Intra and Inter Domain HyperGraph Convolutional Network for Cross-Domain Recommendation

Zhongxuan Han
College of Computer Science and Technology, Zhejiang University
zxhan@zju.edu.cn

Xiaolin Zheng
College of Computer Science and Technology, Zhejiang University
xlzheng@zju.edu.cn

Chaochao Chen*
College of Computer Science and Technology, Zhejiang University
zjuccc@zju.edu.cn

Wenjie Cheng
College of Computer Science and Technology, Zhejiang University
wenjiec@zju.edu.cn

Yao Yang
Zhejiang Lab
yangyao@zhejianglab.com

ABSTRACT

Cross-Domain Recommendation (CDR) aims to solve the data sparsity problem by integrating the strengths of different domains. Though researchers have proposed various CDR methods to effectively transfer knowledge across domains, they fail to address the following key issues, i.e., (1) they cannot model high-order correlations among users and items in every single domain to obtain more accurate representations; (2) they cannot model the correlations among items across different domains. To tackle the above issues, we propose a novel Intra and Inter Domain HyperGraph Convolutional Network (II-HGCN) framework, which includes two main layers in the modeling process, i.e., the *intra-domain layer* and the *inter-domain layer*. In the intra-domain layer, we design a user hypergraph and an item hypergraph to model high-order correlations inside every single domain. Thus we can address the data sparsity problem better and learn high-quality representations of users and items. In the inter-domain layer, we propose an inter-domain hypergraph structure to explore correlations among items from different domains based on their interactions with common users. Therefore we can not only transfer the knowledge of users but also combine embeddings of items across domains. Comprehensive experiments on three widely used benchmark datasets demonstrate that II-HGCN outperforms other state-of-the-art methods, especially when datasets are extremely sparse.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Cross-Domain Recommendation, Hypergraph, Graph Embedding

*Chaochao Chen is the corresponding author.

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583402

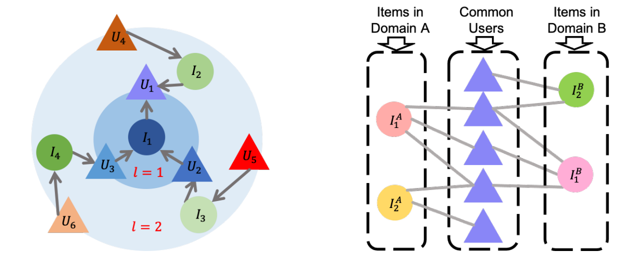
ACM Reference Format:
Zhongxuan Han, Xiaolin Zheng, Chaochao Chen, Wenjie Cheng, and Yao Yang. 2023. Intra and Inter Domain HyperGraph Convolutional Network for Cross-Domain Recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583402

1 INTRODUCTION

Data sparsity is a common problem in the Recommendation System (RS). Nowadays, more and more users participate in multiple domains (platforms) for different purposes, like buying books on Amazon's eBook platform and watching movies on Amazon's Prime Video platform. The data sparsity problem could be mitigated if one can transfer useful knowledge among different domains. Based on this idea, Cross-Domain Recommendation (CDR) [3] was proposed.

Existing research on CDR has introduced different techniques to transfer knowledge across different domains. In the earlier studies, clustering-based [27] and variations of Matrix Factorization (MF)-based CDR methods [18, 32] have been proposed and achieved some improvements compared with single-domain recommendation methods. However, most clustering-based and MF-based CDR methods cannot model the nonlinear patterns in user-item interactions. Thus, researchers designed many deep-learning-based CDR methods [7, 23, 40] to better transfer knowledge across domains and mine more complex user-item interactions. Though existing research has already proved that CDR is a reasonable way to solve the data sparsity problem, they overlook several key issues.

Firstly, the data sparsity problem is not well addressed in CDR since essential high-order correlations among users and items have not been explored in every single domain. Most of the existing CDR methods generate user and item embeddings based on pairwise user-item interactions inside each domain. However, the data sparsity problem is widespread and there is a serious shortage of these pairwise interactions in a sparse domain, which will limit the qualities of the learned embeddings and may lead the negative transfer [34] problem to CDR. To better address the data sparsity problem inside every single domain, some potential high-order correlations should be explored. As Figure 1(a) shows, U_1, U_2, U_3 are likely to be similar since they all interact with I_1. Besides, U_4, U_5, U_6 should have some similar features to U_1, U_2, U_3 since there are direct neighbor relationships among them. Although these kinds of high-order correlations can also be captured by some Graph Neural Networks



(a) High-order correlations among users and items inside a single domain.

(b) Correlations among items in different domains.

Figure 1: Motivating examples. (a) gives an example of high-order correlations among users and items inside a single domain. Paths with arrow indicate the message passing process for U_1. The background circles denote the high-order correlations among users, and the high-order correlation with $l = 1$ is contained by that with $l = 2$. (b) gives an example of correlations among items across different domains. The items that have been interacted with by more common users are more similar, and similar items are shown in similar colors, e.g., I_1^A and I_1^B are more similar than I_1^A and I_2^B.

(GNN)-based CDR methods [15, 43], the graph convolution process is directly related to the degrees of message passing in the graph. For example, the degree of passing message from U_6 to U_1 ($U_6 \rightarrow I_4 \rightarrow U_3 \rightarrow I_1 \rightarrow U_1$) is so high that U_6 has a very limited effect on U_1 by graph convolution. Naturally, if one can link the users or items with high-order correlations and pass messages among them directly, the data sparsity problem will be mitigated.

Secondly, the performance of CDR is always limited since existing CDR methods ignore the correlations among items across domains. The data of users and items are both limited in a sparse domain. However, previous methods mainly focus on learning the overlapping properties of users' preferences and ignore the correlations among items across different domains. Failing to transfer the knowledge of items will lead to inaccurate representations of items, due to the data sparsity problem in every single domain. In fact, correlations among items from different domains can be modeled based on their interactions with common users. For example, as Figure 1(b) shows, I_1^A and I_1^B should be similar since they have been interacted with by a set of common users. By introducing such correlations, CDR methods can learn more accurate items' representations.

In summary, to better address the data sparsity problem in CDR, there are two significant challenges. **CH1:** How to explore and model high-order correlations inside every single domain to better solve the data sparsity problem. **CH2:** How to discover the hidden correlations among items across different domains to address the data sparsity problem better.

In order to address the above two challenges, we propose a novel **I**nter and **I**ntra Domain **H**yper**G**raph **C**onvolutional **N**etwork (II-HGCN) framework in this paper. The main idea of II-HGCN is to enhance the performance of CDR by exploring high-order correlations among users and items. Hypergraph [4] has been proposed to bring more flexibility in handling relationships among nodes in the graph structure. A hypergraph generalizes the concept of an edge to make it connect more than two nodes, providing a natural way to model complex high-order relations among users and items. To the best of our knowledge, we are the first to combine the advantages of hypergraph with CDR to better solve the data sparsity problem.

Our method designs two types of hypergraph structures in the intra-domain layer and the inter-domain layer so that both high-order correlations inside every single domain and across domains can be modeled. To solve **CH1**, we propose a novel *intra-domain hypergraph framework* to model high-order correlations among users and items in every single domain. We build two separate hypergraphs for users and items respectively, and here the hyperedge generation rules can be flexible. For instance, a hyperedge can associate users with similar behaviors or model the similarities among items being interacted with by the same users. Besides, to identify the importance of different hyperedges and avoid generating redundant hyperedges, we define a novel metric *HyperDegree* for a hyperedge so that redundant hyperedges can be filtered out. To solve **CH2**, we design a novel *inter-domain hypergraph framework* to capture correlations among items across domains. Firstly, we define a novel metric *HyperSimilarity* to model the similarity between two items from different domains. Then for each item in a domain, we construct a hyperedge containing several most similar items from another domain so that potential relationships among items from different domains can be explored and transferred. In addition, we propose an adjusted hypergraph convolutional network to aggregate item embeddings in this phase. Finally, an element-wise attention mechanism is used to combine embeddings learned from the intra-domain and inter-domain processes for users and items.

To evaluate the performance of our II-HGCN framework, we conduct extensive experiments on three real-world datasets. Multiple evaluation metrics demonstrate that II-HGCN outperforms the State-Of-The-Art (SOTA) models [7, 40] from various perspectives. Furthermore, after sparsifying the dataset, we find that the improvement of II-HGCN against the SOTA models is more significant, proving that our framework can better deal with the data sparsity problem.

We summarize our main contributions as follows: (1) We propose a novel II-HGCN framework, which can model the high-order correlations among users and items to better address the data sparsity problem in CDR. (2) We design an intra-domain layer and an inter-domain layer to capture high-order correlations inside every single domain and transfer knowledge of both users and items across domains. (3) We conduct extensive experiments on three real-world datasets, demonstrating that II-HGCN outperforms SOTA methods, and the improvement is more significant in sparser datasets.

2 RELATED WORK

2.1 Cross-Domain Recommendation

CDR was proposed to solve the data sparsity problem by transferring knowledge across different domains [3]. Early CDR methods are mainly based on MF [18, 26, 29, 36] and clustering methods [12, 32]. For example, Collective Matrix Factorization (CMF) [36] utilizes multiple auxiliary matrices on users to combine users' features across different domains. While cluster-level matrix factorization [32] uses the K-means method to explore the shared patterns of users and items between two domains. Though these non-deep-learning based CDR methods have achieved better performance compared with single-domain recommendation methods, they cannot model complex patterns in user-item interactions.

Deep-learning can model user-item interactions more flexibly and learn better representations of users and items. Researchers have proposed many deep-learning-based CDR methods [5, 7, 9, 17, 28, 31, 40, 41]. Lm et al. [22] pointed out that users' search modes play an essential role in improving the recommendation accuracy in CDR. Hu et al. [17] proposed a multi-task learning strategy by building a deep cross-connection network to transfer knowledge between user-item interactions across domains. DARec [40] learns shared user representations across domains inspired by domain adaptation technique. DDTCDR [28] proposes a deep dual transfer network to learn user embeddings jointly and combine embeddings from different domains. ETL [7] considers both the users' preference for different domains and the domain-specific properties, and thus has achieved SOTA performance. Some researchers [17, 30, 43] tend to incorporate content information in CDR to mitigate the data sparsity problem in each domain. While content information like attributes [3], social tags [11], and browsing or watching histories [24] is not always available.

Though existing CDR methods have achieved good performance, they only learn representations of users and items from pairwise correlations, which can be severely affected by the data sparsity problem inside every single domain. Besides, previous methods only focus on learning the overlapping properties of users and ignoring the correlations among items that are mostly non-overlapping across different domains. As a comparison, in this paper, we introduce a framework II-HGCN to capture both high-order correlations inside each single domain and hidden correlations among items across domains to better solve the data sparsity problem.

2.2 HyperGraph

The hypergraph [6, 13, 21, 42] structure has been employed to model high-order correlations among data. Zhou et al. [42] first proposed hypergraph learning and designed a propagation process on hypergraph structure. Hypergraph was further employed in the video object segmentation task [19] and the image retrieval task [20]. Furthermore, Uthsav et al. [8] introduced a spectral theory for hypergraphs with edge-dependent vertex weights by the random walk method. Later on, Feng et al. [10] introduced a hypergraph convolution operation to better exploit the high-order data correlations for representation learning, which provides a more effective way to deal with complex correlations.

A graph structure can naturally model user-item interactions of RS and some hypergraph-based recommendation methods [23, 38, 39] have been proposed to explore implicit high-order correlations in RS and achieved good performance. In this paper, we design a hypergraph based framework to better solve the data sparsity problem in CDR.

3 THE PROPOSED METHOD

In this section, we first give the problem formulation and then describe the details of our proposed II-HGCN.

3.1 Problem Formulation

Problem Definition. We assume there are two domains, i.e., A and B, who have the same set of users $\mathcal{U} = \{U_1, U_2, \dots, U_N\}$ with N denoting the number of users. The item sets of domain A and domain B are $\mathcal{I}^A = \{I_1^A, I_2^A, \dots, I_M^A\}$ and $\mathcal{I}^B = \{I_1^B, I_2^B, \dots, I_T^B\}$, where M and T denote the number of items in each domain, respectively. The user-item interactions of domain A and domain B are represented by matrixs $\mathbf{R}^A \in \{0,1\}^{N \times M}$ and $\mathbf{R}^B \in \{0,1\}^{N \times T}$. Usually, $\mathbf{R}^A$ and $\mathbf{R}^B$ are very sparse since users can only interact with a small subset of items in each domain. CDR aims to provide users with accurate recommendation results in a domain with the help of the other domain. In this paper, we do not distinguish a source or target domain since the recommendation process for each domain is performed in a unified way.

Representation of HyperGraph. A hypergraph can be represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the vertex set and $\mathcal{E}$ represents the edge set. An edge can connect two or more vertices in a hypergraph [4]. An adjacency matrix $\mathbf{H} \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ is used to represent the connections among vertices on the hypergraph, where $\mathbf{H}_{ve} = 1$ indicates vertex v belongs to hyperedge e. Two diagonal matrices $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ and $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ are used to represent the degrees of vertices and edges respectively, where $(\mathbf{D}_v)_{vv} = \sum_{e \in \mathcal{E}} \mathbf{H}_{ve}$ and $(\mathbf{D}_e)_{ee} = \sum_{v \in \mathcal{V}} \mathbf{H}_{ve}$.

Notations. We list the important notations in **Appendix A**.

3.2 Overview

In this paper, we propose a novel **I**ntra and **I**nter-Domain **H**yper**G**raph **C**onvolutional **N**etwork for CDR, namely II-HGCN, to better solve the data sparsity problem. As shown in Figure 2, the framework of II-HGCN is divided into four components, i.e., *input layer*, *intra-domain layer*, *inter-domain layer*, and *prediction layer*. We generate the embeddings of users and items from each domain in the *input layer*, and calculate the recommendation result in the *prediction layer*. In the main modeling process, we explore the high-order correlations inside each domain and across domains in the rest two layers to solve **CH1** and **CH2**, respectively. In the *intra-domain layer*, we model each domain's high-order correlations to learn better user and item embeddings. In the *inter-domain layer*, we transfer useful knowledge of both users and items across domains to enhance the recommendation performance. We will describe the details of the intra-domain layer and the inter-domain layer in the following subsections. Then we will introduce the prediction layer and the optimization strategy.

3.3 Intra-Domain Layer (Solving CH1)

As shown in Figure 2, in the intra-domain layer, domain A and domain B share the same process to learn user and item embeddings, i.e., firstly building two hypergraphs for users and items respectively, and then utilizing the hypergraph convolution network to update embeddings. For readability, we construct a common domain to introduce this unified intra-domain process. We suppose there are N users and M items in this common domain. The user set is $\mathcal{U} = \{U_1, U_2, \dots, U_N\}$, the item set is $\mathcal{I} = \{I_1, I_2, \dots, I_M\}$, and the rating matrix is formed as $\mathbf{R} \in \{0,1\}^{N \times M}$.

In the intra-domain layer, the modeling processes of users and items are symmetrical. Without loss of generality, we take the modeling process of users as an example to describe the intra-domain layer in detail.

3.3.1 *HyperGraph Construction for Users.*

To solve **CH1**, we propose a novel hypergraph construction method to explore high-order correlations inside every single domain. Before presenting the details, we first give the definition of *Item's k-order Reachable Users* according to [23] as follows.

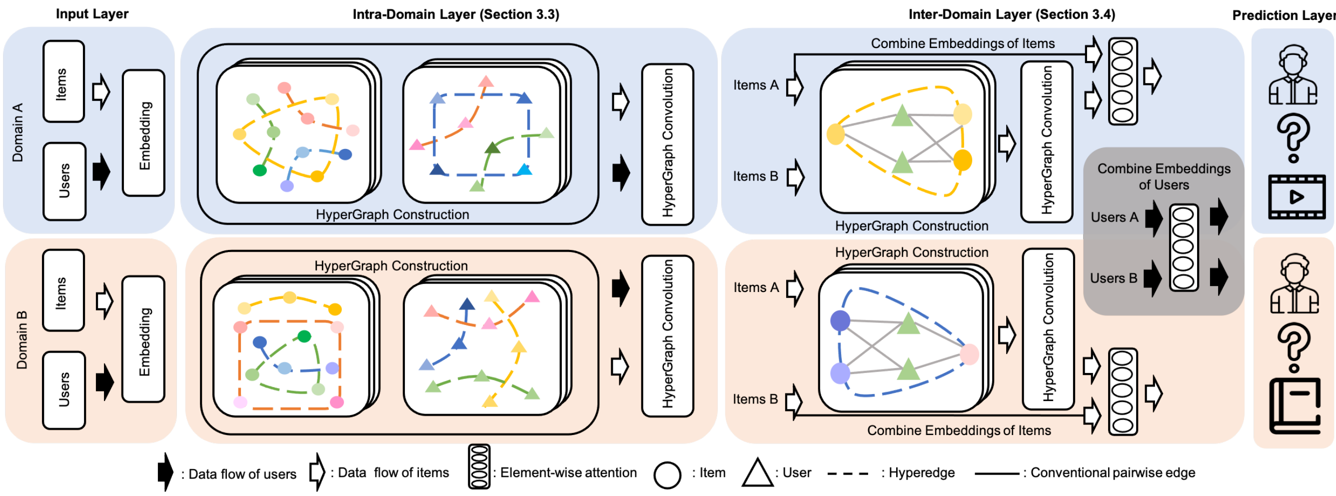


Figure 2: The proposed framework of II-HGCN. The modeling process for domain A is shown in blue, while the modeling process for domain B is shown in orange. The common modeling process for combining embeddings of users is shown in gray. Similar users and items are shown in similar colors.

Definition 3.1 (Item's k-order Reachable Users). In a user-item bipartite graph, U_j is k-order reachable from I_i if at least one of the direct paths between U_j and I_i exists k or less than k users.

The basic idea of constructing a hypergraph for users can be summarized into three steps.

Step 1, extracting an item's k-order reachable user set as a hyperedge. For I_i, its k-order reachable user set is termed as $J^k_{\mathcal{U}}(I_i)$.

Step 2, combining k-order reachable user sets for all items into a k-order user hypergroup $\mathbf{H}^k_{\mathcal{U}} = \{J^k_{\mathcal{U}}(I_i)|I_i \in \mathcal{I}\}$, where $\mathbf{H}^k_{\mathcal{U}}$ denotes the k-order user hypergroup. The matrix form of constructing $\mathbf{H}^k_{\mathcal{U}}$ is as follows:

$$\mathbf{H}^k_{\mathcal{U}} = \mathbf{R} \cdot min(1, pow(\mathbf{R}^T \cdot \mathbf{R}, k-1)), \tag{1}$$

where $pow(\mathbf{R}, k)$ is the function that calculates the k power of a matrix $\mathbf{R}$, $min(x, \mathbf{R})$ is the function that replace all elements bigger than x in matrix $\mathbf{R}$ with x, and $\cdot$ denotes the matrix multiplication.

Step 3, aggregating all k-order user hypergroups for $k = 1, 2, \ldots, k_{intra}$ to generate the user hypergraph. k_{intra} indicates the furthest reachable users we consider for each item. In this paper, we use a simple concatenation operation $||$ to aggregate all hypergroups:

$$\mathbf{H}_{\mathcal{U}} = \mathbf{H}^1_{\mathcal{U}}||\mathbf{H}^2_{\mathcal{U}}||\ldots||\mathbf{H}^{k_{intra}}_{\mathcal{U}}. \tag{2}$$

We give an example in **Appendix B.1** to show the hypergraph construction rule in the intra-domain layer.

However, the above method simply extracts all items' k-order reachable users to construct hypergroups, but no consideration is given to whether every hyperedge is useful. For some users who have interacted with many items (active users), their embeddings can be well learned just based on the neighbors in close proximity. In this case, some noise will be added to these active users by including redundant hyperedges. To avoid such negative effects, we aim to remove redundant hyperedges so that on the one hand, more high-order correlations can be explored for less active users, and on the other hand, active users can be less affected.

Firstly, to evaluate the importance of a hyperedge, we define *HyperDegree*, a novel metric for a hyperedge by adding up all vertices' degrees in this hyperedge. Note that a hyperedge with a big hyperdegree means it has already contained some active users (its higher-order reachable users maybe redundant). The matrix form of calculating hyperdegrees for k-order user hypergroup $\mathbf{D}^k_{\mathcal{U}} \in \{0,1\}^{M\times M}$ is $\mathbf{D}^k_{\mathcal{U}} = f(\mathbf{H}^k_{\mathcal{U}} \odot (\mathbf{D}^k_{\mathcal{U}_v} \cdot \mathbf{1}))$, where $\odot$ is the matrix element-wise multiplication, $\mathbf{1} \in \{1\}^{M\times M}$ is a matrix with all elements being 1, $\mathbf{D}^k_{\mathcal{U}_v}$ indicates the vertex degree matrix of $\mathbf{H}^k_{\mathcal{U}}$, and $f(\mathbf{R})$ denotes the operation that converts a matrix $\mathbf{R} \in \mathbb{R}^{N\times M}$ into a diagonal matrix $\mathbf{R} \in \mathbb{R}^{M\times M}$ by summing up all elements in each column and storing the results in diagonal positions.

Secondly, to evaluate whether each hyperedge should be added to the k-order hypergroup, we sum up the hyperdegrees of each hyperedge in $\{1, 2, \ldots, k-1\}$-order hypergroups as:

$$\tilde{\mathbf{D}}^k_{\mathcal{U}} = \mathbf{D}^1_{\mathcal{U}} + \mathbf{D}^2_{\mathcal{U}} + \cdots + \mathbf{D}^{k-1}_{\mathcal{U}}, \tag{3}$$

where $\tilde{\mathbf{D}}^k_{\mathcal{U}}$ denotes the summing result and + denotes the matrix addition.

Thirdly, to filter the redundant hyperedges, we calculate the median value X^k of all values in $\tilde{\mathbf{D}}^k_{\mathcal{U}}$, and then generate a diagonal matrix $\mathbf{F}^k_{\mathcal{U}} \in \{0,1\}^{M\times M}$ to remove redundant hyperedges:

$$(\mathbf{F}^k_{\mathcal{U}})_{ii} = \begin{cases} 1, & \text{if } (\tilde{\mathbf{D}}^k_{\mathcal{U}})_{ii} < X^k, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Finally, the adjusted hypergraph of users can be formulated as:

$$\mathbf{H}_{\mathcal{U}} = \mathbf{H}^1_{\mathcal{U}}||\mathbf{H}^2_{\mathcal{U}} \cdot \mathbf{F}^2_{\mathcal{U}}||\ldots||\mathbf{H}^{k_{intra}}_{\mathcal{U}} \cdot \mathbf{F}^{k_{intra}}_{\mathcal{U}}. \tag{5}$$

Thus, only hyperedges containing some inactive users will be added to each hypergroup.

3.3.2 HyperGraph Convolution for users. We design a hypergraph convolution network to update embeddings $\mathbf{E}_{\mathcal{U}}$ generated from the input layer. To distill discriminative information and model the correlations among users and items, we apply a shared parameter $\mathbf{W}^l \in \mathbb{R}^{c(l)\times c(l+1)}$ for users and items in each convolution layer, where $c(l)$ indicates the output embedding size of layer l. The matrix form of the hypergraph convolution process for users is as follows:

$$\mathbf{E}_{\mathcal{U}}^{l+1} = \delta(\mathbf{D}_{\mathcal{U}_v}^{-1/2}\mathbf{H}_{\mathcal{U}}\mathbf{D}_{\mathcal{U}_e}\mathbf{H}_{\mathcal{U}}^T\mathbf{D}_{\mathcal{U}_v}^{-1/2}\mathbf{E}_{\mathcal{U}}^l\mathbf{W}^l + \mathbf{E}_{\mathcal{U}}^l), \tag{6}$$

where $\mathbf{D}_{\mathcal{U}_v} \in \mathbb{R}^{N\times N}$ and $\mathbf{D}_{\mathcal{U}_e} \in \mathbb{R}^{M\times M}$ are vertex and hyperedge degrees of the user hypergraph, and δ is the activation function. We add the resnet-like skip connection to allow the model simultaneously considers both its original features and features aggregated from the hypergraph.

The modeling process for items is similar to which of users and we show the details in **Appendix B.2**. Based on the unified intra-domain layer framework, we can learn high-quality representations $(\mathbf{E}_{\mathcal{U}_A}^l, \mathbf{E}_{\mathcal{I}_A}^l)$, $(\mathbf{E}_{\mathcal{U}_B}^l, \mathbf{E}_{\mathcal{I}_B}^l)$ of users and items in domain A and domain B for the further inter-domain process. By addressing the strengths of hypergraph, we can explore high-order correlations inside every single domain and solve the data sparsity problem better, as specified in **CH1**.

3.4 Inter-Domain Layer (Solving CH2)

In this section, we will introduce how to combine the features of users and items between domain A and domain B who share the same set of users. As the modeling process of domain A and domain B is symmetrical, we take domain A as an example to describe the inter-domain layer in detail. The learning process of domain B is shown in **Appendix B.3**.

As shown in Fig. 2, the embedding combination processes of users and items are separate and distinct. Firstly, we will introduce the embedding combination process of users, and then we will describe how to combine embeddings of items by exploring high-order correlations among items across domains.

3.4.1 Combine Embeddings of Users. Since users are overlapping across domains and we want to identify the different proportions of features learned from different domains, we use an element-wise attention mechanism [43] to combine the user embeddings learned from domain A and domain B. Compared with the traditional attention mechanism [2], the element-wise attention mechanism allows more flexibility in identifying the importance of each element of embeddings learned from different domains. Given the output embeddings $\mathbf{E}_{\mathcal{U}_A}^l$ and $\mathbf{E}_{\mathcal{U}_B}^l$ from the l-th layer of the hypergraph convolutional network in the intra-domain process, the combined embeddings of users in domain A is calculated as follows:

$$\tilde{\mathbf{E}}_{\mathcal{U}_A}^l = \mathbf{E}_{\mathcal{U}_A}^l \odot \mathbf{W}_{\mathcal{U}_A}^l + \mathbf{E}_{\mathcal{U}_B}^l \odot (1 - \mathbf{W}_{\mathcal{U}_A}^l), \tag{7}$$

where $\mathbf{W}_{\mathcal{U}_A}^l \in \mathbb{R}^{N\times c(l)}$ is the weight matrices for the attention network of domain A.

3.4.2 Combine Embeddings of Items. Different domains always contain entirely different items. Existing CDR methods have not given a good solution to model the correlations among items across domains. In our framework, we propose a hypergraph-based method to explore potential correlations among items based on their interactions with common users to enhance the recommendation performance. We divide the embedding combination process into two phases: *HyperGraph Construction* and *HyperGraph Convolution.*

HyperGraph Construction. To find high-order relationships among items in each domain, we define the *HyperSimilarity* $\mathbf{S}_{ij}$ between two items I_i^A and I_j^B by calculating the number of users who interact with both I_i^A and I_j^B as $\mathbf{S}_{ij}^A = \sum(\mathbf{R}_{*i}^B \circledast \mathbf{R}_{*j}^A)$, where $\circledast$ denotes the element-wise AND operation, $\sum$ indicates the operation of summing up all values in a vector, and $\mathbf{R}_{*i}$ denotes the i-th column of matrix $\mathbf{R}$. It should be noticed that the time complexity of constructing $\mathbf{S}^A$ is the same as doing matrix multiplication on matrices $\mathbf{R}^A$ and $\mathbf{R}^B$, which means the time consuming of calculating the hypersimilarity matrix is low.

Based on the hypersimilarity matrix $\mathbf{S}^A$, we can construct the hypergraphs $\mathbf{H}_S^A \in \mathbb{R}^{T\times M}$ by extracting top$k$ most similar items in domain B as $(\mathbf{H}_S^A)_{*i} = topK(\mathbf{S}_{*i}^A, k)$, where $topK$ indicates the function of saving top$k$ biggest elements in a vector and replacing all other elements to be 0. We set the value of k to be k_{inter}, which means for every item, we consider the topk_{inter} most similar items in the other domain. We give an example in **Appendix B.4** to show the hypergraph construction rule in the inter-domain layer.

HyperGraph Convolution. Based on the hypergraphs $\mathbf{H}_S^A$, we propose an *adjusted hypergraph convolution network* to transfer knowledge of items from domain B to domain A:

$$\mathbf{P}_{\mathcal{I}_A}^l = \delta(\mathbf{D}_{\mathbf{H}_{S\,v}^A}^{-1/2}\mathbf{H}_S^A\mathbf{D}_{\mathbf{H}_{S\,e}^A}^{-1/2}\mathbf{E}_{\mathcal{I}_B}^l), \tag{8}$$

where $\mathbf{D}_{\mathbf{H}_{S\,v}^A} \in \mathbb{R}^{T\times T}$ and $\mathbf{D}_{\mathbf{H}_{S\,e}^A} \in \mathbb{R}^{M\times M}$ denotes the vertex and hyperedge degrees of $\mathbf{H}_S^A$, $\mathbf{P}_{\mathcal{I}_A}^l \in \mathbb{R}^{M\times c(l)}$ is the aggregated item embeddings.

To combine the features of items learned from the intra-domain layer and the inter-domain layer, we also use the element-wise attention network to combine them together:

$$\tilde{\mathbf{E}}_{\mathcal{I}_A}^l = \mathbf{E}_{\mathcal{I}_A}^l \odot \mathbf{W}_{\mathcal{I}_A}^l + \mathbf{P}_{\mathcal{I}_A}^l \odot (1 - \mathbf{W}_{\mathcal{I}_A}^l), \tag{9}$$

where $\mathbf{W}_{\mathcal{I}_A}^l \in \mathbb{R}^{M\times c(l)}$ is the trainable parameter for the l-th layer of the element-wise attention network for items.

Suppose the number of layers of the hypergraph convolution network in the intra-domain process is L, we can generate the final embeddings of users and items in domain A as:

$$\tilde{\mathbf{E}}_{\mathcal{U}_A} = \tilde{\mathbf{E}}_{\mathcal{U}_A}^1||\tilde{\mathbf{E}}_{\mathcal{U}_A}^2||\dots||\tilde{\mathbf{E}}_{\mathcal{U}_A}^L,\ \tilde{\mathbf{E}}_{\mathcal{I}_A} = \tilde{\mathbf{E}}_{\mathcal{I}_A}^1||\tilde{\mathbf{E}}_{\mathcal{I}_A}^2||\dots||\tilde{\mathbf{E}}_{\mathcal{I}_A}^L, \tag{10}$$

where $\tilde{\mathbf{E}}_{\mathcal{U}_A}$ and $\tilde{\mathbf{E}}_{\mathcal{I}_A}$ denote the combined embeddings.

Through the modeling process of the inter-domain layer, we can transfer knowledge of both users and items across different domains, so that we can address the **CH2** and solve the data sparsity problem better.

3.5 Prediction Layer and Optimization Strategy

Prediction Layer. After generating final embeddings of users and items in each domain, we use the cosine similarity to decide the possibility $\hat{\mathbf{R}}_{ij}^A$ of whether U_i^A will interact with I_j^A:

$$\hat{\mathbf{R}}_{ij}^A = \frac{(\tilde{\mathbf{E}}_{\mathcal{U}_A})_i \cdot (\tilde{\mathbf{E}}_{\mathcal{I}_A})_j}{||(\tilde{\mathbf{E}}_{\mathcal{U}_A})_i||\ ||(\tilde{\mathbf{E}}_{\mathcal{I}_A})_j||}. \tag{11}$$

The possibility $\hat{\mathbf{R}}_{ij}^B$ of whether U_i^B will interact with I_j^B can be calculated in the same way.

Optimization Strategy. We choose the binary cross-entropy loss to optimize our model [35]. Taking domain A's loss function $\mathcal{L}_A$ as an example:

$$\mathcal{L}_A = \sum_{U_i^A \in \mathcal{U}_A, I_j^A \in \mathcal{I}_A} \mathbf{R}_{ij}^A \log \hat{\mathbf{R}}_{ij}^A + (1 - \mathbf{R}_{ij}^A)(1 - \log \hat{\mathbf{R}}_{ij}^A). \tag{12}$$

Similarly, we can obtain the loss function $\mathcal{L}_B$ for domain B. We sum up them to get the total loss function for optimization: $\mathcal{L} = \mathcal{L}_{\mathcal{A}} + \mathcal{L}_{\mathcal{B}}$.

4 EXPERIMENTS AND ANALYSIS

To fully evaluate the proposed II-HGCN framework, we conduct extensive experiments on three real-world datasets to answer the following questions: **Q1:** How does our II-HGCN outperform the state-of-the-art single-domain models? **Q2:** How does our II-HGCN perform compared with the state-of-the-art CDR models? **Q3:** Can our II-HGCN address the data sparsity problem better than other CDR models? **Q4:** How do combining embeddings of users and combining embeddings of items across domains contribute to performance improvement? **Q5:** How do important hyperparameters affect II-HGCN?

4.1 Dataset and Experimental Settings

Dataset Description. We choose three domains from the Amazon dataset[1]: Movies and TV (**Movie**), Books (**Book**), CDs and Vinyl (**Music**) to evaluate the performance of our proposed II-HGCN. These three domains are benchmarks for CDR and have been widely used in recent work [7, 17, 40]. We preprocess these domains as follows to construct experimental datasets for CDR. *Firstly*, since the user-item interaction information in these three domains is basically ratings ranging from 1 to 5, we convert the ratings of 3, 4, and 5 as positive samples and others including non-rating items as negative samples [17]. *Next*, following [7, 17, 40], we combine each of the two domains to obtain **Movie & Book**, **Movie & Music**, and **Music & Book** as our experimental cross-domain datasets, where each domain pairs share the same set of users. *Then*, for each experimental dataset, we filter users and items whose total number of interactions in two domains is less than 5 [7]. As shown in Table 4 in **Appendix C.1**, we can find that both domains for each dataset are still extremely sparse, with at least 99.86% interactions being unobserved. The severe data sparsity problem brings a great challenge to existing CDR methods.

Evaluation Protocols. *Firstly*, we split each experimental dataset into the train set, the validation set, and the test set. Following [7, 16, 17], we utilize leave-one-out (LOO) to do the dataset splitting process. In detail, for each user, we randomly select two items from the positive samples, one as the validation item and the other one as the test item. After it, the remaining items are all considered as train items. *Secondly*, we train and evaluate each model as follows. Following [7, 16, 17], we randomly sample 99 items from negative samples for each user and then evaluate how recommendation models can rank the validation and the test item against these negative items. During the training process, we save the model with the best performance on the validation set. *Finally*, we perform testing with the saved model. In order to comprehensively evaluate the performance of each model, we adopt three widely used metrics [7, 17], i.e., Hit Ratio (HR), Normalized Discounted Cumulative Gain (NDCG), and Mean Reciprocal Rank (MRR). A higher value means a better recommendation performance for all these three metrics and the predicted ranking cut-off is set as top$K = 5, 10$ [7].

Baselines. We compare our II-HGCN with seven baselines, including single-domain methods (PMF, NCF, and NGCF) and cross-domain methods (CoNet, DDTCDR, DARec, and ETL): (1) **PMF** [33]: Probabilistic matrix factorization is a classic factorization-based method for single-domain recommendation. (2) **NCF** [16]: Neural network-based collaborative filtering replaces the inner product with a neural architecture that can learn an arbitrary function from data. (3) **NGCF** [37]: Neural graph collaborative filtering exploits the user-item graph structure by propagating embeddings on it. Note that NGCF also considers the high-order connectivity in an item-user bipartite graph. (4) **CoNet** [17]: CoNet proposes a modified cross-stitch neural network to transfer knowledge between two domains. (5) **DDTCDR** [28]: DDTCDR transfers knowledge across two domains by designing a deep dual transfer network. (6) **DARec** [40]: DARec learns shared user representations across different domains based on the domain adaptation technique. (7) **ETL** [7]: ETL is a recent state-of-the-art CDR model that captures both the overlapping and domain-specific properties to adopt equivalent transformations across two domains.

Besides, we also do ablation experiments to explore the influence of the intra-domain layer and the inter-domain layer for II-HGCN: (1) **II-HGCN-S** indicates the model with only the intra-domain layer, which is a type of single-domain method. (2) **II-HGCN-U** denotes the model which only combines the embeddings of users across two domains in the inter-domain layer. (3) **II-HGCN-I** denotes the model which only combines the embeddings of items across two domains in the inter-domain layer.

Parameter Settings. For a fair comparison, the batch size is set to 256 for all methods. Besides, we use the Adam optimizer [25] with the learning rate as 0.001 to optimize all models, and the Xavier initializer [14] to initialize all models' parameters. To ensure the convergence for all models, we set the number of training epochs to 300. We optimize the unique parameters of all baseline models to get better performance. For II-HGCN, we set k_{intra} as 2 and k_{inter} as 5 based on the experiments of hyperparameters. The number of layers of the hypergraph convolutional network in the intra-domain layer is set to 2 and the embedding size is set to 128. All activation functions in II-HGCN are ReLU [1].

4.2 Overall Comparison (RQ1, RQ2)

To answer Q1 and Q2, we conduct experiments on three experimental datasets to compare the performance of II-HGCN with single-domain methods and cross-domain methods. The results are reported in Table 1, and Table 5 in **Appendix C.2**. *Improv.* is calculated compared with the most competitive baseline.

To answer Q1. In all these three experimental datasets, cross-domain methods generally outperform the single-domain methods, indicating the importance of transferring knowledge across domains in recommendation. This is because these datasets are extremely sparse, and CDR can combine each domain's strengths to mitigate the data sparsity problem.

Comparing the performance of II-HGCN with single-domain methods, generally, our II-HGCN framework significantly outperforms all single-domain models. In each dataset, II-HGCN-S with

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

Table 1: Experiment on Movie & Book Dataset

Movie & Book												
topK	topK = 5						topK = 10					
Domain	Movie			Book			Movie			Book		
Metrics	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR
PMF	0.3889	0.2771	0.2402	0.3057	0.2214	0.1936	0.5218	0.3200	0.2579	0.4100	0.2550	0.2075
NCF	0.4320	0.3062	0.2779	0.3756	0.2759	0.2434	0.5498	0.3474	0.2848	0.4838	0.3113	0.2580
NGCF	0.4275	0.2993	0.2672	0.3685	0.2702	0.2418	0.5544	0.3496	0.2851	0.4903	0.3196	0.2608
CoNet	0.3846	0.2686	0.2304	0.3084	0.2167	0.1866	0.5206	0.3125	0.2485	0.4370	0.2581	0.2036
DDTCDR	0.3968	0.2670	0.2309	0.2979	0.2042	0.1766	0.5593	0.3298	0.2585	0.4439	0.2615	0.2119
DARec	0.4590	0.3270	0.2834	0.4196	0.2839	0.2494	0.6008	0.3729	0.3023	0.5368	0.3350	0.2962
ETL	0.4953*	0.3687*	0.3267*	0.4963*	0.3830*	0.3456*	0.6253*	0.4108*	0.3440*	0.6179*	0.4223*	0.3617*
II-HGCN-S	0.4552	0.3251	0.2812	0.4386	0.3339	0.2995	0.5717	0.3527	0.2910	0.5522	0.3630	0.3260
II-HGCN-I	0.4811	0.3628	0.3246	0.4906	0.3756	0.3477	0.6051	0.3942	0.3513	0.6085	0.4104	0.3820
II-HGCN-U	0.5035	0.3756	0.3456	0.5151	0.3967	0.3509	0.6286	0.4171	0.3626	0.6261	0.4254	0.3870
II-HGCN	**0.5298**	**0.3921**	**0.3472**	**0.5328**	**0.4218**	**0.3852**	**0.6544**	**0.4361**	**0.3654**	**0.6346**	**0.4579**	**0.4009**
Improv.	6.97%	6.35%	6.27%	7.35%	10.13%	11.46%	4.65%	6.16%	6.22%	2.70%	8.43%	10.84%

Note that the results marked with * are the best performing baselines.

Table 2: Experimental results on sparse dataset for domain Music & Book

Music & Book													
Domain		Music						Book					
Sparse Ratio		0	10%	20%	30%	40%	50%	0	10%	20%	30%	40%	50%
HR@5	ETL	0.4638	0.4330	0.4026	0.3746	0.3295	0.2712	0.4594	0.4315	0.4031	0.3764	0.3357	0.2773
	II-HGCN	0.5008	0.4740	0.4477	0.4219	0.3913	0.3489	0.4762	0.4516	0.4253	0.4033	0.3707	0.3531
	Improv.	**7.98%**	**9.47%**	**11.20%**	**12.63%**	**18.76%**	**28.65%**	**3.66%**	**4.66%**	**5.51%**	**7.15%**	**10.43%**	**27.34%**
NDCG@5	ETL	0.3572	0.3320	0.3010	0.2814	0.2479	0.1984	0.3587	0.3353	0.3094	0.2842	0.2538	0.1890
	II-HGCN	0.3839	0.3663	0.3391	0.3198	0.2980	0.2776	0.3771	0.3571	0.3312	0.3145	0.2893	0.2556
	Improv.	**7.47%**	**10.33%**	**12.66%**	**13.65%**	**20.21%**	**39.92%**	**5.13%**	**6.50%**	**7.05%**	**10.66%**	**13.99%**	**35.24%**
MRR@5	ETL	0.3225	0.2985	0.2733	0.2504	0.2195	0.1764	0.3203	0.3028	0.2777	0.2534	0.2239	0.1634
	II-HGCN	0.3457	0.3294	0.3038	0.2806	0.2643	0.2451	0.3435	0.3256	0.3004	0.2828	0.2597	0.2369
	Improv.	**7.19%**	**10.35%**	**11.16%**	**12.06%**	**20.41%**	**38.95%**	**7.24%**	**7.53%**	**8.17%**	**11.60%**	**15.99%**	**44.98%**
HR@10	ETL	0.5867	0.5548	0.5270	0.4943	0.4452	0.3737	0.5728	0.5463	0.5195	0.4909	0.4472	0.3836
	II-HGCN	0.6190	0.5917	0.5684	0.5422	0.5188	0.4807	0.5955	0.5691	0.5434	0.5261	0.4858	0.4417
	Improv.	**5.51%**	**6.65%**	**7.86%**	**9.69%**	**16.53%**	**28.63%**	**3.96%**	**4.17%**	**4.60%**	**7.17%**	**8.63%**	**15.15%**
NDCG@10	ETL	0.3972	0.3710	0.3412	0.3199	0.2837	0.2314	0.3954	0.3714	0.3455	0.3208	0.2898	0.2242
	II-HGCN	0.4239	0.4052	0.3786	0.3556	0.3368	0.3145	0.4221	0.3981	0.3735	0.3566	0.3343	0.2957
	Improv.	**6.72%**	**9.22%**	**10.96%**	**11.16%**	**18.72%**	**35.91%**	**6.75%**	**7.19%**	**8.10%**	**11.16%**	**15.36%**	**31.89%**
MRR@10	ETL	0.3390	0.3145	0.2893	0.2662	0.2341	0.1898	0.3357	0.3177	0.2925	0.2685	0.2384	0.1778
	II-HGCN	0.3629	0.3454	0.3201	0.2967	0.2800	0.2547	0.3581	0.3408	0.3212	0.3001	0.2755	0.2413
	Improv.	**7.05%**	**9.83%**	**10.65%**	**11.46%**	**19.61%**	**34.19%**	**6.67%**	**7.27%**	**9.81%**	**11.77%**	**15.56%**	**35.71%**

only the intra-domain layer has already got the best performance among all single-domain models. While II-HGCN still improves the performance by at least 13.83% and 14.62% for NDCG@5 and NDCG@10 compared with II-HGCN-S, which proves that the embeddings learned from a sparse domain can be optimized with the help of the other domain.

Compared with NGCF which also considers high-order correlations among users and items, our proposed II-HGCN-S achieves better performance. Since by using hypergraphs, we can model high-order relationships in a more flexible way. Besides, users and items with high-order correlations can directly pass messages to each other based on hyperedges instead of using other nodes as bridges in conventional graph structures.

To answer Q2. II-HGCN yields consistent best performance compared with all CDR methods on all datasets. In particular, II-HGCN improves the strongest baseline (i.e., ETL) by 6.27% in average in terms of NDCG@5 and 6.11% in average in terms of NDCG@10.

We attribute the improvement mainly to the flexible and explicit modeling of high-order correlations inside each domain and the ability to combine items' embeddings across domains. Specifically, (1) in the intra-domain layer, we construct user and item hypergraphs for each domain, respectively, to explore high-order relationships. Thus we can deal with the data sparsity problem better than other CDR methods and learn more accurate embeddings from each domain; (2) in the inter-domain layer, we can not only combine embeddings of users but also transfer knowledge of items across two domains. In each dataset, items are totally non-overlapping across domains, and all listed CDR methods cannot model the correlations among items from different domains. In contrast, we propose a hypergraph-based method in II-HGCN to explore the high-order relationships among items according to their interactions with common users. Thus, the embeddings of items in each domain can be optimized with the help of the other domain, and we can calculate more accurate recommendation results compared with other CDR methods.

Compared with ETL, II-HGCN consistently yields better performance. ETL is the strongest baseline among all listed methods, which intends to learn the domain-specific properties as well as the overlapping users' properties. However, correlations among items across different domains are ignored by ETL. As a result, it can learn items' embeddings only based on each extremely sparse single domain, which limits its performance.

4.3 Experiment on Sparse Datasets (RQ3)

To prove that our II-HGCN framework can address the data sparsity problem better than other CDR methods, we conduct exhaustive experiments on sparser datasets and the results are shown in Table 2. We choose the Music & Book dataset for testing since the sparsity of these two domains is relatively similar. For each domain, we randomly drop different ratios of interactions among users and items. To compare the performance of different models

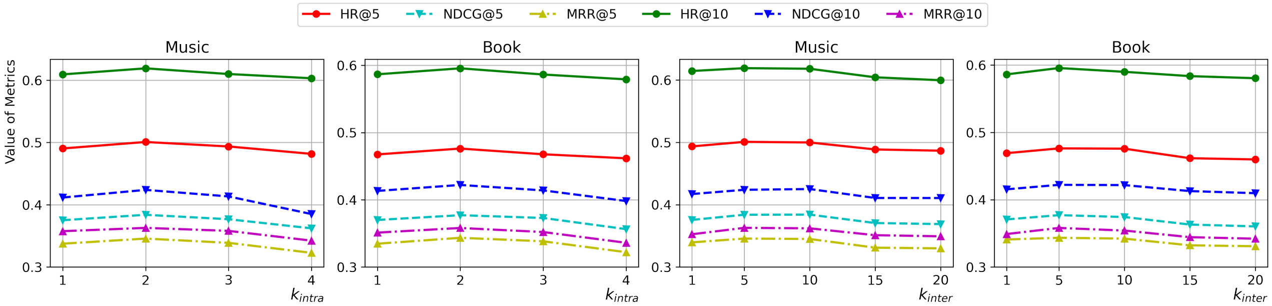


Figure 3: The effect of k_{intra} and k_{inter} on II-HGCN in the Music & Book dataset.

as data sparsity changes, we select drop ratios (i.e., sparse ratio) in $\{10\%, 20\%, 30\%, 40\%, 50\%\}$ as *Sparse Ratio* indicates in Table 2. Note that a higher sparse ratio means a sparser dataset and the sparse ratio of 0 indicates the original dataset. Since ETL is the strongest baseline, we choose it as the comparison method.

Overall, as the sparse ratio increases (the dataset becomes sparser), II-HGCN achieves greater improvement compared with ETL, which proves II-HGCN can better deal with the data sparsity problem. In particular, when we drop 50% interactions, II-HGCN improves ETL by as high as 39.92% and 35.24% on the music domain and the book domain respectively, in terms of NDCG@5. This is because ETL can only model the pairwise interactions among users and items. With the dataset becoming sparser, the loss of user-item interactions will lead to inaccurate representations of users and items, since relationships among users and items cannot be identified well. In addition, when both two domains become sparser, transferring knowledge across them may bring the negative transfer problem [34].

In contrast, II-HGCN can deal with the data sparsity problem better because of two reasons. *Firstly*, II-HGCN can explore high-order correlations in addition to pairwise relationships inside each single domain. Thus II-HGCN can maintain the high-order relationships as much as possible despite the dataset becoming sparser. *Secondly*, II-HGCN can still model the potential correlations among items across different domains in the inter-domain layer so that the knowledge transfer process for CDR can be stable. We give an example in **Appendix C.3** to explain why II-HGCN can maintain more correlations than other methods in a sparse dataset.

4.4 Ablation Study (RQ4)

We conduct ablation experiments to analyze the effect of combining embeddings of users and combining embeddings of items across different domains. Firstly, as Table 1, and Table 5 in **Appendix C.2** show, II-HGCN-S with only the intra-domain layer outperforms all single-domain methods, which proves that II-HGCN-S can generate more accurate embeddings in sparse domains by modeling high-order correlations among users and items. Besides, combining embeddings of only users (II-HGCN-U) or only items (II-HGCN-I) can enhance the recommendation performance compared with II-HGCN-S. Combining embeddings of users across domains is natural since users' behaviors in different domains can show their different features. While II-HGCN-I achieves better performance than II-HGCN-S proves that our inter-domain hypergraph structure can explore the high-order correlations among items in different domains and transfer useful knowledge of items across domains. Finally, II-HGCN achieves the best performance among these models with different combinations, illustrating that transferring knowledge of items and users are both necessary for CDR.

4.5 Impact of Hyperparameters (RQ5)

To answer **Q5**, we select the Music & Book dataset to analyze the effect of important hyperparameters on II-HGCN. Due to space constraints, we only show the effect of the most important hyperparameters of our model: k_{intra} and k_{inter} which influence the hypergraph structure of the intra-domain layer and the inter-domain layer. Their effect is depicted in Figure 3.

Effect of k_{intra}. Acroding to Fig. 3, all metrics peak at $k_{intra} = 2$ in both domains. When $k_{intra} = 1$, the hypergraph of the intra-domain layer degrades to the conventional graph structure. II-HGCN with $k_{intra} = 2$ outperforms that with $k_{intra} = 1$, because some solid high-order correlations among users and items can be explored and modeled, which compensates for the data sparsity problem in each single domain. However, the performance of II-HGCN becomes worse as k_{intra} further increases. The reason is that with higher-order hypergroups being added to the hypergraph in the intra-domain layer, some users or items with little correlations will also be connected in a hyperedge. Considering such too weak correlations will bring some noise to the modeling process.

Effect of k_{inter}. According to Fig. 3, our II-HGCN framework is insensitive to this hyperparameter. Though all metrics peak at $k_{inter} = 5$, the overall fluctuation is very limited. The reason is that we extract topk_{inter} similar items from another domain to construct a hyperedge for each target item. In each hyperedge, an item's weight is proportional to its hypersimilarity with the target item. Thus, the most similar items always have the greatest impact on the target item. With k_{inter} increases, some less similar items will also be added to the hyperedge, but they do not have a significant effect on the target item because of their small weights.

5 CONCLUSION

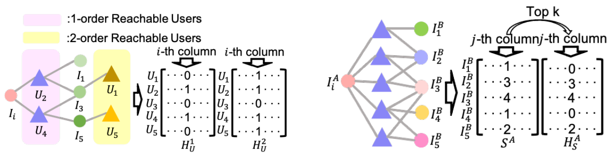
In this paper, we propose an Intra and Inter Domain HyperGraph Convolutional Network for Cross-Domain Recommendation, called II-HGCN, to solve the data sparsity problem. We design a novel intra-domain hypergraph structure to explore the high-order correlations in each sparse domain to generate more accurate embeddings. In addition, we also propose a hypergraph-based inter-domain framework to not only combine features of users but also transfer the knowledge of items across different domains. Comprehensive experiments show that our II-HGCN framework outperforms the state-of-the-art CDR methods and can achieve more significant improvement in sparser datasets, proving that our method can better deal with the data sparsity problem.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (No.62172362) and Leading Expert of "Ten Thousands Talent Program" of Zhejiang Province (No.2021R52001).

REFERENCES

[1] Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375* (2018).
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[3] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-domain mediation in collaborative filtering. In *International Conference on User Modeling*. Springer, 355–359.
[4] Alain Bretto. 2013. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer* (2013).
[5] Chaochao Chen, Huiwen Wu, Jiajie Su, Lingjuan Lyu, Xiaolin Zheng, and Li Wang. 2022. Differential Private Knowledge Transfer for Privacy-Preserving Cross-Domain Recommendation. In *Proceedings of the ACM Web Conference 2022*. 1455–1465.
[6] Hongxu Chen, Hongzhi Yin, Xiangguo Sun, Tong Chen, Bogdan Gabrys, and Katarzyna Musial. 2020. Multi-level graph convolutional networks for cross-platform anchor link prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1503–1511.
[7] Xu Chen, Ya Zhang, Ivor Tsang, Yuangang Pan, and Jingchao Su. 2020. Towards Equivalent Transformation of User Preferences in Cross Domain Recommendation. *arXiv preprint arXiv:2009.06884* (2020).
[8] Uthsav Chitra and Benjamin Raphael. 2019. Random walks on hypergraphs with edge-dependent vertex weights. In *International Conference on Machine Learning*. PMLR, 1172–1181.
[9] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web*. 278–288.
[10] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3558–3565.
[11] Ignacio Fernández-Tobías and Iván Cantador. 2014. Exploiting Social Tags in Matrix Factorization Models for Cross-domain Collaborative Filtering.. In *CBRecSys@ RecSys*. Citeseer, 34–41.
[12] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. 2013. Cross-domain recommendation via cluster-level latent factor model. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 161–176.
[13] Yue Gao, Meng Wang, Zheng-Jun Zha, Jialie Shen, Xuelong Li, and Xindong Wu. 2012. Visual-textual joint relevance learning for tag-based social image search. *IEEE Transactions on Image Processing* 22, 1 (2012), 363–376.
[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.
[15] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: a domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. *arXiv preprint arXiv:2105.03300* (2021).
[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
[17] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 667–676.
[18] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*. 595–606.
[19] Yuchi Huang, Qingshan Liu, and Dimitris Metaxas. 2009.] Video object segmentation by hypergraph cut. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 1738–1745.
[20] Yuchi Huang, Qingshan Liu, Shaoting Zhang, and Dimitris N Metaxas. 2010. Image retrieval via probabilistic hypergraph ranking. In *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 3376–3383.
[21] TaeHyun Hwang, Ze Tian, Rui Kuangy, and Jean-Pierre Kocher. 2008. Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 293–302.
[22] Il Im and Alexander Hars. 2007. Does a one-size recommendation system fit all? the effectiveness of collaborative filtering based recommendation systems across different domains and search modes. *ACM Transactions on Information Systems (TOIS)* 26, 1 (2007), 4–es.
[23] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual channel hypergraph collaborative filtering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020–2029.
[24] Heishiro Kanagawa, Hayato Kobayashi, Nobuyuki Shimizu, Yukihiro Tagami, and Taiji Suzuki. 2019. Cross-domain recommendation via deep domain adaptation. In *European Conference on Information Retrieval*. Springer, 20–29.
[25] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[26] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Twenty-First international joint conference on artificial intelligence*.
[27] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th annual international conference on machine learning*. 617–624.
[28] Pan Li and Alexander Tuzhilin. 2020. Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 331–339.
[29] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: a content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th international conference on World Wide Web companion*. 817–818.
[30] Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. 2022. Collaborative Filtering with Attribution Alignment for Review-based Non-overlapped Cross Domain Recommendation. In *Proceedings of the ACM Web Conference 2022*. 1181–1190.
[31] Weiming Liu, Xiaolin Zheng, Jiajie Su, Longfei Zheng, Chaochao Chen, and Mengling Hu. 2023. Contrastive Proxy Kernel Stein Path Alignment for Cross-Domain Cold-Start Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).
[32] Nima Mirbakhsh and Charles X Ling. 2015. Improving top-n recommendation for cold-start users via cross-domain information. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 9, 4 (2015), 1–19.
[33] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems* 20 (2007).
[34] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.
[35] Reuven Y Rubinstein and Dirk P Kroese. 2004. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. Vol. 133. Springer.
[36] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 650–658.
[37] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
[38] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4503–4511.
[39] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*. 413–424.
[40] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. *arXiv preprint arXiv:1905.10760* (2019).
[41] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-domain recommendation via preference propagation graphnet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2165–2168.
[42] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems* 19 (2006).
[43] Feng Zhu, Yan Wang, Jun Zhou, Chaochao Chen, Longfei Li, and Guanfeng Liu. 2023. A Unified Framework for Cross-Domain and Cross-System Recommendations. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2023), 1171–1184. https://doi.org/10.1109/TKDE.2021.3104873



(a) example of intra-domain layer (b) example of inter-domain layer

Figure 4: (a) and (b) show examples of the hypergraph construction in the intra-domain layer and the inter-domain layer

Table 3: Notations of this paper

Notation	Description
Notations throughout the entire process	
$\mathcal{U}$	the user set in domain A and domain B
A and B	two different domains
$\mathcal{I}^A$ and $\mathcal{I}^B$	the item sets in domain A and domain B
N	the number of users
M	the number of items in domain A
T	the number of items in domain B
$\mathbf{R}^A$ and $\mathbf{R}^B$	the user-item interaction matrices in domain A and domain B
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	the representation of a hypergraph
$\mathbf{H}$	the adjacency matrix of a hypergraph
Notations throughout the intra-domain process	
$\mathcal{E}^k_{\mathcal{U}}$ and $\mathcal{E}^k_{\mathcal{I}}$	the k-order reachable user set and item set
$\mathbf{H}^k_{\mathcal{U}}$ and $\mathbf{H}^k_{\mathcal{I}}$	the k-order user hypergroup and item hypergroup
$\mathbf{H}_{\mathcal{U}}$ and $\mathbf{H}_{\mathcal{I}}$	the user hypergraph and item hypergraph
$\mathbf{D}^k_{\mathcal{U}}$ and $\mathbf{D}^k_{\mathcal{I}}$	the hyperdegree matrices of hyperedges in k-order hypergroups for users and items
$\mathbf{D}_{\mathcal{U}_v}$ and $\mathbf{D}_{\mathcal{U}_e}$	the vertex degrees matrix and the hyperedge degrees matrix of the user hypergraph
$\mathbf{D}_{\mathcal{I}_v}$ and $\mathbf{D}_{\mathcal{I}_e}$	the vertex degrees matrix and the hyperedge degrees matrix of the item hypergraph
$\mathbf{E}_{\mathcal{U}_A}$ and $\mathbf{E}_{\mathcal{I}_A}$	embeddings of users and items generated from the input layer for domain A.
$\mathbf{E}_{\mathcal{U}_B}$ and $\mathbf{E}_{\mathcal{I}_B}$	embeddings of users and items generated from the input layer for domain B.
$\mathbf{E}^l_{\mathcal{U}}$ and $\mathbf{E}^l_{\mathcal{I}}$	the output user embeddings and item embeddings from the l-th layer of the hypergraph convolution network
Notations throughout the inter-domain process	
$\mathbf{S}^A$ and $\mathbf{S}^b$	the hypersimilarity matrices for items in domain A and domain B
$\mathbf{H}^A_S$ and $\mathbf{H}^B_S$	the hypergraphs constructed for items in domain A and domain B
$\mathbf{P}^l_{\mathcal{I}_A}$ and $\mathbf{P}^l_{\mathcal{I}_B}$	the aggregated embeddings for items in domain A and domain B obtained from hypergraph convolutional network in the inter-domain process
$\tilde{\mathbf{E}}^l_{\mathcal{U}_A}$ and $\tilde{\mathbf{E}}^l_{\mathcal{U}_B}$	the l-th layer combined embeddings of users obtained by element-wise mechanism.
$\tilde{\mathbf{E}}^l_{\mathcal{I}_A}$ and $\tilde{\mathbf{E}}^l_{\mathcal{I}_B}$	the l-th layer combined embeddings of items obtained by element-wise mechanism.
$\tilde{\mathbf{E}}_{\mathcal{U}_A}$ and $\tilde{\mathbf{E}}_{\mathcal{U}_B}$	the final combined embeddings of users in domain A and domain B
$\tilde{\mathbf{E}}_{\mathcal{I}_A}$ and $\tilde{\mathbf{E}}_{\mathcal{I}_B}$	the final combined embeddings of items in domain A and domain B

Table 4: The statistics of experimental datasets

Dataset	Users	Domain	Items	Interactions	Density
Movie & Book	29,476	Movie	24,091	591,258	0.08%
		Book	41,884	579,131	0.05%
Movie & Music	15,914	Movie	17,794	416,228	0.14%
		Music	20,058	280,398	0.09%
Music & Book	16,267	Music	18,467	233,251	0.08%
		Book	23,988	291,325	0.07%

A NOTATIONS

For clearly describe the details of II-HGCN, we list the notations throughout the entire process, the intra-domain process and the inter-domain process in Table 3.

B MORE MODELING DETAILS

B.1 Example of The Hypergraph Construction in The Intra-Domain Layer

We give an example of the hypergraph construction in the intra-domain layer in Figure 4(a). Each column in $\mathbf{H}^1_{\mathcal{U}}$ and $\mathbf{H}^2_{\mathcal{U}}$ represents a hyperedge containing several users. U_2 and U_4 are 1-order reachable users for I_i, thus their values are set to 1 in the i-th column of $\mathbf{H}^1_{\mathcal{U}}$. Note that $(k-1)$-order reachable users are also included in the k-order reachable user set for an item, and thus the values of U_2 and U_4 are also set to 1 in addition to U_1 and U_5 in the i-th column of $\mathbf{H}^2_{\mathcal{U}}$.

B.2 Modeling Process of Items in The Intra-Domain Layer

HyperGraph Construction for Items. Firstly we give the definition of User's k-order Reachable Items as follows:

Definition B.1 (User's k-order Reachable Items). In a user-item bipartite graph, I_j is k-order reachable from U_i if at least one of the direct paths between I_j and U_i exist k or less than k items.

Then the k-order item hypergroup $\mathbf{H}^k_{\mathcal{I}} \in \{0,1\}^{M\times N}$ can be formulated as:

$$\mathbf{H}^k_{\mathcal{I}} = \mathbf{R}^T \cdot min(1, pow(\mathbf{R}\cdot\mathbf{R}^T, k-1)). \tag{13}$$

Finally, the item hypergraph can be constructed as:

$$\mathbf{H}_{\mathcal{I}} = \mathbf{H}^1_{\mathcal{I}} || \mathbf{H}^2_{\mathcal{I}} \cdot \mathbf{F}^2_{\mathcal{I}} || \dots || \mathbf{H}^{k_{intra}}_{\mathcal{I}} \cdot \mathbf{F}^{k_{intra}}_{\mathcal{I}}, \tag{14}$$

where $\mathbf{F}^k_{\mathcal{I}} \in \{0,1\}^{N\times N}$ can be calculated in the same way as which of users.

HyperGraph Convolution. Embeddings of items can be updated based on the hypergraph constructed above:

$$\mathbf{E}^{l+1}_{\mathcal{I}} = \delta(\mathbf{D}^{-1/2}_{\mathcal{I}_v}\mathbf{H}_{\mathcal{I}}\mathbf{D}_{\mathcal{I}_e}\mathbf{H}^T_{\mathcal{I}}\mathbf{D}^{-1/2}_{\mathcal{I}_v}\mathbf{E}^l_{\mathcal{I}}\mathbf{W}^l + \mathbf{E}^l_{\mathcal{I}}). \tag{15}$$

B.3 Modeling Process of Domain B in The Inter-Domain Layer

Combine Embeddings of Users. We also use the element-wise attention mechanism to transfer knowledge of users from domain A to domain B:

$$\tilde{\mathbf{E}}^l_{\mathcal{U}_B} = \mathbf{E}^l_{\mathcal{U}_B} \odot \mathbf{W}^l_{\mathcal{U}_B} + \mathbf{E}^l_{\mathcal{U}_A} \odot (1 - \mathbf{W}^l_{\mathcal{U}_B}). \tag{16}$$

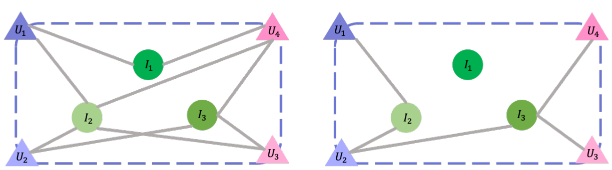
Combine Embeddings of Items. For domain B, we firstly construct the hypersimilarity matrix $\mathbf{S}^B \in \mathbb{R}^{M\times T}$ as $\mathbf{S}^B_{ij} = \sum(\mathbf{R}^A_{*i} \circledast \mathbf{R}^B_{*j})$. Then we can calculate the hypergraph $\mathbf{H}^B_S \in \mathbb{R}^{M\times T}$ as $(\mathbf{H}^B_S)_{*i} = topK(\mathbf{S}^B_{*i}, k)$. Based on the hypergraph, we use an adjusted hypergraph convolution network to transfer knowledge of items from domain A to domain B:

$$\mathbf{P}^l_{\mathcal{I}_B} = \delta(\mathbf{D}^{-1/2}_{\mathbf{H}^B_{S v}}\mathbf{H}^B_S\mathbf{D}^{-1/2}_{\mathbf{H}^B_{S e}}\mathbf{E}^l_{\mathcal{I}_A}). \tag{17}$$

Table 5: Experiment on Movie & Music and Music & Book datasets

Movie & Music												
topK	topK = 5						topK = 10					
Domain	Movie			Music			Movie			Music		
Metrics	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR
PMF	0.3797	0.2771	0.2433	0.3611	0.2677	0.2369	0.4991	0.3144	0.2586	0.4672	0.3020	0.2510
NCF	0.4171	0.3012	0.2630	0.4354	0.3210	0.2835	0.5482	0.3335	0.2803	0.5433	0.3587	0.2990
NGCF	0.4177	0.3089	0.2664	0.4234	0.3042	0.2651	0.5576	0.3443	0.2852	0.5557	0.3464	0.2826
CoNet	0.3763	0.2618	0.2241	0.3736	0.2576	0.2194	0.5185	0.3077	0.2430	0.5173	0.3039	0.2385
DDTCDR	0.3943	0.2736	0.2405	0.3975	0.2703	0.2216	0.5104	0.3141	0.2571	0.4969	0.2956	0.2261
DARec	0.4468	0.3199	0.2786	0.4521	0.3446	0.2867	0.5559	0.3351	0.2672	0.5879	0.3874	0.3238
ETL	0.4844*	0.3629*	0.3217*	0.5183*	0.3992*	0.3596*	0.6247*	0.4066*	0.3398*	0.6479*	0.4410*	0.3769*
II-HGCN-S	0.4303	0.3283	0.2789	0.4502	0.3381	0.2979	0.5542	0.3639	0.2977	0.6092	0.3926	0.3163
II-HGCN-I	0.4790	0.3584	0.3050	0.5187	0.3889	0.3564	0.6018	0.3835	0.3236	0.6473	0.4332	0.3645
II-HGCN-U	0.4906	0.3694	0.3161	0.5300	0.4032	0.3650	0.6173	0.3961	0.3254	0.6559	0.4497	0.3701
II-HGCN	**0.5104**	**0.3737**	**0.3271**	**0.5528**	**0.4214**	**0.3776**	**0.6519**	**0.4171**	**0.3458**	**0.6828**	**0.4675**	**0.3996**
Improv.	5.37%	2.98%	1.68%	6.66%	5.56%	5.01%	4.35%	2.58%	1.77%	5.39%	6.01%	6.02%
Music & Book												
topK	topK = 5						topK = 10					
Domain	Music			Book			Music			Book		
Metrics	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR
PMF	0.3035	0.2216	0.1947	0.2949	0.2135	0.1867	0.4119	0.2566	0.2091	0.3937	0.2448	0.1996
NCF	0.4062	0.3007	0.2658	0.3494	0.2579	0.2281	0.5236	0.3386	0.2814	0.4564	0.2926	0.2423
NGCF	0.3665	0.2618	0.2278	0.3343	0.2374	0.2058	0.4954	0.3031	0.2448	0.4538	0.2762	0.2217
CoNet	0.3259	0.2124	0.2016	0.3171	0.1910	0.2026	0.4517	0.2597	0.2379	0.4432	0.2317	0.2325
DDTCDR	0.4017	0.3162	0.2781	0.3748	0.2907	0.2864	0.4969	0.3333	0.2834	0.4768	0.3268	0.2812
DARec	0.4630*	0.3497	0.3251*	0.4438	0.3391	0.2984	0.5727	0.3807	0.3219	0.5538	0.3776	0.3243
ETL	0.4628	0.3572*	0.3225	0.4594*	0.3587*	0.3203*	0.5867*	0.3972*	0.3390*	0.5728*	0.3954*	0.3357*
II-HGCN-S	0.4379	0.3255	0.2900	0.4050	0.3020	0.2682	0.5618	0.3600	0.2978	0.5293	0.3424	0.2750
II-HGCN-I	0.4754	0.3548	0.3151	0.4486	0.3470	0.3070	0.6166	0.4075	0.3357	0.5671	0.3948	0.3324
II-HGCN-U	0.4806	0.3662	0.3284	0.4639	0.3474	0.3154	0.6235	0.4159	0.3448	0.5773	0.4031	0.3407
II-HGCN	**0.5008**	**0.3839**	**0.3457**	**0.4762**	**0.3771**	**0.3435**	**0.6190**	**0.4239**	**0.3629**	**0.5955**	**0.4221**	**0.3581**
Improv.	8.16%	7.47%	6.33%	3.66%	5.13%	7.24%	5.51%	6.72%	7.05%	3.96%	6.75%	6.67%

Note that the results marked with * are the best performing baselines.



(a) The original dataset. (b) The dataset after sparsifying.

Figure 5: (a) and (b) show examples of the correlations among users and items *before* and *after* a dataset becomes sparser.

The embeddings learned from the intra-domain layer and the inter-domain layer are also combined by the element-wise attention mechanism:

$$\tilde{\mathbf{E}}^{l}_{\mathcal{I}_B} = \mathbf{E}^{l}_{\mathcal{I}_B} \odot \mathbf{W}^{l}_{\mathcal{I}_B} + \mathbf{P}^{l}_{\mathcal{I}_B} \odot (1 - \mathbf{W}^{l}_{\mathcal{I}_B}). \tag{18}$$

Finally, we can generate the final embeddings of users and items of domain B:

$$\tilde{\mathbf{E}}_{\mathcal{U}_B} = \tilde{\mathbf{E}}^{1}_{\mathcal{U}_B} || \tilde{\mathbf{E}}^{2}_{\mathcal{U}_B} || \dots || \tilde{\mathbf{E}}^{L}_{\mathcal{U}_B}, \; \tilde{\mathbf{E}}_{\mathcal{I}_B} = \tilde{\mathbf{E}}^{1}_{\mathcal{I}_B} || \tilde{\mathbf{E}}^{2}_{\mathcal{I}_B} || \dots || \tilde{\mathbf{E}}^{L}_{\mathcal{I}_B}. \tag{19}$$

B.4 Example of The Hypergraph Construction in The Inter-Domain Layer

Figure 4(b) shows the construction example of $\mathbf{H}^{A}_{S}$, where we assume $k_{inter} = 3$. For example, there are three users who interact with both I^{A}_{j} and I^{B}_{2}, thus the hypersimilarity of I^{B}_{2} and I^{A}_{j} is 3. By extracting top3 most similar items I^{B}_{2}, I^{B}_{3}, and I^{B}_{5}, we can construct the j-th hyperedge in $\mathbf{H}^{A}_{S}$. Then the embedding of I^{A}_{j} will be updated based on this hyperedge.

C MORE EXPERIMENTAL DETAILS

C.1 The Statistics of Experimental Datasets

We list the statistics of experimental datasets in Table 4.

C.2 Experimental Results on Movie & Music and Music & Book Dataset

We list the experiment results on Movie & Music and Music & Book datasets in Table 5.

C.3 Example of Why II-HGCN Can Deal with The Data Sparsity Problem Better

We give an example in Figure 5 to show why II-HGCN can perform better than other methods in a sparse dataset. Solid lines indicate the pairwise user-item interactions and the dotted line indicates the hyperedge. In this example, when the dataset becomes sparser, the degree of message passing between U_1 and U_4 becomes so high that the correlation between these two users is nearly destroyed in conventional pairwise relationships. However, the high-order correlations among U_1, U_2, U_3, and U_4 can still be maintained since they are included in 2-order reachable user set of I_2. Thus, when the dataset becomes sparser, II-HGCN can maintain more correlations among users and items, compared with traditional CDR methods which based on pairwise user-item interactions.