Heterogeneous Graph Transfer Learning for Category-aware Cross-Domain Sequential Recommendation

Anonymous Author(s)*

Abstract

Cross-domain sequential recommendation (CDSR) is proposed to alleviate the data sparsity issue while capturing users' sequential preferences. However, most existing methods do not explore the item transition patterns across different domains and can also not be applied to a multi-domain scenario. Moreover, previous methods rely on overlapping users as bridges to transfer knowledge, which struggles to capture the complex associations across domains without sufficient overlapping users. In this paper, we introduce item attributes into CDSR, and propose a heterogeneous graph transfer learning method to address these issues. Specifically, we construct a cross-domain heterogeneous graph to allow the association of user, item, and category nodes from different domains, and enhance the flexibility of the model by enabling message propagation between more nodes through edge expansion based on the semantic similarity and co-occurrence probability. In addition, we devise meta-paths from different perspectives for nodes at item, user and category levels to guide information aggregation, which can transfer knowledge across domains and reduce the reliance on the number of overlapping users. We further design attention modules to capture users' dynamic preferences from the item sequences they have interacted with in each domain, and explore the transition patterns within category sequences which reflect users' coarse-grained preferences. Finally, we perform knowledge transfer across different domains, and predict the most likely items that users will interact with in each domain. Extensive empirical studies on three real-world datasets indicate that our HGTL significantly outperforms the state-of-the-art baselines in all cases. The source codes of our HGTL and the datasets are available at https://anonymous.4open.science/r/HGTL-C135.

CCS Concepts

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

Keywords

Cross-domain recommendation, Sequential recommendation, Heterogeneous graph, Transfer learning

ACM Reference Format:

Anonymous Author(s). 2025. Heterogeneous Graph Transfer Learning for Category-aware Cross-Domain Sequential Recommendation. In *Proceedings of the ACM Web Conference 2025 (WWW '25)), 28 April - 2 May, 2025, Sydney, Australia.* ACM, New York, NY, USA, 12 pages.

WWW '25, 28 April - 2 May, 2025, Sydney, Australia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

Figure 1: Illustration of category-aware cross-domain sequential recommendation.

1 Introduction

Cross-domain sequential recommendation (CDSR) [6, 18, 25, 28, 29, 37] combines sequential recommendation (SR) [7, 10, 12, 16, 45] with cross-domain recommendation (CDR) [5, 8, 23, 40, 42] by introducing some extra source-domain data, aiming to capture item transition patterns while alleviating the sparsity problem. As shown in Fig. 1, taking the "Movie" and "Book" domains as an example, the user's interaction with the book "Murder on the Orient Express" is mainly due to the fact that the user has watched the movie based on the book, whereas it's difficult to account for the transition between the books "Oliver Twist" and "Murder on the Orient Express" in the "Book" domain alone. In CDSR, a common approach to associate two separated domains is to learn user features in different domains separately, and perform knowledge transfer between them to obtain comprehensive user preferences [1]. However, such an approach relies on the same users in different domains as a bridge for knowledge transfer, whereas the overlapping users may only account for a small fraction [24, 44]. Therefore, information from the item aspect has the potential to serve as a complementary way to bridge two domains.

In SR, categories are commonly used as an auxiliary attribute [3, 22, 36] since the transition relationships between different categories in a user's sequence also reveal coarse-grained transition patterns. However, most existing methods only consider scenarios where a single item corresponds to a single category, whereas in practical applications, a single item often corresponds to multiple categories. Moreover, there are also some complicated relationships between items and categories across different domains, as shown in Fig. 1 the category "Suspense" preferred by the user in both domains can be utilized to reveal the relationships among some items from two domains.

To address the above challenges, in this paper we study a new and emerging problem, i.e., category-aware cross-domain sequential recommendation (CCDSR). Our goal is to introduce category information into CDSR, which can reflect users' intra- and inter-domain preferences over a period of time.

In CCDSR, we not only face the challenges existing in CDSR and attribute-aware SR, but also need to consider the category-category

relationships and category-item relationships across different domains, as well as how to transfer knowledge of category features and sequential features between domains. Furthermore, due to the domain-specific characteristics in cross-domain scenarios, the introduction of excessive additional information may result in noise and the negative transfer problem. Moreover, in real-world applications, the proportion of overlapping users is relatively small compared to the total user population. For non-overlapping users, although their behavioral sequences only exist in a single domain, this part of data is also valuable for the model to learn from.

To tackle the above issues, we propose a novel solution named heterogeneous graph transfer learning (HGTL). Specifically, we construct a cross-domain heterogeneous graph to connect items from different domains via user or category associations, which reduces the reliance on overlapping users as the only bridges between different domains. In order to obtain richer contextual information for nodes and achieve knowledge transfer across different domains, we devise meta-paths from three different perspectives for nodes at item, user and category levels, and adopt contrastive mechanisms to jointly learn complementary information from different meta-paths. Moreover, we further enhance the correlation between different domains through edge expansion based on the semantic similarity and co-occurrence probability, which enable message propagation between more nodes. For a user's behavior sequence in each domain, we employ the attention mechanisms to capture the item-based sequential preferences. Furthermore, we design an item-category attention layer to measure the importance of multiple categories for a given item, and explore the category-based sequential preferences that reflect coarse-grained transition patterns within the sequence. Finally, we aggregate the information from different domains via some transfer units and predict the most likely items that users will interact with in each domain. It is worth mentioning that our model can be applied to a multi-domain scenario by expanding the heterogeneous graph and including more sequential preference learning modules, for which we have also conducted experiments.

- We introduce the category information into CDSR and define a new problem, for which we propose a novel solution named heterogeneous graph transfer learning (HGTL).
- We associate nodes from different domains by constructing a new cross-domain heterogeneous graph, and design a novel meta-path guided node representation learning module for knowledge transfer across domains, which can reduce the reliance on the number of overlapping users.
- We design a novel category-aware sequential preference learning module to capture the item-based and category-based preferences of users in each domain, and two specific transfer units to adaptively aggregate the information from different domains.
- We conduct extensive empirical studies on three real-world datasets, where the results show that our HGTL significantly outperforms the state-of-the-art baselines in all cases.

2 Related Work

Cross-Domain Sequential Recommendation. π-Net [26] is one of the earliest works for CDSR, which employs GRUs to capture and share the sequential information in different domains. CDSASRec [1] is an improved version of SASRec [14] implemented for CDSR, which integrates the users' preferences in the source domain into the target domain. DA-GCN [9] constructs a domain-aware graph to capture the explicit structural information and associations among items of different domains. Recently, there are some methods that utilize the contrastive mechanism to enhance the capability of model representation [4, 38, 41]. For example, C^{2}DSR [4] proposes a framework that incorporates two sequential objectives with a contrastive objective, to jointly learn the single-domain and cross-domain user representations. MGCL [38] adopts the contrastive mechanism in an intra-domain item representation view and an inter-domain user preference view. However, these methods neglect to explore the rich information of item attributes, and rely on the number of overlapping users as bridges to transfer knowledge. Moreover, none of the above methods consider the multi-domain recommendation scenario, i.e., utilizing data from more than two domains and simultaneously improving the performance of multiple domains, which is more common and more challenging in real-world applications.

Attribute-Aware Sequential Recommendation. Efficiently utilizing rich side information to learn higher-quality item representations has become a popular research topic in sequential recommendation. CaFe [17] employs an attention mechanism to learn user intent from item attributes and uses it as a prior knowledge to guide item representation learning. NOVA [21] devises a non-invasive attention mechanism to learn a better distribution of attention instead of fusing side information directly into the item representations. Moreover, there are some methods focusing on modeling category information [3, 22, 36] since categories not only provide rich contextual information, but also reveal coarse-grained sequence transition patterns. For instance, CoCoRec [3] utilizes item categories to retrieve users with similar preferences, enhancing collaborative learning among users. Although these methods have been shown effective, none of them introduces cross-domain data and may still suffer from the data sparsity problem. While in the cross-domain scenario, there are also some complicated relationships between items and categories across different domains, which is out of consideration in traditional single-domain methods, resulting in the inability to associate multiple domains and perform knowledge transfer. Furthermore, most existing works focus on the problem in which an item only belongs to one single category, which cannot be applied to a more realistic setting that an item belongs to one or more categories.

3 Proposed Method

In this section, we formally define the CCDSR task and introduce the components of our model in detail. Notice that for the sake of brevity, we take two domains as an example, and our model can be extended to a multi-domain scenario, for which we conduct empirical studies in Section 4.

3.1 Problem Definition

Let $\mathcal{U}$ denote the set of users, for each user $u \in \mathcal{U}$, we represent two single-domain behavior sequences (ordered by the interaction time) of a user as $S^A = \{A_1, A_2, \ldots, A_L\}$ and $S^B = \{B_1, B_2, \ldots, B_L\}$, where A_i denotes an interaction at the time step i in domain A. L denotes the maximum length of a sequence from a single domain.

If the length of the sequence is less than L, a padding item will be repeatedly appended at the beginning of the sequence.

We employ categories as the auxiliary information of items and denote the set of categories as C. Since an item may belong to more than one category, we use $A_i = \{v_i^A, c_i^{(1)}, \ldots, c_i^{(l)}\}$ to denote the ith interaction of the user, where v_i^A represents the item ID and $c_i^{(j)}$ represents the jth category of the interacted item i. Given a user's behavior sequences S^A and S^B, the goal of CCDSR is to predict the next preferred item v_{i+1}^A in domain A and v_{i+1}^B in domain B.

3.2 Heterogeneous Graph Construction

3.2.1 Heterogeneous graph A heterogeneous graph [2, 42] is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, and it is associated with a node type function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and an edge type function $\varphi : \mathcal{E} \rightarrow \mathcal{R}$, where $\mathcal{A}$ and $\mathcal{R}$ denote sets of predefined node types and relation types, respectively, having constraints $|\mathcal{A}| + |\mathcal{R}| > 2$.

As shown in Fig. 2(a), the heterogeneous graph contains three types of nodes, including the user node u, the item node v and the category node c, where the item node can be subdivided into A-domain item node v^A and B-domain item node v^B. Meanwhile, there are two types of relations. The edge between a user node and an item node represents that the user interacted with the item, and the edge between an item node and a category node represents that the item belongs to that category.

By constructing a cross-domain heterogeneous graph, item nodes in different domains can be associated, enabling rich knowledge transfer between domains. It is worth mentioning that in traditional cross-domain recommendation, it usually solely depends on the overlapping users to bridge items from different domains, whereas in our heterogeneous graph, items of different domains can also be associated by common categories, which makes the performance no longer severely rely the number of overlapping users.

3.2.2 Meta-path In a heterogeneous graph, two nodes can be linked by different meta-paths [13]. A meta-path ρ is defined as a path in the form of $\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_{n-1}} \mathcal{A}_n$, which describes a composite relation $\mathcal{R}_1 \circ \mathcal{R}_2 \circ \cdots \circ \mathcal{R}_{n-1}$ between a start node type $\mathcal{A}_1$ and an end node type $\mathcal{A}_n$. We use $\circ$ to denote the composition operator on relations. By utilizing the designed meta-paths, we can identify pairs of entities in the heterogeneous graph that have a certain association, despite being distant from each other.

As shown in Fig. 2(a), at the item level, two item nodes can be linked by different meta-paths, such as item-user-item (I-U-I) and item-category-item (I-C-I). For the design of meta-paths in the context of a real-world scenario, different meta-paths often reveal different semantics, e.g., I-U-I represents that two items have been purchased by a same user, indicating a certain connection between them. I-C-I represents two items belong to a same category, suggesting that they share some similar features. We also design user-level and category-level meta-paths. At the user level, we have user-item-user (U-I-U) and user-item-category-item-user (U-I-C-I-U), where the former connects two users who have purchased a same item, and the latter connects two users who prefer a same category of items. At the category level, we have category-item-category (C-I-C), which associates different categories of a same item, and category-item-user-item-category (C-I-U-I-C), which connects two categories preferred by a same user.

3.2.3 Edge expansion Considering that in real-world scenarios, there are some similar categories, and additionally there may be associations between different categories. It would be helpful to link such category nodes by some extra edges when constructing the graph, so that the items can obtain useful information from more categories, to alleviate the problem of a small number of overlapping users. We measure the correlation between two category nodes in terms of the semantic similarity and the co-occurrence patterns, and link two nodes with a strong correlation via an extended edge.

Semantic similarity-based correlation. In most real-world datasets, there are categories which may be semantically similar but not literally identical, such as "Fantasy Films" and "Magical Films". Due to the limitation of data collection, items that should belong to several categories may not be fully tagged. To address this issue, we utilize a pre-trained BERT model for semantic encoding to calculate the similarity between any two categories and link those with a high similarity. In this paper, we employ cosine similarity as a measure of semantic similarity, which can be defined as follows:

$$sim(c_i, c_j) = cos(\boldsymbol{h}_{c_i}, \boldsymbol{h}_{c_j}) = \frac{\boldsymbol{h}_{c_i}^T \boldsymbol{h}_{c_j}}{\|\boldsymbol{h}_{c_i}\| \|\boldsymbol{h}_{c_j}\|}, \tag{1}$$

where $\boldsymbol{h}_{c_i} \in \mathbb{R}^d$ denotes the embedding of category c_i, and is initialized by the pre-trained BERT model.

Co-occurrence pattern-based correlation. On the other hand, in real-world scenarios, there exists a certain dependency between different categories, where some category often appears together with the presence of another category. For example, "Thriller Films" often appears alongside "Suspense Films", indicating a strong connection between them. To further explore and capture these important dependencies, we define their correlation by mining co-occurrence patterns of categories in the data. Specifically, we model the category dependency in the form of a conditional probability, denoted as $P_{i,j} = P(c_j|c_i)$, which represents the probability of category c_j appearing when category c_i is present. We use N_i to represent the number of occurrences of c_i in the dataset and M_{ij} to represent the number of co-occurrences of c_i and c_j, and obtain the conditional probability as follows:

$$P_{ij} = P(c_j|c_i) = \frac{\boldsymbol{M}_{ij}}{\boldsymbol{N}_i}. \tag{2}$$

Finally, we determine whether to associate category c_i with category c_j based on the semantic similarity and co-occurrence probability between them:

$$\delta_{ij} = \begin{cases} 1, & sim(c_i, c_j) \geq \tau_1 \quad or \quad P_{ij} \geq \tau_2 \\ 0, & otherwise, \end{cases} \tag{3}$$

where τ_1 and τ_2 are two constants representing the thresholds for edge expansion. When δ_{ij} is 1, we construct an extended edge between node c_i and node c_j.

Notice that when a meta-path passes through a category node, we allow it to be connected to other category nodes via some extended edges, e.g., I-C-C-I. For instance, as shown in Fig. 2(a), item node v_1^A can be directly connected to another item node v_2^A via the category node c_1. And if we allow a category node to be connected to other category nodes via extended edges, item node v_1^A could

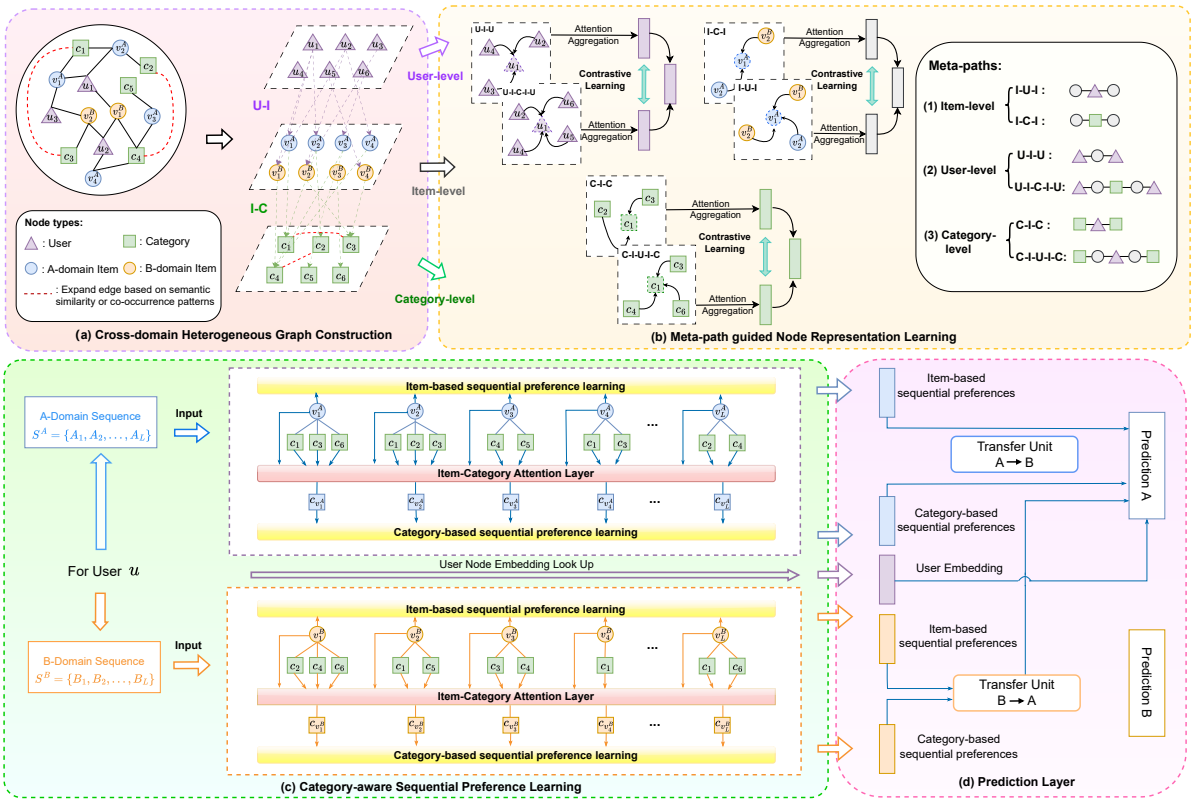


Figure 2: The framework of our proposed HGTL (best viewed in color). We first construct a cross-domain heterogeneous graph that connects users, items, and category nodes, and enhances their correlation via edge expansion. Then, we design different meta-paths based on node types to aggregate information from their neighbors and employ contrastive learning to complement each other. For each user, we input their behavior sequences in each domain into the category-aware sequential preferences learning module to obtain item-based and category-based sequential preferences, and share the preferences from different domains via the transfer unit. Notice that our HGTL can be easily expanded to a multi-domain version.

also be connected to category node c_3 through category node c_1, and eventually find another item node v_2^B. With edge expansion, we can further explore the important dependencies between categories, enabling item nodes to find their neighboring nodes through more useful meta-paths. On the other hand, two domains with weak association may limit the flexibility of the model due to fewer common categories. By expanding the edges, we could allow more categories to be associated with each other. Furthermore, constructing more category bridges is also beneficial to alleviate the problem of few overlapping users.

3.3 Meta-path Guided Node Representation Learning

3.3.1 Meta-path based neighbors As shown in Fig. 2(b), given a start node i and a type of meta-path, the meta-path-based neighbors $N_\rho(i)$ of node i is defined as the set of end nodes connected to node i via meta-path ρ. It is obvious that for the same start node, employing different types of meta-paths will lead to different sets of neighboring nodes. Notice that since the number of neighbors may be large, to improve model efficiency and effectiveness, we only take the top K nodes with the largest number of connections (most relevant) as the set of neighbors for each type of meta-path.

3.3.2 Meta-path guided neighbor aggregation By constructing meta-paths, node i can identify nodes of a same type that are associated with it. Our goal is to propagate messages between the node and its neighboring node set, enabling it to gather richer contextual information and achieve knowledge transfer across different domains. Since different nodes in the neighboring node set have different influences on node i, we adopt an attention aggregation approach to aggregate the information of these nodes.

$$\boldsymbol{h}_i^\rho = \sum_{k \in N_\rho(i)} a_k \boldsymbol{h}_k, \tag{4}$$

$$a_k = \frac{\exp\left(\sigma(\boldsymbol{h}_i{}^T \boldsymbol{h}_k)\right)}{\sum_{k' \in N_\rho(i)} \exp\left(\sigma\left(\boldsymbol{h}_i^T \boldsymbol{h}_{k'}\right)\right)}, \tag{5}$$

where a_k is the weight assigned to $\boldsymbol{h}_k$, representing the importance of the neighboring node k to the target node i, and $\sigma(\cdot)$ is the sigmoid function.

3.3.3 Meta-path based contrastive learning According to Section 3.2.2, for either the item, user or category level, we adopt two different perspectives and design two different meta-paths to explore and aggregate its neighboring nodes. To jointly learn the

complementary information from different sets of neighbors, we apply a contrastive mechanism to the node embeddings that have been aggregated via two different meta-paths. We conduct a contrastive task to maximize the mutual information between positive pairs while minimizing the agreement between negative samples.

Specifically, we consider the aggregated node embeddings (i.e., $\boldsymbol{h}_i^{\rho}$ and $\boldsymbol{h}_i^{\rho'}$) of a same node i via two different meta-paths ρ and ρ' of a same type level as a positive pair. Then, we naturally regard the representations of different nodes (i.e., $\boldsymbol{h}_i^{\rho}$ and $\boldsymbol{h}_j^{\rho'}$) as a negative pair. We employ InfoNCE [31, 32] with a standard binary cross-entropy loss form for optimization. The loss function can be formalized as:

$$\mathcal{L}_{CL} = -\sum_i \log(\sigma(D(\boldsymbol{h}_i^{\rho}, \boldsymbol{h}_i^{\rho'}))) + \log(\sigma(1 - D(\boldsymbol{h}_i^{\rho}, \boldsymbol{h}_j^{\rho'}))), \quad (6)$$

where $D(\cdot)$ is a dot product to measure the similarity between two node representations. For each node at the item, user or category level, we will calculate its corresponding contrastive loss.

3.4 Category-Aware Sequential Preference Learning

3.4.1 Item-based sequential preference learning

Heterogeneous graph embedding aggregation. As discussed in Section 3.2.2, the aggregation of one node at the same level via two different meta-paths results in two different representations, which contain diverse information. To obtain the suitable node representation, we employ mean pooling to aggregate them:

$$\boldsymbol{h}_i = \frac{\boldsymbol{h}_i^{\rho} + \boldsymbol{h}_i^{\rho'}}{2}, \quad (7)$$

where $\boldsymbol{h}_i$ can be the node embedding of any type of item, user and category. Meanwhile, ρ and ρ' denote two different meta-paths of the corresponding type.

Multi-head attention blocks. In order to capture the dynamic preferences of users and the sequential dependencies among items, we employ self-attention blocks as the encoder of item sequences since it has been shown as an effective and efficient model in sequential recommendation [14]. For simplicity, we take a sequence from domain A as an illustration.

To take into account the positions of the previous items, a learnable position embedding $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_L\} \in \mathbb{R}^{L\times d}$ is added to the embedding of the A-domain sequence $\boldsymbol{V}^A = \{\boldsymbol{h}_{v_1^A}, \boldsymbol{h}_{v_2^A}, \ldots, \boldsymbol{h}_{v_L^A}\} \in \mathbb{R}^{L\times d}$. Then, we obtain the position-aware input embedding $\boldsymbol{X}^A = \{\boldsymbol{x}_{v_1^A}, \boldsymbol{x}_{v_2^A}, \ldots, \boldsymbol{x}_{v_L^A}\} \in \mathbb{R}^{L\times d}$:

$$\boldsymbol{x}_i^A = \boldsymbol{h}_i^A + \boldsymbol{p}_i, i \in \{1, 2, \ldots, L\}. \quad (8)$$

Next, as shown in the top half of Fig. 2(c), we feed the A-domain input sequence into some stacked self-attention blocks (SABs). Omitting the residual connection layers and the normalization layers, each *SAB* is regarded as a self-attention layer *SAL* followed by a feed-forward network *FFN*:

$$SAB(\boldsymbol{X}) = FFL(SAL(\boldsymbol{X})), \quad (9)$$

$$\boldsymbol{X}' = SAL(\boldsymbol{X}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}})\boldsymbol{\Delta} \cdot \boldsymbol{V}, \quad (10)$$

$$FFL(\boldsymbol{X}') = ReLU(\boldsymbol{X}'\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \quad (11)$$

where $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q$, $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K$ and $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$ with $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d\times d}$ are the projected query, key and value matrices, respectively. $\boldsymbol{\Delta}$ is the causality mask used to ensure that only the previous t items are taken into account when predicting the $(t+1)$th item. $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^{1\times d}$ are learnable weights and biases for the two layers of the feed-forward network, respectively.

Stacking *SAB*s is usually helpful for the model to extract the more complex sequential patterns. Finally, we take the final output vector $\boldsymbol{f}_t^{V,A}$ from the top *SAB* as the item-based sequential preferences of the user at time step t in domain A.

3.4.2 Category-based sequential preference learning

Item-category attention layer. A user's dynamic preferences can be reflected not only from a sequence of items he or she interacts with but also from the transition patterns between categories within the sequence. Considering that an item often belongs to multiple categories, most traditional approaches cannot be directly applied to model such sequences. To address this, we design an item-category attention layer to measure the association between an item and its categories, and aggregate multiple category features into a comprehensive category representation.

Specifically, for a user's interaction sequence $S^A = \{A_1, A_2, \ldots, A_L\}$ in domain A, where for the ith interaction we have $A_i = \{v_i^A, c_i^{(1)}, \ldots, c_i^{(l)}\}$. We consider the item embedding $\boldsymbol{h}_{v_i}$ of the interacted item v_i^A as the query in the attention network, and all the category embeddings $\{\boldsymbol{h}_{c_i^{(1)}}, \boldsymbol{h}_{c_i^{(2)}}, \ldots, \boldsymbol{h}_{c_i^{(l)}}\}$ of the item v_i^A as the keys and values:

$$\boldsymbol{c}_{v_i^A} = \sum_{j=1}^{l} a_j \boldsymbol{h}_{c_i^{(j)}}, \quad (12)$$

$$a_j = \frac{\exp\left(\sigma\left(\boldsymbol{h}_{\boldsymbol{v_i}}{}^T \boldsymbol{h}_{c_i^{(j)}}\right)\right)}{\sum_{j'} \exp\left(\sigma\left(\boldsymbol{h}_i^T \boldsymbol{h}_{c_i^{(j')}}\right)\right)}, \quad (13)$$

where a_j represents the weight assigned to $\boldsymbol{h}_{c_i^{(j)}}$, indicating the importance of category j in characterizing the attributes of item v_i^A, and σ is the sigmoid function. Finally, we use $\boldsymbol{c}_{v_i^A}$ as the category representation of item v_{iA}.

Category-based sequential preferences. After obtaining the user's category representation sequence $\boldsymbol{X}_c^A = \{\boldsymbol{c}_{v_1^A}, \boldsymbol{c}_{v_2^A}, \ldots, \boldsymbol{c}_{v_L^A}\}$, we feed it into the attention module $SAB(\boldsymbol{X}_c^A)$ described in 3.4.1, to explore the patterns of category transitions in the user's behavior sequence, which reflects the coarse-grained preferences of the user. Finally, we take the output $\boldsymbol{f}_t^{C,A}$ from the top *SAB* as the user's category-based sequential preferences at time step t in domain A.

3.4.3 Cross-domain sequential preference learning

Considering that a user's interaction in one domain may affect his or her next interaction in other domains, which implies that there are also some transition patterns across different domains. Therefore, we focus on cross-domain knowledge transfer by considering user preferences in both the target and source domains.

Specifically, as shown in the bottom half of Fig. 2(c), we feed a user's item sequence $\boldsymbol{X}^B = \{\boldsymbol{x}_{v_1}^B, \boldsymbol{x}_{v_2}^B, \ldots, \boldsymbol{x}_{v_L}^B\}$ into the *SAB* module described in Section 3.4.1 and obtain the item-based sequential preferences $\boldsymbol{f}_{t'}^{V,B}$ in domain B. Similar to domain A, we also feed the user's interaction sequence $S^B = \{B_1, B_2, \ldots, B_L\}$ into the item-category attention layer to obtain the category representation

sequence $\boldsymbol{X}_c^B = \{\boldsymbol{c}_{v_1^B}, \boldsymbol{c}_{v_2^B}, \ldots, \boldsymbol{c}_{v_L^B}\}$, and finally capture B-domain category-based sequential preferences $\boldsymbol{f}_{t'}^{C,B}$. Notice that t' is the most recent time step at which the same user interacted with an item in domain B before the real moment corresponding to the time step t in domain A. This is to ensure the causality of the user behaviors from the source domain to the target domain.

It is clear that our model can easily be extended to a multi-domain version when performing cross-domain heterogeneous graph construction and category-aware sequence preference learning. This differs from most previous methods that only utilize a single source domain to assist the target domain [4, 26]. Our HGTL enables knowledge transfer across multiple domains, as well as improves the performance of multiple domains simultaneously, which is of significant importance in real-world applications.

3.5 Prediction Layer

In the prediction layer, we aim to aggregate information from different domains. To alleviate the problem of inconsistencies in distribution between two domains, we draw inspiration from the idea of domain adaptation [43] and employ a cross-domain transfer unit to map users' preferences from the source domain to the feature space of the target domain.

Specifically, taking domain A as the target domain, we concatenate item-based sequential preferences $\boldsymbol{f}_{t'}^{V,B}$ and category-based sequential preferences $\boldsymbol{f}_{t'}^{C,B}$ from domain B. Then, the concatenation is fed into MLP to acquire the final representation of the user's sequential preferences in domain B:

$$\boldsymbol{f}_t^B = \boldsymbol{W}^A \left[\boldsymbol{f}_{t'}^{V,B}, \boldsymbol{f}_{t'}^{C,B}\right] + \boldsymbol{b}^A, \tag{14}$$

where $\boldsymbol{W}^A \in \mathbb{R}^{d\times 2d}$ and $\boldsymbol{b}^A \in \mathbb{R}^d$ are learnable parameters of cross-domain transfer unit from domain B to domain A.

Next, we aggregate the item-based sequential preferences $\boldsymbol{f}_t^{V,A}$ and category-based sequential preferences $\boldsymbol{f}_t^{C,A}$ from domain A, the sequential preferences $\boldsymbol{f}_t^B$ from domain B, and the user node representation $\boldsymbol{u}$ learned in heterogeneous graph to obtain the final representation of the user's preferences:

$$\boldsymbol{f}_t^A = \boldsymbol{W}'^A \left[\boldsymbol{f}_t^{V,A}, \boldsymbol{f}_t^{C,A}, \boldsymbol{f}_t^B, \boldsymbol{u}\right] + \boldsymbol{b}'^A, \tag{15}$$

where $\boldsymbol{W}'^A \in \mathbb{R}^{d\times 2d}$ and $\boldsymbol{b}'^A \in \mathbb{R}^d$ are learnable parameters. Finally, the prediction score of the next recommended item i in domain A can be formalized as follows:

$$r_{t,i}^A = \boldsymbol{f}_t^A (\boldsymbol{v}_i)^T. \tag{16}$$

We adopt the binary cross-entropy loss function as the A-domain recommendation loss in our model:

$$\mathcal{L}_r^A = -\sum_{u\in\mathcal{U}}\sum_{t=1}^{L-1} \delta(v_{t+1})[\log(\sigma(r_{t,v_{t+1}}^A)) + \log(1-\sigma(r_{t,j}^A))], \tag{17}$$

where $j \in \mathcal{V}^A \backslash \mathcal{S}^u$ is a sampled negative item and $\sigma(\cdot)$ is the sigmoid function. The indicator function $\delta(v_{t+1}) = 1$ only if v_{t+1} is not a padding item, and 0 otherwise.

Similarly, we can transfer the user's sequential features from domain A to domain B, and calculate the loss in domain B:

$$\mathcal{L}_r^B = -\sum_{u\in\mathcal{U}}\sum_{t=1}^{L-1} \delta(v_{t+1})[\log(\sigma(r_{t,v_{t+1}}^B)) + \log(1-\sigma(r_{t,j}^B))]. \tag{18}$$

Finally, we combine the two recommendation losses and the contrastive loss as the final training loss of our model:

$$\mathcal{L} = \mathcal{L}_r^A + \mathcal{L}_r^B + \lambda \mathcal{L}_{CL}, \tag{19}$$

where λ is the hyper-parameter to control the intensity of the contrastive task.

We have analyzed the model complexity in terms of heterogeneous graph construction and attention networks in Appendix A.1.

4 Experiments

In this section, we give a brief introduction to the experimental settings, and conduct extensive empirical studies to answer the following research questions: **RQ1:** How does our HGTL perform compared with the state-of-the-art methods? **RQ2:** What's the influence of various components in our HGTL? **RQ3:** Can the proposed HGTL alleviate the data sparsity issue? **RQ4:** How does our model perform with different proportions of overlapping users?

4.1 Experimental Settings

4.1.1 Datasets We conduct empirical studies on the Amazon[1] data [27], which contains overlapping users in multiple domains and records categories of items. According to the setting in [14, 38], we choose three datasets with different types, i.e., "Movie", "CD" and "Book" from the Amazon data, and preprocess the datasets. More details of the datasets processing are introduced in Appendix A.2.

4.1.2 Baselines To study the effectiveness of our HGTL, we compare it with fifteen competitive baselines, including five sequential recommendation methods (i.e., GRU4Rec [11], Caser [30], GCSAN [35], SASRec [14] and CL4SRec [33]), seven cross-domain sequential recommendation methods (i.e., DA-GCN [9], CD-SASRec [1], Tri-CDR [25], RecGURU [15], C^{2}DSR [4], MGCL [38] and TJAPL [39]) and three attribute-aware sequential recommendation methods (i.e., CaFe [17], NOVA [21] and DIF-SR [34]). Notice that some of our experimental results are copied from [38] for fair comparison. We employ two widely used ranking-oriented metrics, i.e., HR@10 (hit ratio) and NDCG@10 (normalized discounted cumulative gain) for evaluation. More details of the evaluation metrics and baseline methods are introduced in Appendix A.3 and Appendix A.4.

4.1.3 Implementation details For the general setting, the latent dimensionality d is configured as $d = 50$. The mini-batch size is set to 128, the dropout rate is set to 0.5 and the maximum length of a sequence L is set to 100. For our HGTL, we adopt the Adam optimizer with a learning rate of 0.001, and set the maximum number of node neighbors K to 10. For other baselines, the key parameters are configured following the suggestions of the corresponding papers or are tuned on the validation data.

For CDR methods, we only report the performance of the best-performing model with the corresponding source domain (i.e., when the target domain is "Movie", we use "CD" or "Book" as a source domain to assist in training, and only show the best results). For our proposed HGTL, we further report the results of **the multi-domain version (i.e., HGTL-Multi)** which can simultaneously utilize two other datasets as source domains. More implementation details and hyper-parameter settings of our method and the baselines are introduced in Appendix A.5 and Appendix A.6.

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

4.2 Overall Performance Comparison (RQ1)

Table 1 illustrates the results of our HGTL and the baselines. The best results among all methods are marked in bold, and the results of the strongest baselines are underlined.

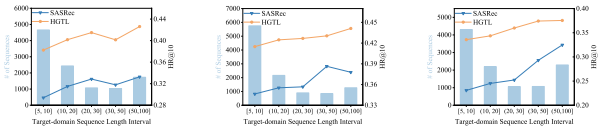
From the comparison results, we can observe that our HGTL significantly outperforms all the baselines on all three datasets, and gains 14.58% NDCG@10 and 12.52% HR@10 improvements on average against the strongest baseline, which demonstrates that our HGTL can effectively utilize data as well as category information from multiple domains by constructing a cross-domain heterogeneous graph, and explore the transition patterns across domain through knowledge transfer and attention modules. Moreover, it can be observed that among the methods within the same group, the attention-based methods tend to achieve the best results, which demonstrates the effectiveness of Transformer in capturing sequential patterns. Additionally, cross-domain sequential methods and attribute-aware sequential methods generally achieve better performance than traditional sequential methods. This verifies the significance of introducing the cross-domain data and attribute information, while also demonstrating the potential of their combination. Furthermore, **the multi-domain version** of our HGTL (i.e., HGTL-Multi) achieves the best results on the "Movie" dataset and the "Book" dataset, while on the "CD" dataset our HGTL that leverages only two domains to assist each other performs better. The reason is that although knowledge transfer across multiple domains simultaneously can utilize more information to alleviate the sparsity problem, there may be a negative transfer problem when the source and target domains are not strongly related.

4.3 Ablation Study (RQ2)

In this subsection, we conduct an ablation study to evaluate the contribution of different components of our HGTL.

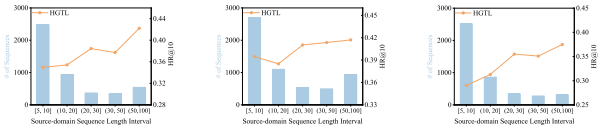
Specifically, we separate out the important components of our HGTL: (1) "w/o node representation learning" represents the version of our HGTL without meta-path guided node representation learning, (2) "w/o edge expansion" represents the version without edge expansion in heterogeneous graph construction, (3) "w/o edge contrastive learning" represents the version without meta-path based contrastive learning, (4) "w/o category-based preferences" represents the version without category-based sequential preference learning, and (5) "w/o cross-domain preferences" represents the version only utilizing data from a single target domain.

From Table 2, we observe that the largest decrease in performance occurs when the meta-path guided node representation learning is removed, which indicates that constructing heterogeneous graphs on such sparse datasets and utilizing meta-paths for node aggregation plays an important role in improving the performance. Moreover, there is also a significant decrease in the performance of the version without cross-domain preferences, which demonstrates the significance of performing knowledge transfer across domains to alleviate the sparsity problem. Furthermore, it is evident that if any individual component is removed, the performance will decrease. We can thus see that the superiority of our proposed HGTL comes from the contribution of these novel components.



(a) Movie (b) CD (c) Book

Figure 3: Performance of SASRec and our HGTL w.r.t. different sequence lengths in the target domain.



(a) Movie (b) CD (c) Book

Figure 4: Performance of our HGTL w.r.t. different sequence lengths in the source domain.

4.4 Performance Analysis w.r.t. Sparsity (RQ3)

In this subsection, we conduct experiments to explore the effect of target-domain sequence length and source-domain sequence length on the performance, in order to verify the effectiveness of introducing cross-domain data to alleviate the data sparsity problem. Notice that we report the results on "Movie" and "CD" by leveraging the "Book" domain for knowledge transfer, and the results on "Book" by leveraging the "Movie" domain, since they achieve the best performance in a single source-domain scenario.

4.4.1 Effect of target-domain sequence length We divide users into five groups according to their behavior sequence lengths in the target domain, and compare the performance of SASRec and HGTL in different user groups.

From Fig. 3, we can see that the short sequences dominate the majority in the target domain, and the number of sequences decreases as the sequences become longer, which demonstrates the overall sparsity of user behaviors is serious. Moreover, our HGTL shows its most significant improvement over SASRec in the short-sequence interval, since the single-domain method (i.e., SASRec) struggles to capture transition patterns on such a sparse data. In contrast, our HGTL can leverage abundant source domain data to enhance the learning of user preferences via knowledge transfer, effectively alleviating the issue of data sparsity. Furthermore, it can be seen that our HGTL outperforms SASRec in all cases, which demonstrates the superiority of our HGTL in sequential recommendation.

4.4.2 Effect of source-domain sequence length To examine the influence of source-domain sequence length, we fix the target-domain sequence in the shortest interval, and divide users into groups based on their sequence lengths in the source domain. The results are presented in Fig. 4.

It can be observed that similar to the target domain, the shortest source-domain sequence interval contains the largest number of sequences. Moreover, we find that the performance in the target domain generally benefits from longer source-domain sequences. This is because the richer source-domain data enables the model to capture more comprehensive preferences, which can then be transferred to and exploited by the target domain more effectively.

Table 1: Recommendation performance of our HGTL and the baselines on three datasets.

	Method	Movie		CD		Book	
		NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
SR	GRU4Rec	0.1017	0.1984	0.1210	0.2247	0.1066	0.2162
	Caser	0.1231	0.2243	0.1267	0.2473	0.1163	0.2274
	GCSAN	0.1576	0.2889	0.1783	0.3206	0.1291	0.2409
	SASRec	0.1740	0.3126	0.1965	0.3539	0.1402	0.2597
	CL4SRec	0.1821	0.3179	0.1936	0.3350	0.1409	0.2556
CDSR	DA-GCN	0.1736	0.3124	0.1897	0.3458	0.1283	0.2375
	CD-SASRec	0.1787	0.3159	0.1995	0.3610	0.1438	0.2677
	Tri-CDR	0.1735	0.3190	0.1873	0.3331	0.1480	0.2716
	RecGURU	0.1884	0.3433	0.2044	0.3649	0.1373	0.2556
	C^2DSR	0.1922	0.3423	0.1978	0.3435	0.1486	0.2752
	MGCL	0.2092	0.3693	0.2156	0.3797	0.1542	0.2842
	TJAPL	0.2133	0.3769	0.2199	0.3907	0.1632	0.2984
A-SR	CaFe	0.1874	0.3332	0.1925	0.3647	0.1445	0.2794
	NOVA	0.2040	0.3602	0.2128	0.3694	0.1526	0.2752
	DIF-SR	0.2124	0.3712	0.2195	0.3741	0.1634	0.2826
	HGTL	0.2335	0.4052	**0.2437**	**0.4262**	0.1942	0.3475
	HGTL-Multi	**0.2386**	**0.4129**	0.2344	0.4131	**0.1978**	**0.3549**
	Improv.	11.86%	9.55%	10.82%	9.09%	21.05%	18.93%

Table 2: Recommendation performance in ablation studies of our HGTL with different architectures.

Architecture	Movie		CD		Book	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
HGTL	**0.2355**	**0.4052**	**0.2437**	**0.4262**	**0.1942**	**0.3475**
w/o node representation learning	0.2009	0.3611	0.2123	0.3726	0.1727	0.3091
w/o edge expansion	0.2298	0.3960	0.2376	0.4159	0.1914	0.3396
w/o contrastive learning	0.2279	0.3918	0.2340	0.4059	0.1742	0.3228
w/o category-based preferences	0.2237	0.3896	0.2323	0.4084	0.1721	0.3247
w/o cross-domain preferences	0.2209	0.3850	0.2304	0.4027	0.1776	0.3192

Table 3: Recommendation performance with different portions of overlapping users ranging from 0% to 100%.

Dataset	OverlapRadio	Movie		CD		Book	
Meritic		NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
SASRec	–	0.1740	0.3126	0.1965	0.3539	0.1402	0.2597
DIF-SR	–	0.2124	0.3712	0.2985	0.3741	0.1634	0.2826
CD-SASRec	0%	0.1669	0.2985	0.1957	0.3518	0.1387	0.2561
	20%	0.1697	0.3022	0.2026	0.3561	0.1404	0.2613
	40%	0.1759	0.3078	0.2053	0.3598	0.1475	0.2644
	60%	0.1771	0.3129	0.2042	0.3597	0.1472	0.2673
	100%	0.1787	0.3159	0.1995	0.3610	0.1438	0.2677
MGCL	0%	0.1752	0.3154	0.1964	0.3521	0.1425	0.2654
	20%	0.1779	0.3216	0.1995	0.3554	0.1484	0.2725
	40%	0.1881	0.3405	0.2069	0.3651	0.1516	0.2774
	60%	0.1957	0.3479	0.2114	0.3723	0.1527	0.2825
	100%	0.2092	0.3693	0.2156	0.3797	0.1542	0.2842
HGTL	0%	0.2212	0.3793	0.2343	0.4084	0.1833	0.3274
	20%	0.2243	0.3827	0.2364	0.4102	0.1894	0.3318
	40%	0.2276	0.3861	0.2408	0.4138	0.1912	0.3391
	60%	0.2314	0.3925	0.2418	0.4186	0.1936	0.3469
	100%	0.2335	0.4052	0.2437	0.4262	0.1942	0.3475

4.5 Study of Overlapping Users (RQ4)

In order to verify the performance of our model in scenarios with different user overlap proportions, we vary the overlapping user proportions to 0%, 20%, 40%, 60% and 100%. Specifically, we retain the entire training data of all users in the target domain, and randomly select a corresponding number of users according to the overlap proportion, retaining only the connections of these users in the source domain and the target domain. For SR methods and A-SR methods, varying the proportion of overlapping users does not affect the performance since these models only utilize data from a single domain. However, for most previous cross-domain recommendation methods, reducing the proportion of overlapping users results in fewer bridges for knowledge transfer, which may lead to performance degradation.

We can observe from Table 3 that the traditional CDSR methods do not perform well when the percentage of overlapping users is low, and in some cases are even worse than the single-domain recommendation methods (especially when the users are entirely non-overlapping) since they rely on overlapping users as a bridge to transfer knowledge. In contrast, our model achieves the best results in all cases due to its ability to utilize the information of non-overlapping users through categories, which demonstrates the robustness of our model in terms of the overlapping user proportion.

5 Conclusions and Future Work

In this paper, we study a new and emerging problem, i.e., category-aware cross-domain sequential recommendation, and propose our HGTL to deal with it. Specifically, we associate nodes from different domains by constructing a cross-domain heterogeneous graph, and design a meta-path guided node representation learning module for knowledge transfer across domains. Moreover, we enhance the correlation between different domains via an edge expansion module. After that, we employ a category-aware sequential preference learning module to capture the item-based and category-based preferences in each domain, which reflect the coarse-grained and fine-grained interests of users, respectively. Notice that our HGTL can be applied to a multi-domain scenario, which is more adaptable and flexible in real-world applications. Extensive empirical studies on three real-world datasets indicate that our HGTL significantly outperforms various competitive baselines in all cases. For future works, we intend to employ our HGTL for cross-domain or cross-organization privacy-aware federated recommendation [20].

References

[1] Nawaf Alharbi and Doina Caragea. 2022. Cross-Domain Self-Attentive Sequential Recommendations. In *Proceedings of International Conference on Data Science and Applications (ICONDATA '22)*. 601–614.

[2] Ye Bi, Liqiang Song, Mengqiu Yao, Zhenyu Wu, Jianming Wang, and Jing Xiao. 2020. A Heterogeneous Information Network based Cross Domain Insurance Recommendation System for Cold Start Users. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR '20)*. 2211–2220.

[3] Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. 2021. Category-aware Collaborative Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval1 (SIGIR '21)*. 388–397.

[4] Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. 2022. Contrastive Cross-Domain Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*. 138–147.

[5] Jiangxia Cao, Xixun Lin, Xin Cong, Jing Ya, Tingwen Liu, and Bin Wang. 2022. DisenCDR: Learning Disentangled Representations for Cross-Domain Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval2 (SIGIR '22)*. 267–277.

[6] Shu Chen, Zitao Xu, Weike Pan, Qiang Yang, and Zhong Ming. 2024. A Survey on Cross-Domain Sequential Recommendation. In *Proceedings of the 33rd Joint Conference on Artificial Intelligence (IJCAI '24)*. 7989–7998.

[7] Chenguang Du, Kaichun Yao, Hengshu Zhu, Deqing Wang, Fuzhen Zhuang, and Hui Xiong. 2023. Seq-HGNN: Learning Sequential Node Representation on Heterogeneous Graph. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. 1721–1730.

[8] Jingtong Gao, Xiangyu Zhao, Bo Chen, Fan Yan, Huifeng Guo, and Ruiming Tang. 2023. AutoTransfer: Instance Transfer for Cross-Domain Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. 1478–1487.

[9] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: A Domain-aware Attentive Graph Convolution Network for Shared-account Cross-domain Sequential Recommendation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI '21)*. 2483–2489.

[10] Yongqiang Han, Hao Wang, Kefan Wang, Likang Wu, Zhi Li, Wei Guo, Yong Liu, Defu Lian, and Enhong Chen. 2024. Efficient Noise-Decoupling for Multi-Behavior Sequential Recommendation. In *Proceedings of the ACM on Web Conference 2024 (WWW '24)*. 3297–3306.

[11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR '16)*.

[12] Jun Hu, Wenwen Xia, Xiaolu Zhang, Chilin Fu, Weichang Wu, Zhaoxin Huan, Ang Li, Zuoli Tang, and Jun Zhou. 2024. Enhancing Sequential Recommendation via LLM-based Semantic Embedding Learning. In *Companion Proceedings of the ACM on Web Conference 2024 (WWW '24)*. 103–111.

[13] Zepeng Huai, Yuji Yang, Mengdi Zhang, Zhongyi Zhang, Yichun Li, and Wei Wu. 2023. M2GNN: Metapath and Multi-interest Aggregated Graph Neural Network for Tag-based Cross-domain Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. 1468–1477.

[14] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *Proceedings of the 18th IEEE International Conference on Data Mining (ICDM '18)*. 197–206.

[15] Chenglin Li, Mingjun Zhao, Huanming Zhang, Chenyun Yu, Lei Cheng, Guoqiang Shu, Beibei Kong, and Di Niu. 2022. RecGURU: Adversarial Learning of Generalized User Representations for Cross-Domain Recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM '22)*. 571–581.

[16] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian J. McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. 1258–1267.

[17] Jiacheng Li, Tong Zhao, Jin Li, Jim Chan, Christos Faloutsos, George Karypis, Soo-Min Pantel, and Julian J. McAuley. 2022. Coarse-to-Fine Sparse Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 2082–2086.

[18] Guanyu Lin, Chen Gao, Yu Zheng, Jianxin Chang, Yanan Niu, Yang Song, Kun Gai, Zhiheng Li, Depeng Jin, Yong Li, and Meng Wang. 2024. Mixed Attention Network for Cross-domain Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24)*. 405–413.

[19] Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys '20)*. 130–139.

[20] Zhaohao Lin, Weike Pan, Qiang Yang, and Zhong Ming. 2023. Recommendation Framework via Fake Marks and Secret Sharing. *ACM Transactions on Information Systems* 41(2) (2023), 40:1–40:37.

[21] Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. 2021. Noninvasive Self-attention for Side Information Fusion in Sequential Recommendation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI '21)*. 4249–4256.

[22] Lin Liu, Li Wang, and Tao Lian. 2021. CaSe4SR: Using Category Sequence Graph to Augment Session-based Recommendation. *Knowledge-Based Systems* 212 (2021), 106558.

[23] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Jiajie Su, Yanchao Tan, and Fan Wang. 2024. User Distribution Mapping Modelling with Collaborative Filtering for Cross Domain Recommendation. In *Proceedings of the ACM on Web Conference 2024 (WWW '24)*. 334–343.

[24] Weiming Liu, Xiaolin Zheng, Jiajie Su, Mengling Hu, Yanchao Tan, and Chaochao Chen. 2022. Exploiting Variational Domain-Invariant User Embedding for Partially Overlapped Cross Domain Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 312–321.

[25] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. 2024. Triple Sequence Learning for Cross-domain Recommendation. *ACM Transactions on Information Systems* 42, 4 (2024), 1–29.

[26] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π-net: A Parallel Information-Sharing Network for Shared-Account Cross-Domain Sequential Recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. 685–694.

[27] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. 43–52.

[28] Chung Park, Taesan Kim, Hyungjun Yoon, Junui Hong, Yelim Yu, Mincheol Cho, Minsung Choi, and Jaegul Choo. 2024. Pacer and Runner: Cooperative Learning Framework between Single- and Cross-Domain Sequential Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. 2071–2080.

[29] Wenchao Sun, Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2023. Parallel Split-Join Networks for Shared Account Cross-domain Sequential Recommendations. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2023), 4106–4123.

[30] Jiaxi Tang and Ke Wang. 2018. Personalized top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM '18)*. 565–573.

[31] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Xiang Ao, Xin Chen, Xu Zhang, Fuzhen Zhuang, Leyu Lin, and Qing He. 2022. Multi-view Multi-behavior Contrastive Learning in Recommendation. In *Proceedings of the 27th International Conference on Database Systems for Advanced Applications (DASFAA '22)*. 166–182.

[32] 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 35th AAAI Conference on Artificial Intelligence (AAAI '21)*. 4503–4511.

[33] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *Proceedings of the 38th IEEE International Conference on Data Engineering (ICDE '22)*. 1259–1273.

[34] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled Side Information Fusion for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 1611–1621.

[35] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-attention Network for Session-based Recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19)*. 3940–3946.

[36] Hao Xu, Bo Yang, Xiangkun Liu, Wenqi Fan, and Qing Li. 2022. Category-aware Multi-relation Heterogeneous Graph Neural Networks for session-based recommendation. *Knowledge-Based Systems* 251 (2022), 109246.

[37] Wujiang Xu, Qitian Wu, Runzhong Wang, Mingming Ha, Qiongxu Ma, Linxun Chen, Bing Han, and Junchi Yan. 2024. Rethinking Cross-Domain Sequential Recommendation under Open-World Assumptions. In *Proceedings of the ACM on Web Conference 2024 (WWW '24)*. 3173–3184.

[38] Zitao Xu, Weike Pan, and Zhong Ming. 2023. A Multi-view Graph Contrastive Learning Framework for Cross-Domain Sequential Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. 491–501.

[39] Zitao Xu, Weike Pan, and Zhong Ming. 2024. Transfer Learning in Cross-domain Sequential Recommendation. *Informatuon Sciences* 669 (2024), 120550.

[40] Wenhao Yang, Yingchun Jian, Yibo Wang, Shiyin Lu, Lei Shen, Bing Wang, Haihong Tang, and Lijun Zhang. 2024. Not All Embeddings are Created Equal: Towards Robust Cross-domain Recommendation via Contrastive Learning. In *Proceedings of the ACM on Web Conference 2024 (WWW '24)*. 3195–3206.
[41] Xiaoxin Ye, Yun Li, and Lina Yao. 2023. DREAM: Decoupled Representation via Extraction Attention Module and Supervised Contrastive Learning for Cross-Domain Sequential Recommender. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. 479–490.
[42] Jiangyi Yin, Yuchun Guo, and Yishuai Chen. 2019. Heterogenous Information Network Embedding Based Cross-Domain Recommendation System. In *Proceedings of the 2019 International Conference on Data Mining Workshops (ICDM '19)*. 362–369.
[43] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. 2019. Bridging Theory and Algorithm for Domain Adaptation. In *Proceedings of the 36th International Conference on Machine Learning (ICML '19)*. 7404–7413.
[44] Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, and Jibing Gong. 2023. Beyond the Overlapping Users: Cross-Domain Recommendation via Adaptive Anchor Link Learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. 1488–1497.
[45] Peilin Zhou, You-Liang Huang, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, and Sunghun Kim. 2024. Is Contrastive Learning Necessary? A Study of Data Augmentation vs Contrastive Learning in Sequential Recommendation. In *Proceedings of the ACM on Web Conference 2024 (WWW '24)*. 3854–3863.

A APPENDIX

A.1 Complexity Analysis

We can analyze model complexity in terms of two main modules: heterogeneous graph construction and attention networks:

The complexity of a graph neural network mainly depends on the way the graph is constructed and the way message is propagated between nodes. We employ a meta-path approach to constructing the graph, which in a practical implementation would only connect the head and tail nodes on the eligible paths, and these two head and tail nodes can be considered as neighboring nodes to each other.

Assuming there are $|\mathcal{V}|$ items, $|\mathcal{U}|$ users, and $|C|$ categories, the time complexity for constructing the meta-paths with two edges (i.e., I-C-I, U-I-U, C-I-C, I-U-I) is $O(|\mathcal{V}||N_c|)$, $O(|\mathcal{U}||N_{u_i}|)$, $O(|C||N_{c_i}|)$ and $O(|\mathcal{V}||N_u|)$, respectively. Here $|N_c|$ represents the number of categories contained under each item, $|N_{u_i}|$ represents the number of items each user has interacted with, $|N_{c_i}|$ represents the number of items each category contains, and $|N_u|$ represents the number of users each item being interacted with. Since the values of $|N|$ are generally not too large, the time complexity remains relatively low. For meta-paths with four edges (i.e., U-I-C-I-U, C-I-U-I-C), the time complexity is $O(|\mathcal{U}||N_{u_i}||N_c||N_{c_i}|)$ and $O(|C||N_{c_i}||N_u||N_{u_i}|)$, respectively, which is still within an acceptable range.

For the self-attention block, it generally consists of the following steps: mapping the input features to query, key, and value, calculating similarity, normalizing the similarity using the softmax function, calculating the weighted sum, and finally passing through the non-linear layer FFN. The time complexity of the above steps are $O(3Ld^2)$, $O(L^2d)$, $O(L^2)$, $O(L^2d)$, and $O(2Ld^2)$, where L is the length of the sequence, and d is the dimensionality of embedding. If we ignore the constant term, then we can get the time complexity of SASRec as $O(2L^2d + 5Ld^2)$.

For our model, the time complexity of the item-based attention block and category-based attention block is the same as the self-attention block, and the time complexity of the item-category attention layer is $O(L(2|N_{c_i}|d + |N_{c_i}|^2))$, which does not significantly increase the complexity due to the small value of $|N_{c_i}|$. Finally, the complexity of all attention networks in the cross-domain scenario can be summarized as $O(|M|[4L^2d + 10Ld^2 + L(2|N_{c_i}|d + |N_{c_i}|^2)])$, where $|M|$ denotes the number of domains. Since we only need to train one model to accomplish the goal of making recommendations on multiple domains at the same time, and the self-attention layer is amenable to GPU acceleration, there is no significant increase in time complexity compared with the methods that train in each domain separately.

Table 4: Statistical details of the datasets.

Dataset	# Users	# Items	# Categories	# Interactions	Density
Movie	10929	59513	733	460226	0.07%
CD	10929	91169	670	344221	0.03%
Book	10929	236049	581	607657	0.02%

A.2 Details of Experimental Datasets

We conduct empirical studies on the Amazon[2] data [27], which contains overlapping users in multiple domains and records categories of items. According to the setting in [14, 38], we choose three datasets with different types, i.e., "Movie", "CD" and "Book" from the Amazon data, and preprocess the datasets as follows:

(1) We take the occurrence of interaction behaviors as positive feedback and use the timestamps to determine the order of the interactions.
(2) We retain only users and items with at least five interactions, and discard later duplicate (user, item) pairs.
(3) We only keep the sequence of a user who has interactions in all the three domains.
(4) We adopt the leave-one-out evaluation which divides a user's sequence into three parts, i.e., the last interaction for test, the penultimate interaction for validation and the rest for training.

Table 4 shows the statistical details of the processed datasets. Notice that the dataset is constructed with fully overlapping users for a fair comparison with traditional cross-domain recommendation methods. To verify that our model can also perform well with a small number of overlapping users, we have conducted experiments in Section 4.5.

A.3 Evaluation Metrics

We employ two widely used ranking-oriented metrics, i.e., HR@10 (hit ratio) and NDCG@10 (normalized discounted cumulative gain) for evaluation , where the former corresponds to recall because there is exactly one preferred item for each user in the test data in our case. In particular, HR@10 denotes to the proportion of ground-truth items presenting in the top-10 recommended lists, while NDCG@10 is sensitive to the exact ranking positions of the items in the lists. To avoid heavy computation on all the (user, item) pairs, we follow the common strategy in [14, 38] which samples some negative items as candidates. And these sampled items have not been interacted with by the corresponding users and based on the popularity to ensure that they are informative and representative [19]. Notice that we do not use the full item set as candidates since the evaluation results for all baselines are extremely poor on such a sparse data, and the parameter settings as well as the

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

randomness of the neural network have a significant impact on the results in this scenario.

A.4 Details of Baseline Methods

To study the effectiveness of our HGTL, we compare it with fifteen competitive baselines, including five sequential recommendation methods (i.e., GRU4Rec [11], Caser [30], GCSAN [35], SASRec [14] and CL4SRec [33]), seven cross-domain sequential recommendation methods (i.e., DA-GCN [9], CD-SASRec [1], Tri-CDR [25], RecGURU [15], C^{2}DSR [4], MGCL [38] and TJAPL [39]) and three attribute-aware sequential recommendation methods (i.e., CaFe [17], NOVA [21] and DIF-SR [34]). Notice that some of our experimental results are copied from [38] for fair comparison.

- GRU4Rec [11]. An RNN-based method for sequential recommendation which adopts GRU to model users' behavior sequences.
- Caser [30]. A CNN-based model which adopts horizontal and vertical convolutional filters to capture the item dependencies over the sequences.
- GCSAN [35]. A GNN-based model which constructs a directed graph for each session and employs gated GNNs to learn the node representations in the graph.
- SASRec [14]. An attention-based model that captures the dynamic preferences of users by applying the attention mechanism.
- CL4SRec [33]. A self-supervised model that adopts different data augmentation approaches to construct contrastive learning tasks.
- DA-GCN [9]. A GNN-based model which employs graph convolution networks to learn the complicated relationships and the structural information in a cross-domain sequence graph.
- CD-SASRec [1]. An improved method based on SASRec [14] which fuses the source-domain aggregated vector into the target-domain item embedding to transfer information across domains.
- Tri-CDR [25]. A novel model which jointly models the source-domain, target-domain, and mixed behavior sequences to explore the triple correlation and learn more accurate multi-domain representations.
- RecGURU [15]. An adversarial learning model which employs a self-attentive autoencoder to derive latent user representations, and unify user embeddings generated from different domains into a single global generalized user representation, in order to captures the overall preferences of users.
- C^{2}DSR [4]. A novel model which devises two sequential objectives with a contrastive objective to jointly learn the single-domain and cross-domain user representations.
- MGCL [38]. A novel model which adopts the contrastive mechanism in an intra-domain item representation view and an inter-domain user preference view to learn the dynamic sequential information and the static collaborative information.
- TJAPL [39]. A novel model which tackles the CDSR problem from the perspective of attentive preference learning, transfering knowledge from multiple source domains to a target domain.
- CaFe [17]. An attribute-aware model which learns intents from coarse-grained sequences and fuses intent representations into the output of the item encoder to obtain item representations.
- NOVA [21]. An improved model under the BERT framework which makes use of side information to generate better attention distribution, rather than directly altering the item embeddings.
- DIF-SR [34]. An extended model based on NOVA [21] which moves the fusion process from the input layer to the attention layer and proposes an auxiliary attribute predictor to activate the interaction between the attributes and items.

A.5 Implementation Datails of the Methods

We implement our HGTL by PyTorch and follow the released codes[3-12] to implement the baselines. For the general setting, the latent dimensionality d is chosen from $\{10, 20, 30, 40, 50\}$ and finally configure it as $d = 50$ since we find that on such sparse datasets, these methods usually benefit from a larger value of d [14, 30]. The mini-batch size is set to 128, the dropout rate is set to 0.5 and the maximum length of a sequence L is set to 100. For our HGTL, we adopt the Adam optimizer with a learning rate of 0.001. We select the value of the maximum number of node neighbors K from the range of $\{5, 10, 15, 20, 25\}$, and finally obtained the best result on the validation set with $K = 10$. For the other hyper-parameters, like the weight of the contrastive learning task λ, and the value of the thresholds of edge expansion (i.e., τ_1 for the semantic similarity and τ_2 for the co-occurrence probability), we conduct experiments to study the influence on the model performance in Session A.6.

For Caser, the numbers of vertical and horizontal filters are set to 4 and 16, respectively. For the methods with Transformer architectures, we adopt single-head attention layers and two attention blocks. For all GNN-based methods, the depth of the GNN layer is set to 2. For the attribute-aware methods, we regard the categories of items as attributes. For the shared-account recommendation methods (i.e., π-Net and DA-GCN), the latent user number is set to 1. For other baselines, the key parameters are configured following the suggestions of the corresponding papers or are tuned on the validation data.

For cross-domain recommendation methods, we only report the performance of the best-performing model with the corresponding source domain (i.e., when the target domain is "Movie", we use "CD" or "Book" as a source domain to assist in training, and only show the best results). For our proposed HGTL, since our model can be applied to a multi-domain scenario, we further report the results of the multi-domain version (i.e., HGTL-Multi) which utilizes two other datasets as source domains. All the models are trained using Tesla V100 PCIe GPU with 32 GB memory. The source codes of our HGTL and datasets are available at https://anonymous.4open.science/r/HGTL-C135.

A.6 Influence of Hyper-parameter

In this subsection, we study the influence of hyper-parameters on the model performance. Specifically, we vary the value of the weight of the meta-path based contrastive learning task λ in the range of $\{0, 0.25, 0.5, 0.75, 1\}$, and the value of the thresholds of

[3]https://github.com/hidasib/GRU4Rec
[4]https://github.com/graytowne/caser_pytorch
[5]https://github.com/kang205/SASRec
[6]https://github.com/YChen1993/CoSeRec
[7]https://github.com/hulkima/Tri-CDR
[8]https://github.com/Chain123/RecGURU
[9]https://github.com/cjx96/C2DSR
[10]https://csse.szu.edu.cn/staff/panwk/publications/MGCL/
[11]https://github.com/JiachengLi1995/CAFE
[12]https://github.com/AIM-SE/DIF-SR

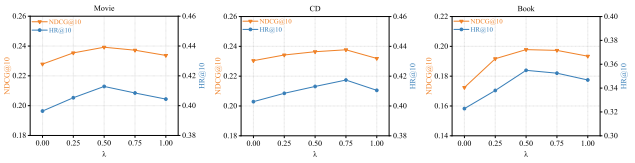
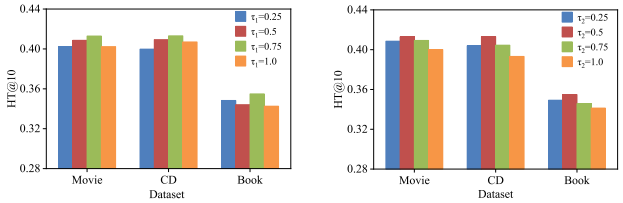


Figure 5: Performance of our HGTL with different weight on the contrastive loss.



(a) Semantic similarity-based correlation

(b) Co-occurrence patterns-based correlation

Figure 6: Performance of different edge expansion methods.

edge expansion (i.e., τ_1 for the semantic similarity and τ_2 for the co-occurrence probability) in the range of $\{0.25, 0.5, 0.75, 1\}$.

From Fig. 5, we observe that the performance improves on all datasets as the value of the parameter λ increases from 0 to some larger values, which indicates the effectiveness of the contrastive learning task. Moreover, when the parameter value is set to 0.5, our model can achieve the best results in most cases, but as the parameter value further increases to 1, the performance gradually decreases. This suggests that over-focusing on the contrastive learning task may hurt the recommendation performance, since the supervised signal should assist rather than dominate the training process.

From Fig. 6, we observe that when the values of τ_1 and τ_2 become extremely high or low, the model tends to be less effective. That's because larger parameter values indicate stricter conditions for edge expansion, whereas smaller values imply more relaxed conditions. When the value of the parameter equals to 1, it indicates that this type of edge expansion method is not employed. And a small value of the parameter expands lots of weakly correlated edges, in which case the introduction of extra information and noise would degrade the performance.