich the interaction information of the sparse target domain. Specifically, SCDGN constructs a cross-domain user-cluster graph and develops a new debiasing graph convolutional layer to extract unbiased graph knowledge from the source domain. SCDGN also introduces restriction losses to learn user and item debiasing vectors. Furthermore, we developed a metric-invariant dimension reduction approach to alleviate over-fitting caused by the sparse data. The experimental results on public datasets and a pair of proprietary datasets demonstrate the superiority of SCDGN.

REFERENCES

[1] Z. Li, D. Amagata, T. Maekawa, K. Yonekawa, M. Kurokawa, and T. Har, “Trends-enhanced attention & memory networks for e-commerce recommendation,” in SIGIR Workshop on ECommerce, 2022.
[2] D. Amagata, T. Har, and C. Xiao, “Dynamic set knn self-join,” in ICDE, 2019, pp. 818–829.
[3] T. D. T. Do and L. Cao, “Metadata-dependent infinite poisson factorization for efficiently modelling sparse and large matrices in recommendation,” in IJCAI, 2018, pp. 5010–5016.
[4] Z. Li, D. Amagata, Y. Zhang, T. Maekawa, T. Har, K. Yonekawa, and M. Kurokawa, “Hml4rec: Hierarchical meta-learning for cold-start recommendation in flash sale e-commerce,” Knowledge-Based Systems, vol. 255, p. 109674, 2022.
[5] C. Li, M. Zhao, H. Zhang, C. Yu, L. Cheng, G. Shu, B. Kong, and D. Niu, “Recguru: Adversarial learning of generalized user representations for cross-domain recommendation,” in WSDM, 2022, pp. 571–581.
[6] M. Liu, J. Li, G. Li, and P. Pan, “Cross domain recommendation via bi-directional transfer graph collaborative filtering networks,” in CIKM, 2020, pp. 885–894.
[7] W. Liu, X. Zheng, M. Hu, and C. Chen, “Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation,” in WWW, 2022, pp. 1181–1190.
[8] C. Wang, M. Niepert, and H. Li, “Recsys-dan: Discriminative adversarial networks for cross-domain recommender systems,” Trans. Neural Networks Learn. Syst., vol. 31, no. 8, pp. 2731–2740, 2020.
[9] S. Li, L. Yao, S. Mu, W. X. Zhao, Y. Li, T. Guo, B. Ding, and J. Wen, “Debiasing learning based cross-domain recommendation,” in KDD, 2021, pp. 3190–3199.
[10] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, “Lightgcn: Simplifying and powering graph convolution network for recommendations,” in SIGIR, 2020, pp. 639–648.
[11] Z. Li, D. Amagata, Y. Zhang, T. Har, S. Haruta, K. Yonekawa, and M. Kurokawa, “Debiasing graph transfer learning via item semantic clustering for cross-domain recommendations,” arXiv preprint arXiv:2211.03390, 2022.
[12] Y. Zhang, Y. Liu, P. Han, C. Miao, L. Cui, B. Li, and H. Tang, “Learning personalized itemset mapping for cross-domain recommendation,” in IJCAI, 2020, pp. 2561–2567.
[13] J. Cao, X. Lin, X. Cong, J. Ya, T. Liu, and B. Wang, “Disenclu: Learning disentangled representations for cross-domain recommendation,” in SIGIR, 2022, pp. 267–277.
[14] Z. Chen, R. Xiao, C. Li, G. Ye, H. Sun, and H. Deng, “Essam: Discriminative domain adaptation with non-displayed items to improve long-tail performance,” in SIGIR, 2020, p. 579–588.
[15] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, “Neural graph collaborative filtering,” in SIGIR, 2019, pp. 165–174.
[16] K. Xu, Y. Xie, L. Chen, and Z. Zheng, “Expanding relationship for cross domain recommendation,” in CIKM, 2021, pp. 2251–2260.
[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in NAACL-HLT, 2019, pp. 4171–4186.
[18] D. Amagata and T. Har, “Reverse maximum inner product search: How to efficiently find users who would like to buy my item?” in RecSys, 2019, pp. 4171–4186.
[19] J. MacQueen et al., “Some methods for classification and analysis of multivariate observations,” in Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14, 1967, pp. 281–297.
[20] Z. Yonekawa, H. Niu, M. Kurokawa, A. Kobayashi, D. Amagata, T. Maekawa, and T. Har, “Advertiser-assisted behavioral ad-targeting via denoised distribution induction,” in IEEE BigData, 2019, pp. 5611–5619.
[21] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, “Neural collaborative filtering,” in WWW, 2017, pp. 173–182.
[22] Z. Du, X. Wang, H. Yang, J. Zhou, and J. Tang, “Sequential scenario-specific meta learner for online recommendation,” in KDD, 2019, pp. 2895–2904.