DDGHM: Dual Dynamic Graph with Hybrid Metric Training for Cross-Domain Sequential Recommendation

Xiaolin Zheng
College of Computer Science, Zhejiang University
Hangzhou, China
xlzheng@zju.edu.cn

Jiajie Su
College of Computer Science, Zhejiang University
Hangzhou, China
sujiajie@zju.edu.cn

Weiming Liu
College of Computer Science, Zhejiang University
Hangzhou, China
21831010@zju.edu.cn

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

ABSTRACT

Sequential Recommendation (SR) characterizes evolving patterns of user behaviors by modeling how users transit among items. However, the short interaction sequences limit the performance of existing SR. To solve this problem, we focus on Cross-Domain Sequential Recommendation (CDSR) in this paper, which aims to leverage information from other domains to improve the sequential recommendation performance of a single domain. Solving CDSR is challenging. On the one hand, how to retain single domain preferences as well as integrate cross-domain influence remains an essential problem. On the other hand, the data sparsity problem cannot be totally solved by simply utilizing knowledge from other domains, due to the limited length of the merged sequences. To address the challenges, we propose **DDGHM**, a novel framework for the CDSR problem, which includes two main modules, i.e., dual dynamic graph modeling and hybrid metric training. The former captures intra-domain and inter-domain sequential transitions through dynamically constructing two-level graphs, i.e., the local graphs and the global graphs, and incorporating them with a fuse attentive gating mechanism. The latter enhances user and item representations by employing hybrid metric learning, including collaborative metric for achieving alignment and contrastive metric for preserving uniformity, to further alleviate data sparsity issue and improve prediction accuracy. We conduct experiments on two benchmark datasets and the results demonstrate the effectiveness of **DDGHM**.

CCS CONCEPTS

• **Information systems → Recommender systems**.

---

*Chaochao Chen is the corresponding author.

KEYWORDS

Cross-Domain Recommendation; Sequential Recommendation; Dynamic Graph

1 INTRODUCTION

Sequential Recommendation (SR) has attracted increasing attention due to its significant practical impact. SR aims to find the potential patterns and dependencies of items in a sequence, and understand a user's time-varying interests to make next-item recommendation [7, 10, 29, 32, 34]. Though some SR models have been proposed, they face the difficulty in characterizing user preferences when behavior sequences are short, e.g., data sparsity [13]. Therefore, it is necessary to utilize more information, e.g., side information or data from other domains, to mitigate the above problem.

In this paper, we focus on the Cross-Domain Sequential Recommendation (CDSR) problem, which considers the next-item prediction task for a set of common users whose interaction histories are recorded in multiple domains during the same time period. Similar as how traditional Cross-Domain Recommendation (CDR) helps leverage information from a source domain to improve the recommendation performance of a target domain [46], CDSR shows its superiority by incorporating sequential information from different domains [18]. Intuitively, a user's preference could be reflected by his behaviors in multiple domains. We motivate this through the example in Figure 1, where a user has some alternating interactions in a movie domain and a book domain during a period of time. From it, we can easily observe that the user's choice for the next movie 'Harry Potter' depends not only on his previous interest for mystery and fantasy movies (*intra-domain*), but also on his reading experience of the original book 'Harry Potter' (*inter-domain*).

Solving the CDSR problem is challenging. On the one hand, complex sequential item transitions exist simultaneously inside domains and across domains, which makes it difficult to capture and transfer useful information. On the other hand, although transferring auxiliary information from another domain helps explore sequential patterns, data sparsity problem still exists because a large

Figure 1: A motivating example.

number of items in both domains never or rarely appear in historical sequences. Thus, how to effectively represent and aggregate both intra-domain and inter-domain sequential preferences as well as further alleviate data sparsity problem remains a crucial issue.

Existing researches on SR, CDR, and CDSR cannot overcome these challenges well. First, various methods have been proposed for SR [7, 10, 26, 29, 33, 38], but most of them only focus on the transition correlations in a single domain without considering the auxiliary information from other domains. Second, although several representative CDR approaches [2, 9, 15, 47] address the data sparsity and cold-start problem by transferring knowledge within domains, they cannot take full advantage of sequential patterns. Third, existing CDSR methods [1, 4, 14, 18, 30] reach the breakthrough of exploring sequential dependencies and modeling structure information that bridges two domains. However, these CDSR models cannot extract and incorporate both the intra-domain and the inter-domain item transitions in a dynamical and synchronous way. Besides, the long-standing data sparsity problem is still overlooked.

To overcome the above limitations, we propose **DDGHM**, a dual dynamic graphical model with hybrid metric training, to solve the CDSR problem. The purpose of **DDGHM** is twofold: (1) exploit and fuse the evolving patterns of users' historical records from two aspects, i.e., intra-domain and inter-domain. (2) Enhance representation learning to address the remaining data sparsity problem so as to further improve sequential recommendation performance. To do that, we build two modules in **DDGHM**, i.e., *dual dynamic graph modeling* and *hybrid metric training*. (i) The *dual dynamic graph modeling module* simultaneously constructs separate graphs, i.e., local dynamic graphs and global dynamic graphs, to encode the complex transitions intra-domain and inter-domain respectively. The structure of directed graph equipped with gated recurrent units makes it possible to integrate users' long-term and short-term interest discriminatively into the sequential representation. Meanwhile, a novel gating mechanism with specialized fuse attention algorithm is adopted to effectively filter and transfer cross-domain information that might be useful for single domain recommendation. Dual dynamic graph modeling generates both item representations and complete sequence representations that indicate users' general preferences, and send them into (ii) the *hybrid metric training module*. Hybrid metric training not only completes estimating recommendation scores for each item in both domains, but also alleviates the impact of data sparsity in CDSR by employing representation enhancement in two perspectives, i.e., a) *collaborative metric* for realizing alignment within similar instances and b) *contrastive metric* for preserving uniformity of feature distributions.

We summarize our main contributions as follows. (1) We propose an innovative framework **DDGHM** that effectively leverages cross-domain information to promote recommendation performance under CDSR scenario. (2) We design a dual dynamic graph modeling module which captures both intra-domain and inter-domain transition patterns and attentively integrate them in an interpretable way. (3) We develop a hybrid metric training module which weakens the data spasity impact in CDSR by enhancing representation learning of user preferences. (4) We conduct experiments on two real-word datasets and the results demonstrate the effectiveness of **DDGHM**.

2 RELATED WORK

Sequential Recommendation. Sequential Recommendation (SR) is built to characterize the evolving user patterns by modeling sequences. Early work on SR usually models the sequential patterns with the Markov Chain assumption [26]. With the advance in neural networks, Recurrent Neural Networks (RNN) based [3, 6, 7, 37], Convolutional Neural Networks (CNN) based [32], Graph Neural Networks (GNN) based [24, 38, 43, 44] methods, and Transformers [10, 29] have been adopted to model the dynamic preferences of users over behavior sequences. Recently, unsupervised learning based models [23, 39] are introduced to extract more meaningful user patterns by deriving self-supervision signals. While these methods aim to improve the overall performance via representation learning for sequences, they suffer from weak prediction power for cold-start and data sparsity issues when the sequence length is limited.

Cross-Domain Recommendation. Cross-Domain Recommendation (CDR) is proposed to handle the long-standing cold-start and data sparsity problems that commonly exist in traditional single domain recommender systems [12, 21]. The basic assumption of CDR is that different behavioral patterns from multiple domains jointly characterize the way users interact with items [45]. According to [46], existing CDR have three main types, i.e., *transfer-based*, *multitask-based*, and *clustered-based* methods. Transfer-based methods [19, 42] learn a linear or nonlinear mapping function across domains. Multitask-based methods [9, 16, 47] enable dual knowledge transfer by introducing shared connection modules in neural networks. Clustered-based methods [35] adopt co-clustering approach to learn cross-domain comprehensive embeddings by collectively leveraging single-domain and cross-domain information within a unified framework. However, conventional CDR approaches cannot perfectly solve the CDSR problem, because they fail to capture sequential dependencies that commonly exist in transaction data.

Cross-Domain Sequential Recommendation. Existing researches on CDSR can be divided into three categories, i.e., RNN-based, GNN-based, and attentive learning based methods. First, RNN-based methods, e.g., π-Net [18] and PSJNet [30], employ RNN to generate user-specific representations, which emphasize exploring sequential dependencies but fail to depict transitions among associated entities. Second, GNN-based method DA-GCN [4] devises a domain-aware convolution network with attention mechanisms to learn node representations, which bridges two domains via knowledge transfer but cannot capture inter-domain transitions on the item level. Third, attentive learning based methods [1, 14] adopt dual

attentive sequential learning to bidirectionally transfer user preferences between domain pairs, which show the effectiveness in leveraging auxiliary knowledge but cannot excavate structured patterns inside sequential transitions. To summary, the existing CDSR methods cannot extract and integrate both inter-domain and intra-domain information dynamically and expressively, and they all neglect the data sparsity issue which still remains after aggregating information from multiple domains.

3 THE PROPOSED MODEL

3.1 Problem Formulation

CDSR aims at exploring a set of common users' sequential preference with given historical behavior sequences from multiple domains during the same time period. Without loss of generality, we take two domains (A and B) as example, and formulate the CDSR problem as follows. We represent two single-domain behavior sequences of a user as $S_A = \{A_1, A_2, ..., A_i, ...\}$ and $S_B = \{B_1, B_2, ..., B_j, ...\}$, where A_i and B_j are the consumed items in domain A and B respectively. Then the cross-domain behavior sequence S is produced by merging S_A and S_B in the chronological order, for which the above example can be given as $S_M = \{B_1, A_1, B_2, ..., A_i, B_j, ...\}$. Given S_A, S_B, and S_M, CDSR tries to predict the next item that will be consumed in domain A and B.

3.2 An overview of DDGHM

The aim of **DDGHM** is providing better sequential recommendation performance for a single domain by leveraging useful cross-domain knowledge. The overall structure of our proposed **DDGHM** is illustrated in Figure 2. **DDGHM** consists of two main modules: (1) *dual dynamic graph modeling* and (2) *hybrid metric training*. In the dual dynamic graph modeling module, we explore intra-domain and inter-domain preference features in parallel, and then transfer the information recurrently at each timestamp. Thus, this module has two parts, i.e., *dual dynamic graphs* and *fuse attentive gate*. In the dual dynamic graphs part, we build two-level directed graphs : a) *local dynamic graphs* for extracting *intra-domain* transitions; b) *global dynamic graphs* for extracting *inter-domain* transitions. After it, we propose a fuse attentive gate which adopts attention mechanism to integrate item embeddings from global graphs into local graphs. To this end, cross-domain information is effectively leveraged to enrich the single domain representations. Later, sequence representations that combine intra-domain and inter-domain features are sent into the hybrid metric training module.

Though cross-domain information has been utilized, the data sparsity problem still exists. Thus, in the hybrid metric training module, we propose the representation enhancement for optimizing representation learning so as to reduce data sparsity impact. The enhancement is twofold : a) adopt collaborative metric learning to realize alignment between similar instance representations; b) employ contrastive metric learning to preserve uniformity within different representation distributions. Finally, the model outputs the probability of each item to be the next click for both domains.

3.3 Dual Dynamic Graph Modeling

The prime task for solving the CDSR problem is exploiting the expressive sequential representations to depict user interests from the cross-domain behavior sequences. To this end, we propose a dual dynamic graph modeling module that extracts the latent item embeddings from both local and global dynamic graphs.

3.3.1 Dual dynamic graphs. We first introduce dual dynamic graphs, i.e., local dynamic graphs and global dynamic graphs.

Local dynamic graphs. In this part, we apply a dynamic GNN to capture the intra-domain sequential transitions that represent user preference patterns in each single domain [38, 41]. The graph modeling has two steps: **Step 1**: local dynamic graph construction, and **Step 2**: local dynamic graph representation.

Step 1: local dynamic graph construction. We first introduce how to convert the single-domain behavior sequence into a dynamic graph. Taking domain A as example, a sequence can be represented by $S_A = [v_1, v_2, ..., v_i, ...]$, where v_i represents a consumed item of the user within the sequence S_A. The local dynamic graphs can be defined as $\{\mathcal{G}_A^1, \mathcal{G}_A^2, ..., \mathcal{G}_A^t, ...\}$, where $\mathcal{G}_A^t = \{\mathcal{V}_A^t, \mathcal{E}_A^t\}$ is the graph at snapshot t, $\mathcal{V}_A^t$ is the node set that indicates items, and $\mathcal{E}_A^t$ is the edge set that shows the transitions between items. When the user of the sequence S_A acts on item v_i at time t, an edge e is established from v_{i-1} to v_i in the graph $\mathcal{G}_A^t$. The local dynamic graph $\mathcal{G}_B^t$ in domain B could be constructed similarly.

Step 2: local dynamic graph representation. Then we describe how to achieve message propagations and generate the sequence embeddings in local dynamic graphs. A common way to model dynamic graphs is to have a separate GNN handle each snapshot of the graph and feed the output of each GNN to a peephole LSTM, like GCRN [28] and DyGGNN[31]. Since we aim to provide a general dual graph modeling structure for CDSR, the choice of GNN and LSTM is not our focus. In this special sequential setting, the challenge is how to encode the structural information as well as the sequential context of neighbors in the graph. Thus, we adopt SRGNN [38] here, a GNN that uses gated recurrent units (GRU) to unroll the recurrence of propagation for a fixed number of steps and then obtain the node embeddings after updating. In this way, the message passing process decides what information to be preserved and discarded from both graphical and sequential perspectives. After updating, the node embeddings can be denoted as $\mathbf{H}_{al}^t = \{\mathbf{h}_{al,1}^t, \mathbf{h}_{al,2}^t, ..., \mathbf{h}_{al,i}^t\}$, where $\mathbf{h}_{al,i}^t \in \mathbb{R}^D$ indicates the latent vector of the item v_i in $\mathcal{G}_A^t$. Later, we aggregate the item embeddings in the sequence with an attention mechanism which gives each item a specific weight that indicating their different influences to the current interest. Finally, the complete sequence representation is denoted as $\mathbf{SE}_{al}^t$. We simply describe the propagation and representation generation mechanism of SRGNN in Appendix A and more details can be found in [38]. Similarly, in the local dynamic graph $\mathcal{G}_B^t$ of domain B, we obtain the node embeddings as $\mathbf{H}_{bl}^t = \{\mathbf{h}_{bl,1}^t, \mathbf{h}_{bl,2}^t, ..., \mathbf{h}_{bl,j}^t\}$ and the sequence representation as $\mathbf{SE}_{bl}^t$.

Global dynamic graphs. Besides intra-domain transitions, it is also necessary to model inter-domain transitions to leverage cross-domain knowledge. To this end, we apply global dynamic graphs to capture diverse trends of user preferences in a shared feature space. We also introduce this part in two steps. For *step 1*, the *global dynamic graph construction* is similar as that of local dynamic graphs. The only difference is that we model all the items of two domains in a shared graph and the training target becomes
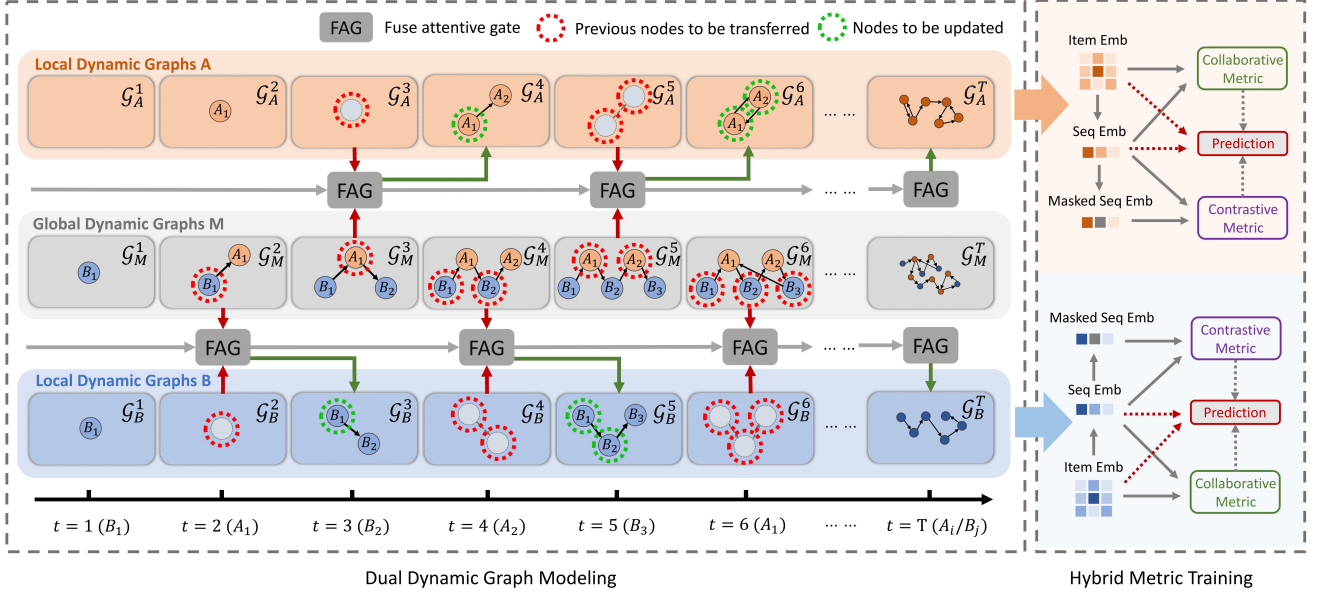
Figure 2: An overview of DDGHM . Local dynamic graph for domain A is shown in orange, domain B is shown in blue, and global dynamic graph is shown in gray. Section 3.2 contains a walkthrough of the model.

the merged sequence, such as $S_M = \{B_1, A_1, B_2, ..., A_i, B_j, ...\}$. As a user acts on the items from two domains, we add the directed edges accordingly, then consequently get a set of graphs modeling merged sequences as $\{\mathcal{G}_M^1, \mathcal{G}_M^2, ..., \mathcal{G}_M^T\}$. For **step 2**, the way to generate the *global dynamic graph representation* is the same as that of local dynamic graphs. We denote the node embeddings in $\mathcal{G}_M^t$ as $\mathbf{H}_g^t = \{\mathbf{h}_{g,1}^t, \mathbf{h}_{g,2}^t, ..., \mathbf{h}_{g,k}^t\}$. Then item embeddings from domain A and domain B in the global graph $\mathcal{G}_M^t$ could be extracted from $\mathbf{H}_g^t$, defined as $\mathbf{H}_{ag}^t = \{\mathbf{h}_{ag,1}^t, \mathbf{h}_{ag,2}^t, ..., \mathbf{h}_{ag,i}^t\}$ and $\mathbf{H}_{bg}^t = \{\mathbf{h}_{bg,1}^t, \mathbf{h}_{bg,2}^t, ..., \mathbf{h}_{bg,j}^t\}$, respectively. Finally, the sequence embeddings of the global dynamic graph from two domains can be obtained as $\mathbf{SE}_{ag}^t$ and $\mathbf{SE}_{bg}^t$.

3.3.2 Fuse attentive gate. We then introduce how to build the transferring bridge between local graphs and global graphs. The purposes of transferring lay in two folds : (1) users' general evolving interest patterns in local graphs and global graphs should be incorporated both in item-level and sequence-level. (2) Due to the structured encoding technique used in graph modeling, the influence of neighbours in the graph should also be considered to avoid the loss of context information.

To achieve these purposes, we innovatively design a fuse attentive gate (FAG), which is illustrated in Figure 3. The transferring procedure can be divided into four steps: 1) sequence-aware fusion, 2) self-attentive aggregation, 3) neighbour-attentive aggregation, and 4) updated state generation. Here we take the merged sequence $\{B_1, A_1, B_2, A_2, B_3, ...\}$ for example. At timestamp t, a direct edge from B_2 to B_3 is added into the local graph $\mathcal{G}_B^t$, and we can get the transferring node set, i.e., the nodes that exist both in $\mathcal{G}_B^{t-1}$ and $\mathcal{G}_M^{t-1}$, as $\{B_1, B_2\}$. First, we generate the sequence embeddings on the global graph and the local graph to achieve the *sequence-aware fusion*. Second, we apply *self-attentive aggregation* between $\{B_1, B_2\}$ themselves in $\mathcal{G}_M^{t-1}$ and $\mathcal{G}_B^{t-1}$. Third, we apply neighbour-attentive

aggregation between $\{B_1, B_2\}$ in $\mathcal{G}_B^{t-1}$ and their neighbour node set in $\mathcal{G}_M^{t-1}$, such as B_1 with its neighbour $\{A_1\}$ and B_2 with its neighbours $\{A_1, A_2\}$. Finally, the results of two aggregations are integrated by a GRU unit to complete the *updated state generation*. Taking the transfer between the global graph M and the local graph of domain B for example, we show details of each step as follows.

Sequence-aware fusion. Firstly, we take the sequence embedding $\mathbf{SE}_{bg}^{t-1}$ and $\mathbf{SE}_{bl}^{t-1}$ generated from the global graph $\mathcal{G}_M^{t-1}$ and the local graph $\mathcal{G}_B^{t-1}$ respectively to obtain the sequence-aware attention weight: $\mathbf{W}_{att}^t = \mathbf{W}_{gen}^t[\mathbf{SE}_{bg}^{t-1}, \mathbf{SE}_{bl}^{t-1}] + \mathbf{b}_{gen}^t$.

To efficiently and accurately fuse the structural information, we consider the feature aggregation in two perspectives, i.e., *self-attentive aggregation* and *neighbour-attentive aggregation*.

Self-attentive aggregation. The self-attentive aggregation is designed to retain the self-consistency information by transferring self-feature of all nodes in $\mathcal{G}_B^{t-1}$ from the global space to the local space. We take the local embedding $\mathbf{h}_{bl,x}^{t-1}$ of a node x from $\mathcal{G}_B^{t-1}$ and its global embedding $\mathbf{h}_{bg,x}^{t-1}$ from $\mathcal{G}_M^{t-1}$, then apply the self-attention:

$$
\begin{aligned}
\alpha_{self-l} &= v^\top \sigma(\mathbf{W}_{att}^t \mathbf{h}_{bl,x}^{t-1} + \mathbf{W}_{self}^t \mathbf{h}_{bl,x}^{t-1}) \\
\alpha_{self-g} &= v^\top \sigma(\mathbf{W}_{att}^t \mathbf{h}_{bg,x}^{t-1} + \mathbf{W}_{self}^t \mathbf{h}_{bl,x}^{t-1}) \\
\widetilde{\mathbf{h}_{bl,x}^t} &= \alpha_{self-l} \cdot \mathbf{h}_{bl,x}^{t-1} + \alpha_{self-g} \cdot \mathbf{h}_{bg,x}^{t-1},
\end{aligned}
\tag{1}
$$

where α_{self-l} and α_{self-g} are attention coefficients for local features and global features respectively, and $\widetilde{\mathbf{h}_{bl,x}^t}$ denotes the aggregated self-aware feature of the node x.

Neighbour-attentive aggregation. The neighbour-attentive aggregation aims to transfer context information of neighbours from the global graph to the local graph. To do this, we generate the first-order neighbour set $N(bg, x)$ for each node x in the transferring
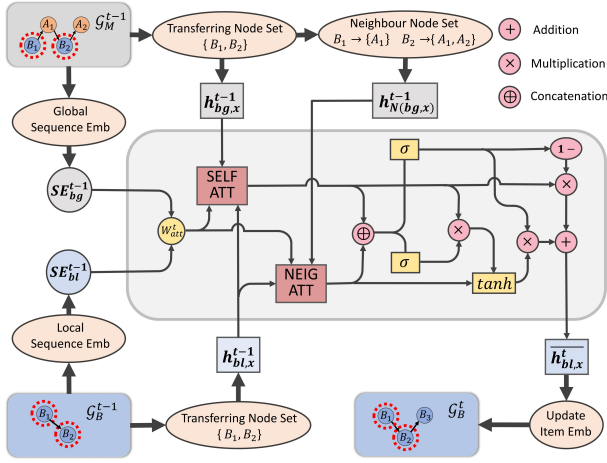
Figure 3: The structure of the fuse attentive gate. Here, we take the transfer between the global graph M and the local graph of domain B as an example.

node set. After that, we apply the neighbour-attention between the node x and its first-order neighbours in the global graph :

$$\alpha_{neig}^i = \frac{exp((\mathbf{h}_{bg,i}^{t-1})^\top \mathbf{W}_{nei}^t \mathbf{h}_{bl,x}^{t-1})}{\sum_{i \in N(bg,x)} exp((\mathbf{h}_{bg,i}^{t-1})^\top \mathbf{W}_{nei}^t \mathbf{h}_{bl,x}^{t-1})},$$

$$\widehat{\mathbf{h}_{bl,x}^t} = \sum_{i \in N(bg,x)} \alpha_{neig}^i \cdot \mathbf{W}_{att}^t \mathbf{h}_{bg,i}^{t-1}. \quad (2)$$

Here, α_{neig}^i indicates the attention coefficients for different neighbours and $\widehat{\mathbf{h}_{bl,x}^t}$ denotes the aggregated neighbour-aware feature of the node x.

Updated state generation. With $\widehat{\mathbf{h}_{bl,x}^t}$ and $\widetilde{\mathbf{h}_{bl,x}^t}$, we then employ a GRU-based gating mechanism as the recurrent unit:

$$\begin{aligned}
z_{gate} &= \sigma(\mathbf{W}_{gz}[\widehat{\mathbf{h}_{bl,x}^t}, \widetilde{\mathbf{h}_{bl,x}^t}]), \\
r_{gate} &= \sigma(\mathbf{W}_{gr}[\widehat{\mathbf{h}_{bl,x}^t}, \widetilde{\mathbf{h}_{bl,x}^t}]), \\
\mathbf{h}_{xg} &= \tanh(\mathbf{W}_{gh}[\widehat{\mathbf{h}_{bl,x}^t}, r_{gate} \odot \widetilde{\mathbf{h}_{bl,x}^t}], \\
\overline{\mathbf{h}_{bl,x}^t} &= (1 - z_{gate}) \odot \widetilde{\mathbf{h}_{bl,x}^t} + z_{gate} \odot \mathbf{h}_{xg},
\end{aligned} \quad (3)$$

where $\mathbf{W}_{gz}$, $\mathbf{W}_{gr}$, $\mathbf{W}_{gh}$ are weight matrices, and $\overline{\mathbf{h}_{bl,x}^t}$ is the output of the gate.

With dual dynamic graphs and fuse attentive gates, we complete extracting item and sequence embeddings of both domains. The thorough algorithm of this module is show in Appendix B.

3.4 Hybrid Metric Training

With the final sequence representations from domain A and B, the hybrid metric training module makes the prediction. This module mainly has two purposes, including (1) exploring user preferences for different items in two domains, which is the prime purpose of the model, (2) alleviating the remaining data sparsity problem. For the first purpose, we apply the basic prediction loss, similar as existing researches. To achieve the second purpose, we optimize the representation learning process with two enhancements, i.e., *collaborative metric* for retaining representation alignment and

contrastive metric for preserving uniformity, to further improve the recommendation performance.

Basic prediction. Motivated by previous studies [4, 18, 30], we calculate the matching between sequence-level embeddings $\mathbf{SE}_{al,S_A}$, $\mathbf{SE}_{bl,S_B}$ and item embedding matrices $\mathbf{H}_{al}$, $\mathbf{H}_{bl}$ in corresponding domains, to compute the recommendation probabilities:

$$\begin{aligned}
P(A_{i+1}|S_A, S_B, S_M) &= softmax(\mathbf{H}_{al} \cdot \mathbf{SE}_{al,S_A}^\top + \mathbf{b}_A), \\
P(B_{j+1}|S_B, S_A, S_M) &= softmax(\mathbf{H}_{bl} \cdot \mathbf{SE}_{bl,S_B}^\top + \mathbf{b}_B),
\end{aligned} \quad (4)$$

where $\mathbf{b}_A$ and $\mathbf{b}_B$ are bias items. Then the negative log-likelihood loss function is employed as follows:

$$\begin{aligned}
\mathcal{L}_A &= -\frac{1}{|\mathbb{S}|} \sum_{S_A \in \mathbb{S}} \sum_{A_i \in S_A} \log P(A_{i+1}|S_A, S_B, S_M), \\
\mathcal{L}_B &= -\frac{1}{|\mathbb{S}|} \sum_{S_B \in \mathbb{S}} \sum_{B_j \in S_B} \log P(B_{j+1}|S_B, S_A, S_M),
\end{aligned} \quad (5)$$

where $\mathbb{S}$ denotes the training sequences in both domains.

Collaborative metric. The collaborative metric aims to achieve representation alignment, which means that similar instances are expected to have similar representations. Here, we model the observed behavior sequences as a set of positive user-item pairs $\mathcal{T}$ and learn a user-item joint metric to encode these relationships [8]. The learned metric tries to pull the pairs in $\mathcal{T}$ closer and push the other pairs further apart. Specifically, we learn the model parameters by optimizing the large margin nearest neighbor objective [22]:

$$\mathcal{L}_{col} = \sum_{(i,j) \in \mathcal{T}} \sum_{(i,k) \notin \mathcal{T}} \mathbf{W}_{ij}^{rank} \cdot max(m + D(i,j)^2 - D(i,k)^2, 0), \quad (6)$$

where j is an item in the historical sequences of user i, k is an item that never appears in these sequences. $D(i,j)$ is defined as the Euclidean distance between the sequence embedding of user i and the embedding of item j. Specifically, $\mathbf{W}_{ij}^{rank}$ denotes the Weighted Approximate-Rank Pairwise (WARP) loss [36], calculated as $log(rank_D(i,j) + 1)$ where $rank_D(i,j)$ indicates the rank of item j in user i's recommendations. Following [8], we empirically set the safety margin m to 1.5 in our experiments.

Contrastive metric. The contrastive metric aims to realize representation uniformity, which means that the distribution of representations is preferred to preserve as much information as possible. To do this, the contrastive loss minimizes the difference between the augmented and the original views of the same user historical sequence, and meanwhile maximizes the difference between the augmented sequences derived from different users.

Here we apply a random *Item Mask* as the augmentation operator. We take domain A for example. For each user historical sequence $S_{A,u}$, we randomly mask a proportion of items and denote the masked sequence as $S_{A,u}^{mask}$. For a mini-batch of N users $\{u_1, u_2, ...u_N\}$, we apply augmentation to each user's sequence and obtain $2N$ sequences $\left[S_{A,u_1}, S_{A,u_1}^{mask}, S_{A,u_2}, S_{A,u_2}^{mask}, ..., S_{A,u_n}, S_{A,u_n}^{mask} \right]$. For each user u, we treat $(S_{A,u}, S_{A,u}^{mask})$ as the positive pair and treat other $2(N-1)$ examples as negative samples. We utilize dot product to measure the similarity between each representation, $sim(u, v) = u^\top v$. Then the contrastive loss function is defined using

Table 1: Dataset statistics of Amazon.

Movie & Book			Food & Kitchen		
Movie	#Items	58,371	*Food*	#Items	47,520
Book	#Items	104,895	*Kitchen*	#Items	53,361
#Overlapped-users		12,746	#Overlapped-users		8,575
#Sequences		158,373	#Sequences		67,793
#Training-sequences		118,779	#Training-sequences		50,845
#Validation-sequences		23,756	#Validation-sequences		10,169
#Test-sequences		15,838	#Test-sequences		6,779
Sequence Avg Length		32.1	Sequence Avg Length		18.7

the widely used softmax cross entropy loss as:

$$\mathcal{L}_{con} = \sum_{\mathcal{P} \in \mathbb{P}} -\frac{1}{|\mathcal{P}|} \sum_{(x,y) \sim \mathcal{P}} log \frac{exp(sim(\mathbf{SE}_x, \mathbf{SE}_y))}{\sum_{x' \in \{y\} \cup \mathcal{X}_{\mathcal{P}}^-} exp(sim(\mathbf{SE}_x, \mathbf{SE}_{x'}))}, \quad (7)$$

where $\mathbb{P}$ is the set of positive pairs, $\mathcal{X}_{\mathcal{P}}^-$ is the sampled negative set for $\mathcal{P}$.

Putting together. Finally, we combine the basic prediction loss of both domains with collaborative and contrastive metrics via a multi-task training scheme:

$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_B + \lambda_{col}\mathcal{L}_{col} + \lambda_{con}\mathcal{L}_{con}, \quad (8)$$

where λ_{col} and λ_{con} are hyper-parameters to balance different types of losses.

4 EXPERIMENTS AND ANALYSIS

In this section, we conduct experiments to answer the following research questions: **RQ1**: How does our model perform compared with the state-of-the-art SR, CDR, and CDSR methods? **RQ2**: How does each component of the dual dynamic graph modeling module contribute to the final performance? **RQ3**: How does each component of the hybrid metric training module contribute to the final performance? **RQ4**: How does our model perform with different sequence lengths, i.e., with different data sparsity degree? **RQ5**: How do hyper-parameters affect our model performance?

4.1 Experimental Setup

Datasets. We conduct experiments on the Amazon dataset [20], which consists of user interactions (e.g. userid, itemid, ratings, timestamps) from multiple domains. Compared with other recommendation datasets, the Amazon dataset contains overlapped user interactions in different domain and is equipped with adequate sequential information, and thus is commonly used for CDSR research [4, 14, 18]. Specifically, we pick two pairs of complementary domains "**Movie** & **Book**" and "**Food** & **Kitchen**" for experiments. Following the data preprocessing method of [17, 18], we first pick the users who have interactions in both domains and then filter the users and items with less than 10 interactions. After that, we order and split the sequences from each user into several small sequences, with each sequence containing interactions within a period, i.e, three months for "Movie&Book" dataset and two years for "Food&Kitchen" dataset. We also filter the sequences that contain less than five items in each domain. The detailed statistics of these datasets are shown in Table 1.

Evaluation method. Following [4, 18, 30], we use the latest interacted item in each sequence as the ground truth. We randomly

select 75% of the sequences as the training set, 15% as the validation set, and the remaining 10% as the test set. We choose three evaluation metrics, i.e., Hit Rate (HR), Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR), where we set the cut-off of the ranked list as 5, 10, and 20. For all the experiments, we repeat them five times and report the average results.

Parameter settings. For a fair comparison, we choose Adam [11] as the optimizer, and tune the parameters of **DDGHM** and the baseline models to their best values. Specifically, we study the effect of the hidden dimension D by varying it in $\{16, 32, 64, 128, 256, 512\}$, and the effects of hyper-parameters λ_{col} and λ_{con} by varying them in $\{0.1, 0.3, 0.5, 0.7, 1, 3, 5, 10\}$. And we set the batch size $N = 100$.

4.2 Comparison Methods

Traditional recommendations. POP is the simplest baseline that ranks items according to their popularity judged by the number of interactions. **BPR-MF** [25] optimizes the matrix factorization with implicit feedback using a pairwise ranking loss. **Item-KNN** [27] recommends the items that are similar to the previously interacted items in the sequence, where similarity is defined as the cosine similarity between the vectors of sequences.

Sequential recommendations. GRU4REC [7] uses GRU to encode sequential information and employs ranking-based loss function. **BERT4Rec** [29] adopts a bi-directional transformer to extract the sequential patterns. **SRGNN** [38] obtains item embeddings through a gated GNN layer and then uses a self-attention mechanism to compute the session level embeddings. **CL4SRec** [39] uses item cropping, masking, and reordering as augmentations for contrastive learning on interaction sequences.

Cross-Domain Recommendations. NCF-MLP++ is a deep learning based method which uses multilayer perceptron (MLP) [5] to learn the inner product in the traditional collaborative filtering process. We adopt the implementation in [18]. **Conet** [9] enables dual knowledge transfer across domains by introducing a cross connection unit from one base network to the other and vice versa. **DDTCDR** [15] introduces a deep dual transfer network that transfers knowledge with orthogonal transformation across domains. **DARec** [40] transfers knowledge between domains with shared users, learning shared user representations across different domains via domain adaptation technique.

Cross-domain sequential recommendations. π-Net [18] proposes a novel parallel information-sharing network with a shared account filter and a cross-domain transfer mechanism to simultaneously generate sequential recommendations for two domains. **PSJNet** [30] is a variant of π-Net, which learns cross-domain representations by extracting role-specific information and combining useful user behaviors. **DA-GCN** [4] devises a domain-aware graph convolution network with attention mechanisms to learn user-specific node representations under the cross-domain sequential setting. **DAT-MDI** [1] uses a potential mapping method based on a slot attention mechanism to extract the user's sequential preferences in both domains. **DASL** [14] is a dual attentive sequential learning model, consisting of dual embedding and dual attention modules to match the extracted embeddings with candidate items.

Table 2: Experimental results on Amazon datasets.

	Movie-domain						Book-domain					
	HR@5	NDCG@5	MRR@5	HR@20	NDCG@20	MRR@20	HR@5	NDCG@5	MRR5	HR@20	NDCG@20	MRR@20
POP	.0107	.0052	.0044	.0165	.0058	.0047	.0097	.0033	.0026	.0146	.0041	.0035
BPR-MF	.0479	.0402	.0254	.0584	.0430	.0373	.0374	.0320	.0245	.0546	.0385	.0304
Item-KNN	.0745	.0513	.0428	.0758	.0557	.0456	.0579	.0442	.0367	.0663	.0495	.0420
GRU4REC	.2232	.1965	.1772	.2456	.2078	.1830	.2012	.1648	.1498	.2145	.1672	.1520
SR-GNN	.2362	.2073	.1836	.2592	.2289	.1950	.2189	.1733	.1578	.2254	.1745	.1611
BERT4Rec	.2425	.2104	.1989	.2637	.2372	.2146	.2203	.1820	.1655	.2274	.1834	.1704
CL4SRec	.2547	.2231	.2035	.2740	.2418	.2197	.2325	.1944	.1701	.2382	.1987	.1798
NCF-MLP++	.1038	.0679	.0557	.1283	.0746	.0675	.0972	.0550	.0484	.1080	.0607	.0535
Conet	.1247	.0742	.0628	.1354	.0818	.0732	.1130	.0611	.0545	.1205	.0649	.0569
DDTCDR	.1328	.0933	.0840	.1597	.1040	.0996	.1243	.0868	.0688	.1548	.0912	.0735
DARec	.1564	.1179	.1053	.1796	.1283	.1164	.1475	.1089	.0849	.1735	.1122	.1004
DAT-MDI	.2453	.2121	.2043	.2614	.2340	.2124	.2205	.1838	.1676	.2282	.1865	.1735
π-Net	.2665	.2243	.2108	.2803	.2565	.2433	.2385	.2024	.1815	.2433	.2134	.2042
PSJNet	.2771	.2294	.2156	.2917	.2611	.2478	.2430	.2093	.1879	.2609	.2311	.2110
DASL	.2835	.2314	.2225	.3012	.2682	.2510	.2479	.2133	.1937	.2694	.2382	.2201
DA-GCN	.2821	.2342	.2276	.2989	.2719	.2615	.2392	.2156	.2038	.2658	.2430	.2327
DDGHM-L	.2674	.2241	.2139	.2905	.2627	.2518	.2435	.2126	.1898	.2526	.2219	.2176
DDGHM-G	.2320	.1978	.1744	.2542	.2175	.1901	.2092	.1695	.1510	.2192	.1709	.1589
DDGHM-GA	.2895	.2423	.2329	.3068	.2716	.2648	.2486	.2234	.2004	.2673	.2414	.2321
DDGHM	**.3083**	**.2517**	**.2431**	**.3257**	**.2892**	**.2734**	**.2615**	**.2379**	**.2198**	**.2890**	**.2592**	**.2448**

	Food-domain						Kitchen-domain					
	HR@5	NDCG@5	MRR@5	HR@20	NDCG@20	MRR@20	HR@5	NDCG@5	MRR5	HR@20	NDCG@20	MRR@20
POP	.0085	.0048	.0031	.0123	.0052	.0035	.0087	.0053	.0034	.0152	.0117	.0074
BPR-MF	.0298	.0237	.0195	.0386	.0340	.0255	.0349	.0297	.0261	.0422	.0354	.0285
Item-KNN	.0572	.0389	.0299	.0685	.0443	.0396	.0590	.0453	.0428	.0702	.0516	.0463
GRU4REC	.1845	.1533	.1407	.2076	.1648	.1520	.1974	.1608	.1521	.2149	.1807	.1685
SR-GNN	.1987	.1735	.1558	.2196	.1902	.1693	.2254	.1969	.1811	.2390	.2078	.1874
BERT4Rec	.2092	.1824	.1605	.2247	.2013	.1745	.2307	.2035	.1898	.2432	.2109	.1926
CL4SRec	.2153	.1918	.1731	.2350	.2145	.1876	.2415	.2163	.1979	.2582	.2234	.2065
NCF-MLP++	.0801	.0565	.0433	.1043	.0672	.0548	.0947	.0680	.0573	.1166	.0889	.0654
Conet	.0955	.0730	.0544	.1138	.0876	.0685	.1069	.0854	.0712	.1244	.1015	.0830
DDTCDR	.1134	.0975	.0728	.1259	.1026	.0759	.1268	.1024	.0845	.1398	.1156	.0985
DARec	.1210	.1025	.0899	.1304	.1118	.0925	.1331	.1146	.0933	.1445	.1280	.1053
DAT-MDI	.2123	.1899	.1685	.2375	.2144	.1830	.2348	.2102	.1956	.2514	.2188	.2013
π-Net	.2348	.2054	.1834	.2469	.2286	.1945	.2575	.2220	.2135	.2644	.2290	.2206
PSJNet	.2430	.2118	.1905	.2552	.2347	.2039	.2635	.2389	.2296	.2718	.2459	.2375
DASL	.2572	.2248	.2145	.2689	.2430	.2255	.2743	.2515	.2483	.2869	.2608	.2501
DA-GCN	.2524	.2197	.2089	.2664	.2415	.2240	.2785	.2593	.2495	.2926	.2682	.2574
DDGHM-L	.2360	.2078	.1897	.2522	.2304	.2001	.2599	.2386	.2285	.2714	.2448	.2383
DDGHM-G	.2187	.1851	.1644	.2282	.2199	.1902	.2404	.2167	.2038	.2569	.2235	.2072
DDGHM-GA	.2632	.2348	.2159	.2677	.2426	.2250	.2764	.2582	.2397	.2816	.2605	.2444
DDGHM	**.2745**	**.2431**	**.2216**	**.2789**	**.2578**	**.2364**	**.2840**	**.2627**	**.2571**	**.3043**	**.2736**	**.2660**

4.3 Model Comparison (for RQ1)

General comparison. We report the comparison results on two datasets **Movie** & **Book** and **Food** & **Kitchen** in Table 2. For space save, we report the results when the cut-off of the ranked list is 10 in Appendix C. From the results, we can find that: (1) **DDGHM** outperforms **SR** baselines, indicating that our model can effectively capture and transfer the cross-domain information to promote the recommendation performance of a single domain. (2) Comparing **DDGHM** with **CDR** baselines, the improvement in both domains on two datasets is also evident. It proves the effectiveness of **DDGHM** in dynamically modeling the intra-domain and the inter-domain sequential transitions with a graphical framework. (3) Compared with **CDSR** baselines, **DDGHM** achieves better performance in terms of all metrics, which demonstrates that **DDGHM** is able to extract user preferences more accurately by leveraging complementary information from both domains.

4.4 In-depth Analysis (for RQ2-RQ5)

Study of the dual dynamic graph modeling (RQ2). To study how each component of the dual dynamic graph modeling module contributes to the final performance, we compare **DDGHM** with its

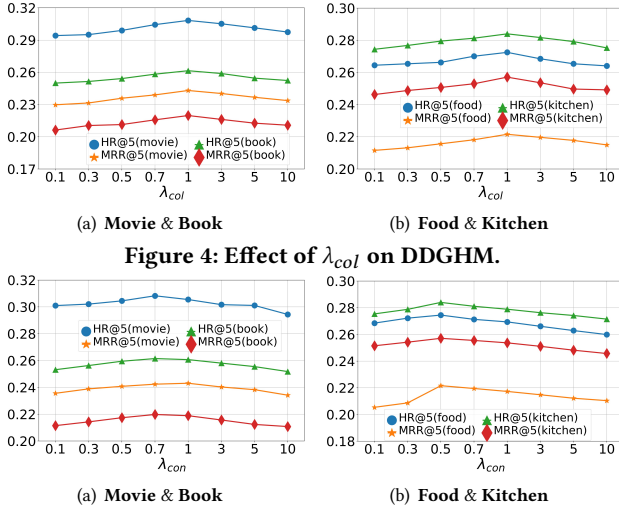**Table 3: Model performances of different sequence lengths.**

	Movie-domain			Book-domain		
	L=10.7	L=21.5	L=32.1	L=10.7	L=21.5	L=32.1
CL4SRec	.1192	.1848	.2740	.1063	.1679	.2382
DARec	.0925	.1139	.1796	.0832	.1313	.1735
DASL	.1352	.2528	.3012	.1372	.2128	.2694
DA-GCN	.1467	.2496	.2989	.1391	.2119	.2658
DDGHM	**.1635**	**.2766**	**.3257**	**.1548**	**.2342**	**.2890**
Imp.	11.5%	9.41%	8.13%	11.3%	10.1%	7.28%

several variants, including **DDGHM**-L, **DDGHM**-G, and **DDGHM**-GA. (a) **DDGHM**-L applies only the local graphs in dual dynamic graph modeling to extract item and sequence embeddings. (b) **DDGHM**-G applies only the global graph. (c) **DDGHM**-GA retains the dual dynamic graph structure, but replaces FAG with a simple GRU as cross-domain transfer unit (CTU) [18]. The comparison results are also shown in Table 2. From it, we can observe that: (1) Both **DDGHM**-L and **DDGHM**-G perform worse than **DDGHM**, showing that integrating intra-domain transitions with inter-domain transitions can boost the performance. (2) **DDGHM**-L outperforms **DDGHM**-G on two datasets, indicating that intra-domain collaborative influences still take the dominant place when extracting user preferences. And this is because that simply encoding inter-domain

Table 4: Ablation results on two Amazon datasets.

	Movie-domain		Book-domain	
	HR@20	NDCG@20	HR@20	NDCG@20
DDGHM-col	.3159	.2812	.2786	.2521
DDGHM-con	.3205	.2848	.2835	.2554
DDGHM-cc	.3045	.2786	.2704	.2489
DDGHM	**.3257**	**.2892**	**.2890**	**.2592**
	Food-domain		Kitchen-domain	
	HR@20	NDCG@20	HR@20	NDCG@20
DDGHM-col	.2740	.2518	.2959	.2698
DDGHM-con	.2764	.2540	.2998	.2704
DDGHM-cc	.2712	.2473	.2958	.2695
DDGHM	**.2789**	**.2578**	**.3043**	**.2736**

(a) **Movie** & **Book** (b) **Food** & **Kitchen**

Figure 4: Effect of λ_{col} on DDGHM.

(a) **Movie** & **Book** (b) **Food** & **Kitchen**

Figure 5: Effect of λ_{con} on DDGHM.

transitions in a shared latent space ignores the domain feature distribution bias. (3) **DDGHM** shows superiority over **DDGHM**-GA, which proves that the FAG completes transferring more effectively than other CTUs in existing research and is especially applicable in the dual dynamic graphical model.

Study of the hybrid metric training (RQ3). In order to verify the effectiveness of each component of the hybrid metric training module, we further conduct more ablation studies, where (a) **DDGHM**-col only employs collaborative metric, (b) **DDGHM**-con only applies contrastive metric, and (c) **DDGHM**-cc trains the recommendation model without either metric. The results are shown in Table 4, from which we can conclude that: (1) Both **DDGHM**-con and **DDGHM**-col perform worse than **DDGHM**. The reasons are two-folds. On the one hand, employing collaborative metric loss augments the similarity of similar instance embeddings thus achieving representation alignment. On the other hand, adopting contrastive metric loss preserves the discrepancy of different feature distributions thus retaining representation uniformity. Both of them play an important role in enhancing the representations. (2) **DDGHM** significantly outperforms **DDGHM**-cc, which indicates that the combination of two metrics effectively optimizes the training module and further promotes model performance.

Influence of sequence length (RQ4). We report the comparison results of HR@20 under different sequence average length settings in Table 3. Here, for space save, we only report the results of several baselines which perform best in each category, i.e., SR, CDR, and
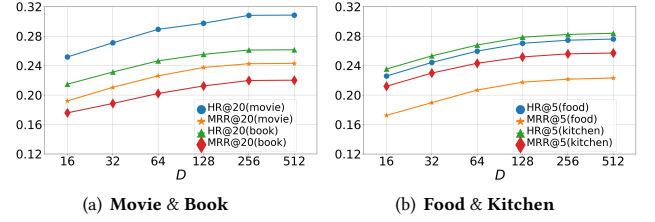

(a) **Movie** & **Book** (b) **Food** & **Kitchen**

Figure 6: Effect of embedding dimension D on DDGHM.

CDSR. And L means the average length of sequences. From it, we can observe that in both domains, the shorter the average length of sequences is, the grater improvement our model brings. Therefore, **DDGHM** is able to effectively alleviate the data sparsity problem and performs well even when sequences are short.

Parameter analysis (RQ5). We now study the effects of hyper-parameters on model performance, including λ_{col}, λ_{con}, and D. We first study the effects of λ_{col} and λ_{con} on the model, varying them in $\{0.1, 0.3, 0.5, 0.7, 1, 3, 5, 10\}$, and then report the results in Fig. 4 and Fig. 5. The bell-shaped curves show that the accuracy will first gradually increase with λ_{col} or λ_{con} and then slightly decrease. We can conclude that when λ_{col} and λ_{con} approach 0, the collaborative loss and contrastive loss cannot produce positive effects. But when λ_{col} and λ_{con} become too large, the metric loss will suppress the negative log-likelihood loss, which also reduces the recommendation accuracy. Empirically, we choose $\lambda_{col} = 1.0$ and $\lambda_{con} = 0.7$ on the dataset **Movie** & **Book** while $\lambda_{col} = 1.0$ and $\lambda_{con} = 0.5$ on the dataset **Food** & **Kitchen**. Finally we perform experiments by varying dimension D in range $\{16, 32, 64, 128, 256, 512\}$. The results on two datasets are shown in Fig. 6. From them, we can see that the performance gradually improves when D increases and finally keeps a fairly stable level after D reaches 256. It indicates that a larger embedding dimension can provide more accurate embeddings for items thus enriching the representations of user preferences. Since a too large embedding dimension will cause huge computational and time cost, we choose $D = 256$ here for both datasets.

5 CONCLUSION

In this paper, we propose **DDGHM**, which includes a dual dynamic graph modeling module and a hybrid metric training module, for solving the Cross-Domain Sequential Recommendation (CDSR) problem. In the dual dynamic graph modeling module, we firstly construct dual dynamic graphs, i.e., global graphs and local graphs, to explore intra-domain and inter-domain transitions, and then adopt a fuse attentive gating mechanism to adaptively integrate them. In the hybrid metric training module, we apply the representation enhancement from two aspects, i.e., collaborative metric for alignment and contrastive metric for uniformity, so that the remaining data sparsity impact in CDSR is alleviated. Extensive experiments conducted on two real-world datasets illustrate the effectiveness of **DDGHM** and the contribution of each component.

ACKNOWLEDGMENTS

REFERENCES

[1] Chen Chen, Jie Guo, and Bin Song. 2021. Dual attention transfer in session-based recommendation with multi-dimensional integration. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 869–878.

[2] 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 Conf. 2022*. 1455–1465.

[3] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*. 152–160.

[4] 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).

[5] 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.

[6] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

[7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[8] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*. 193–201.

[9] 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.

[10] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE Int'l Conf. on Data Mining (ICDM)*. IEEE, 197–206.

[11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] 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*.

[13] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4189–4196.

[14] Pan Li, Zhichao Jiang, Maofei Que, Yao Hu, and Alexander Tuzhilin. 2021. Dual Attentive Sequential Learning for Cross-Domain Click-Through Rate Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3172–3180.

[15] 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.

[16] 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 Conf. 2022*. 1181–1190.

[17] Muyang Ma, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Lifan Zhao, Peiyu Liu, Jun Ma, and Maarten de Rijke. 2022. Mixed Information Flow for Cross-domain Sequential Recommendations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 4 (2022), 1–32.

[18] 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 Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. 685–694.

[19] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach.. In *IJCAI*, Vol. 17. 2464–2470.

[20] 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*. 43–52.

[21] Weike Pan and Qiang Yang. 2013. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence* 197 (2013), 39–55.

[22] Chanyoung Park, Donghyun Kim, Xing Xie, and Hwanjo Yu. 2018. Collaborative translational metric learning. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 367–376.

[23] Ruihong Qiu, Zi Huang, and Hongzhi Yin. 2021. Memory Augmented Multi-Instance Contrastive Predictive Coding for Sequential Recommendation. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 519–528.

[24] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In

[25] *Proceedings of the 28th ACM international conference on information and knowledge management*. 579–588.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[27] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[28] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*. Springer, 362–373.

[29] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[30] Wenchao Sun, Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2021. Parallel Split-Join Networks for Shared Account Cross-domain Sequential Recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[31] Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. 2019. Learning to represent the evolution of dynamic graphs with recurrent models. In *Companion proceedings of the 2019 world wide web conference*. 301–307.

[32] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[33] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In *Proceedings of the eleventh ACM conference on recommender systems*. 138–146.

[34] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1101–1110.

[35] Yaqing Wang, Chunyan Feng, Caili Guo, Yunfei Chu, and Jenq-Neng Hwang. 2019. Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 717–725.

[36] Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning* 81, 1 (2010), 21–35.

[37] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[38] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[39] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive learning for sequential recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[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] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[42] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 229–238.

[43] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2020. Dgtn: Dual-channel graph transition network for session-based recommendation. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 236–242.

[44] Yujia Zheng, Siyi Liu, and Zailei Zhou. 2019. Balancing multi-level interactions for session-based recommendation. *arXiv preprint arXiv:1910.13527* (2019).

[45] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. Dtcdr: A framework for dual-target cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1533–1542.

[46] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-Domain Recommendation: Challenges, Progress, and Prospects. (2021), 4721–4728.

[47] Feng Zhu, Yan Wang, Jun Zhou, Chaochao Chen, Longfei Li, and Guanfeng Liu. 2021. A Unified Framework for Cross-Domain and Cross-System Recommendations. *TKDE* (2021).

A PROPAGATION MECHANISM OF SRGNN

In the local dynamic graphs and the global dynamic graph, we adopt SRGNN [38] to achieve message propagation. Here, we take the local dynamic graph of the domain A for example, a sequence in it can be represented by $S_A = [v_1, v_2, ..., v_i, ...]$, where v_i represents a consumed item of the user within the sequence S_A. We embed every item v in domain A into a unified embedding space and use a node vector $\mathbf{h}_{al} \in \mathbb{R}^D$ to denote the latent vector of the item v learned via local dynamic graphs in domain A, with D denoting the dimensionality. For the node v_i of the graph $\mathcal{G}_A^t$, the recurrence of the propagation functions is given as follows:

$$a_i^k = \mathbf{A}_{i:}^t [\mathbf{h}_{al,1}^{k-1}, ..., \mathbf{h}_{al,n}^{k-1}]^\top \mathbf{W}_h + \mathbf{b}_h, \tag{9}$$

$$z_i^k = \sigma(\mathbf{W}_z a_i^k + \mathbf{U}_z \mathbf{h}_{al,i}^{k-1}), \tag{10}$$

$$r_i^k = \sigma(\mathbf{W}_r a_i^k + \mathbf{U}_r \mathbf{h}_{al,i}^{k-1}), \tag{11}$$

$$\widetilde{\mathbf{h}_i^k} = \tanh(\mathbf{W}_o a_i^k + \mathbf{U}_o(r_i^k \odot \mathbf{h}_{al,i}^{k-1})), \tag{12}$$

$$\mathbf{h}_{al,i}^k = (1 - z_i^k) \odot \mathbf{h}_{al,i}^{k-1} + z_i^k \odot \widetilde{\mathbf{h}_i^k}, \tag{13}$$

where $[\mathbf{h}_{al,1}^{k-1}, ..., \mathbf{h}_{al,n}^{k-1}]$ is the node embedding list in the current sequence and k denotes the propagation step. The matrix $\mathbf{A}^t \in \mathbb{R}^{n \times 2n}$ is the concatenation of two adjacency matrices $\mathbf{A}_{in}^t$ and $\mathbf{A}_{out}^t$, which represents weighted connections of outgoing and incoming edges in the $\mathcal{G}_A^t$ at snapshot t, respectively. And $\mathbf{A}_{i:}^t$ are the two columns in $\mathbf{A}^t$ corresponding to node v_i. which describes how nodes in the graph communicate with each other. For example, consider a sequence $S_A = [v_1, v_2, v_3, v_1, v_4, ...]$, the dynamic graphs $\mathcal{G}_A^t$ and the corresponding matrix $\mathbf{A}^t$ are illustrated in Figure. 7. We select the snapshot $t = 4$ and $t = 5$ as examples, and every column in $\mathbf{A}^t$ is composed of the normalized outgoing and incoming edge weights.

Eq. (9) shows the step that passes information between different nodes of the graph via edges in both directions. After that, a GRU-like update, including update gate $\mathbf{z}_i$ in Eq. (10) and reset gate $\mathbf{r}_i$ in Eq. (11), is adopted to determine what information to be preserved and discarded respectively. $\sigma(\cdot)$ is the sigmoid function and $\odot$ is the element-wise multiplication operator. Then the candidate state is generated by the previous state and the current state under the control of the reset gate in Eq. (12). We combine the previous hidden state with the candidate state using updating mechanism and get the final state in Eq. (13).

After K steps of updating, we can obtain the embeddings of all nodes in $\mathcal{G}_A^t$ as $\mathbf{H}_{al}^t = \{\mathbf{h}_{al,1}^t, \mathbf{h}_{al,2}^t, ..., \mathbf{h}_{al,i}^t\}$. Then we adopt a strategy which attentively combines long-term and short-term preferences into the final sequential representation. As for long-term preference, we apply the soft-attention mechanism to measure the varying importance of previous items and then aggregate them as a whole:

$$\alpha_{al,k}^t = p_{al,t}^\top \sigma(\mathbf{W}_{a1}^t \mathbf{h}_{al,i}^t + \mathbf{W}_{a2}^t \mathbf{h}_{al,k}^t + \mathbf{c}^t),$$

$$\mathbf{SE}_{ac}^t = \sum_{k=1}^i \alpha_{a,k}^t \mathbf{h}_{al,k}^t, \tag{14}$$

where parameters $p_{a,t}^\top \in \mathbb{R}^D$ and $\mathbf{W}_{a1}^t, \mathbf{W}_{a2}^t \in \mathbb{R}^{D \times D}$ control the weights of item embedding vectors. As for *short-term preference*, we concatenate the last item embedding which represents the current interest of the user with the above sequence embedding after

---

Algorithm 1: Dual Dynamic Graph Modeling

Input: user's cross-domain behavior sequence
$\quad\quad S_M$;single-domain behavior sequences: S_A,S_B.
Output: Item embeddings in local domain graphs : $\mathbf{H}_{al}$,
$\quad\quad \mathbf{H}_{bl}$; Sequence embeddings : $\mathbf{SE}_{al,S_A}$, $\mathbf{SE}_{bl,S_B}$

1   Initialize node embeddings in local graphs as $\mathbf{H}_{al}^0$ and $\mathbf{H}_{bl}^0$.

2   Global node embeddings $\mathbf{H}_g^0 = \mathbf{H}_{al}^0 \oplus \mathbf{H}_{bl}^0$

3   **for** $t = 1$ to T **do**

4   Get the current item v_i, v_i's domain as X; previous item in S_M as v_{i-1}^M, in S_X as v_{i-1}^X; previous item set as V_{pre}.

5   **From local graph $\mathcal{G}_X^{t-1}$:**

6     $Emb(V_{pre}) = \mathbf{h}_{xl,pre}^{t-1}$.

7   Get local sequence embedding $\mathbf{SE}_{xl}^{t-1}$ with Eq. (6-7).

8   **From global graph $\mathcal{G}_M^{t-1}$:**

9   Select V_{pre}'s neighbor set as N_{xg}.

10    $Emb(V_{pre}) = \mathbf{h}_{xg,pre}^{t-1}$, $Emb(N_{xg}) = \mathbf{h}_{xg,neig}^{t-1}$.

11   Get global sequence embedding $\mathbf{SE}_{xg}^{t-1}$ with Eq. (6-7).

12   **Do Fuse Attentive Gating:**

13    $\overline{\mathbf{h}_{xl,pre}^{t-1}}$ = GATE $(\mathbf{h}_{xl,pre}^{t-1}, \mathbf{h}_{xg,pre}^{t-1}, \mathbf{h}_{xg,neig}^{t-1}, \mathbf{SE}_{xg}^{t-1}, \mathbf{SE}_{xl}^{t-1})$

14   **On local graph $\mathcal{G}_X^t$ do:**

15   Update V_{pre}'s embeddings to new state $\overline{\mathbf{h}_{xl,pre}^{t-1}}$

16   Add a directed edge from v_{i-1}^X to v_i.

17   Update nodes on graph $\mathcal{G}_X^t$ with Eq. (1-5).

18   **On global graph $\mathcal{G}_M^t$ do:**

19   Add a directed edge from v_{i-1}^M to v_i.

20   Update nodes on graph $\mathcal{G}_M^t$ with Eq. (1-5).

21   **end for**

22   Get sequence embeddings $\mathbf{SE}_{al,S_A}$, $\mathbf{SE}_{bl,S_B}$ with Eq. (6-7).

23   **return** $\mathbf{H}_{al}$, $\mathbf{H}_{bl}$; $\mathbf{SE}_{al,S_A}$, $\mathbf{SE}_{bl,S_B}$
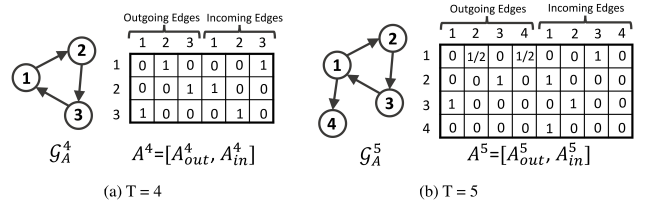

Figure 7: (a) and (b) are dynamic graphs and corresponding connection matrices.

aggregation and then take a linear transformation over them to generate the final sequence embedding:

$$\mathbf{SE}_{al}^t = \mathbf{W}_{a3}^t [\mathbf{SE}_{ac}^t; \mathbf{h}_{al,i}^t]. \tag{15}$$

B DUAL DYNAMIC GRAPH MODELING

We represent the thorough algorithm of the dual dynamic graph modeling module in Algorithm 1. The input of this module includes single-domain and cross-domain behavior sequences, and the output consists of item embeddings and sequence embeddings. The whole procedure of dual dynamic graph modeling can be divided into two parts, i.e., 1) dual dynamic graphs and 2) fuse attentive

Table 5: Experimental results on Amazon datasets when the cut-off of the ranked list is 10.

	Movie-domain			Book-domain			Food-domain			Kitchen-domain		
	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR
POP	.0125	.0056	.0045	.0115	.0038	.0031	.0097	.0049	.0033	.0128	.0089	.0050
BPR-MF	.0513	.0418	.0316	.0457	.0359	.0266	.0345	.0278	.0214	.0382	.0319	.0435
Item-KNN	.0752	.0527	.0433	.0604	.0468	.0395	.0614	.0412	.0341	.0628	.0497	.0443
GRU4REC	.2320	.2014	.1785	.2068	.1651	.1510	.1910	.1587	.1482	.2034	.1745	.1591
SR-GNN	.2468	.2159	.1923	.2226	.1738	.1599	.2048	.1834	.1569	.2298	.2014	.1856
BERT4Rec	.2510	.2248	.2034	.2245	.1825	.1673	.2115	.1932	.1683	.2385	.2084	.1910
CL4SRec	.2656	.2350	.2147	.2352	.1964	.1740	.2235	.2041	.1775	.2487	.2193	.2044
NCF-MLP++	.1142	.0718	.0614	.1015	.0578	. 0501	.0946	.0622	.0498	.1057	.0754	.0602
Conet	.1287	.0788	.0725	.1182	.0624	.0550	.1043	.0795	.0592	.1138	.0940	.0775
DDTCDR	.1433	.0984	.0936	.1327	.0893	.0712	.1182	.0980	.0736	.1299	.1087	.0923
DARec	.1635	.1223	.1099	.1608	.1097	.0922	.1278	.1074	.0906	.1396	.1221	.0981
DAT-MDI	.2526	.2235	.2097	.2247	.1848	.1698	.2237	.2054	.1728	.2458	.2163	.1989
π-Net	.2744	.2359	.2218	.2402	.2075	.1903	.2405	.2128	.1886	.2616	.2247	.2177
PSJNet	.2862	.2429	.2347	.2516	.2161	.2035	.2512	.2233	.1964	.2682	.2412	.2326
DASL	.2940	.2587	.2412	.2586	.2248	.2159	.2620	.2318	.2174	.2812	.2540	.2493
DA-GCN	.2925	.2624	.2477	.2543	.2294	.2198	.2589	.2294	.2138	.2856	.2617	.2533
DDGHM-L	.2786	.2425	.2297	.2468	.2183	.2099	.2415	.2187	.1934	.2655	.2410	.2330
DDGHM-G	.2495	.2133	.1862	.2144	.1701	.1547	.2275	.2089	.1749	.2488	.2208	.2062
DDGHM-GA	.2957	.2688	.2534	.2590	.2316	.2239	.2658	.2378	.2204	.2781	.2599	.2423
DDGHM	**.3148**	**.2745**	**.2630**	**.2754**	**.2418**	**.2343**	**.2762**	**.2515**	**.2284**	**.2931**	**.2689**	**.2642**

gate. We describe how to construct dual dynamic graphs in line 14-20, and introduce how to transfer cross-domain information by fuse attentive gating mechanism in line 5-13.

C EXPERIMENTAL RESULTS

Here, we additionally report the experimental results on two Amazon datasets when the cut-off of the ranked list is 10. The results show that: (1) **DDGHM** outperforms all the baselines of SR, CDR, and CDSR. (2) **DDGHM** also shows the superiority over its variants, i.e., **DDGHM**-L, **DDGHM**-G, and **DDGHM**-GA, indicating the effectiveness of each component in the dual dynamic graph modeling module.

Appendix: DDGHM: Dual Dynamic Graph with Hybrid Metric Training for Cross-Domain Sequential Recommendation

Anonymous Author(s)

A PROPAGATION MECHANISM OF SRGNN

In the local dynamic graphs and the global dynamic graph, we adopt SRGNN [?] to achieve message propagation. Here, we take the local dynamic graph of the domain A for example, a sequence in it can be represented by $S_A = [v_1, v_2, ..., v_i, ...]$, where v_i represents a consumed item of the user within the sequence S_A. We embed every item v in domain A into a unified embedding space and use a node vector $\mathbf{h}_{al} \in \mathbb{R}^D$ to denote the latent vector of the item v learned via local dynamic graphs in domain A, with D denoting the dimensionality. For the node v_i of the graph $\mathcal{G}_A^t$, the recurrence of the propagation functions is given as follows:

$$a_i^k = \mathbf{A}_{i:}^t [\mathbf{h}_{al,1}^{k-1}, ..., \mathbf{h}_{al,n}^{k-1}]^\top \mathbf{W}_h + \mathbf{b}_h, \quad (1)$$

$$z_i^k = \sigma(\mathbf{W}_z a_i^k + \mathbf{U}_z \mathbf{h}_{al,i}^{k-1}), \quad (2)$$

$$r_i^k = \sigma(\mathbf{W}_r a_i^k + \mathbf{U}_r \mathbf{h}_{al,i}^{k-1}), \quad (3)$$

$$\widetilde{\mathbf{h}_i^k} = \tanh(\mathbf{W}_o a_i^k + \mathbf{U}_o(r_i^k \odot \mathbf{h}_{al,i}^{k-1})), \quad (4)$$

$$\mathbf{h}_{al,i}^k = (1 - z_i^k) \odot \mathbf{h}_{al,i}^{k-1} + z_i^k \odot \widetilde{\mathbf{h}_i^k}, \quad (5)$$

where $[\mathbf{h}_{al,1}^{k-1}, ..., \mathbf{h}_{al,n}^{k-1}]$ is the node embedding list in the current sequence and k denotes the propagation step. The matrix $\mathbf{A}^t \in \mathbb{R}^{n \times 2n}$ is the concatenation of two adjacency matrices $\mathbf{A}_{in}^t$ and $\mathbf{A}_{out}^t$, which represents weighted connections of outgoing and incoming edges in the $\mathcal{G}_A^t$ at snapshot t, respectively. And $\mathbf{A}_{i:}^t$ are the two columns in $\mathbf{A}^t$ corresponding to node v_i. which describes how nodes in the graph communicate with each other. For example, consider a sequence $S_A = [v_1, v_2, v_3, v_1, v_4, ...]$, the dynamic graphs $\mathcal{G}_A^t$ and the corresponding matrix $\mathbf{A}^t$ are illustrated in Figure. 1. We select the snapshot $t = 4$ and $t = 5$ as examples, and every column in $\mathbf{A}^t$ is composed of the normalized outgoing and incoming edge weights.

Eq. (1) shows the step that passes information between different nodes of the graph via edges in both directions. After that, a GRU-like update, including update gate $\mathbf{z}_i$ in Eq. (2) and reset gate $\mathbf{r}_i$ in Eq. (3), is adopted to determine what information to be preserved and discarded respectively. $\sigma(\cdot)$ is the sigmoid function and $\odot$ is the element-wise multiplication operator. Then the candidate state is generated by the previous state and the current state under the control of the reset gate in Eq. (4). We combine the previous hidden state with the candidate state using updating mechanism and get the final state in Eq. (5).
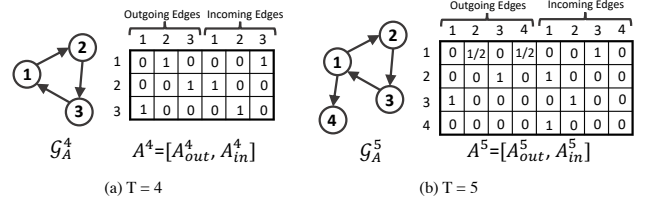
Figure 1: (a) and (b) are dynamic graphs and corresponding connection matrices.

Algorithm 1: Dual Dynamic Graph Modeling

Input: user's cross-domain behavior sequence S_M; single-domain behavior sequences: S_A, S_B.

Output: Item embeddings in local domain graphs : $\mathbf{H}_{al}$, $\mathbf{H}_{bl}$; Sequence embeddings : $\mathbf{SE}_{al,S_A}$, $\mathbf{SE}_{bl,S_B}$

1 Initialize node embeddings in local graphs as $\mathbf{H}_{al}^0$ and $\mathbf{H}_{bl}^0$.

2 Global node embeddings $\mathbf{H}_g^0 = \mathbf{H}_{al}^0 \oplus \mathbf{H}_{bl}^0$

3 **for** $t = 1$ to T **do**

4   Get the current item v_i, v_i's domain as X; previous item in S_M as v_{i-1}^M, in S_X as v_{i-1}^X; previous item set as V_{pre}.

5   **From local graph $\mathcal{G}_X^{t-1}$:**

6       $Emb(V_{pre}) = \mathbf{h}_{xl,pre}^{t-1}$.

7   Get local sequence embedding $\mathbf{SE}_{xl}^{t-1}$ with Eq. (6-7).

8   **From global graph $\mathcal{G}_M^{t-1}$:**

9   Select V_{pre}'s neighbor set as N_{xg}.

10      $Emb(V_{pre}) = \mathbf{h}_{xg,pre}^{t-1}$, $Emb(N_{xg}) = \mathbf{h}_{xg,neig}^{t-1}$.

11   Get global sequence embedding $\mathbf{SE}_{xg}^{t-1}$ with Eq. (6-7).

12   **Do Fuse Attentive Gating:**

13      $\overline{\mathbf{h}_{xl,pre}^{t-1}}$ = GATE $(\mathbf{h}_{xl,pre}^{t-1}, \mathbf{h}_{xg,pre}^{t-1}, \mathbf{h}_{xg,neig}^{t-1}, \mathbf{SE}_{xg}^{t-1}, \mathbf{SE}_{xl}^{t-1})$

14   **On local graph $\mathcal{G}_X^t$ do:**

15   Update V_{pre}'s embeddings to new state $\overline{\mathbf{h}_{xl,pre}^{t-1}}$

16   Add a directed edge from v_{i-1}^X to v_i.

17   Update nodes on graph $\mathcal{G}_X^t$ with Eq. (1-5).

18   **On global graph $\mathcal{G}_M^t$ do:**

19   Add a directed edge from v_{i-1}^M to v_i.

20   Update nodes on graph $\mathcal{G}_M^t$ with Eq. (1-5).

21 **end for**

22 Get sequence embeddings $\mathbf{SE}_{al,S_A}$, $\mathbf{SE}_{bl,S_B}$ with Eq. (6-7).

23 **return** $\mathbf{H}_{al}$, $\mathbf{H}_{bl}$; $\mathbf{SE}_{al,S_A}$, $\mathbf{SE}_{bl,S_B}$

After K steps of updating, we can obtain the embeddings of all nodes in $\mathcal{G}_A^t$ as $\mathbf{H}_{al}^t = \{\mathbf{h}_{al,1}^t, \mathbf{h}_{al,2}^t, ..., \mathbf{h}_{al,i}^t\}$. Then we adopt a strategy which attentively combines long-term and short-term preferences into the final sequential representation. As for long-term preference, we apply the soft-attention mechanism to measure

Table 1: Experimental results on Amazon datasets when the cut-off of the ranked list is 10.

	Movie-domain			Book-domain			Food-domain			Kitchen-domain		
	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR	HR	NDCG	MRR
POP	.0125	.0056	.0045	.0115	.0038	.0031	.0097	.0049	.0033	.0128	.0089	.0050
BPR-MF	.0513	.0418	.0316	.0457	.0359	.0266	.0345	.0278	.0214	.0382	.0319	.0435
Item-KNN	.0752	.0527	.0433	.0604	.0468	.0395	.0614	.0412	.0341	.0628	.0497	.0443
GRU4REC	.2320	.2014	.1785	.2068	.1651	.1510	.1910	.1587	.1482	.2034	.1745	.1591
SR-GNN	.2468	.2159	.1923	.2226	.1738	.1599	.2048	.1834	.1569	.2298	.2014	.1856
BERT4Rec	.2510	.2248	.2034	.2245	.1825	.1673	.2115	.1932	.1683	.2385	.2084	.1910
CL4SRec	.2656	.2350	.2147	.2352	.1964	.1740	.2235	.2041	.1775	.2487	.2193	.2044
NCF-MLP++	.1142	.0718	.0614	.1015	.0578	. 0501	.0946	.0622	.0498	.1057	.0754	.0602
Conet	.1287	.0788	.0725	.1182	.0624	.0550	.1043	.0795	.0592	.1138	.0940	.0775
DDTCDR	.1433	.0984	.0936	.1327	.0893	.0712	.1182	.0980	.0736	.1299	.1087	.0923
DARec	.1635	.1223	.1099	.1608	.1097	.0922	.1278	.1074	.0906	.1396	.1221	.0981
DAT-MDI	.2526	.2235	.2097	.2247	.1848	.1698	.2237	.2054	.1728	.2458	.2163	.1989
π-Net	.2744	.2359	.2218	.2402	.2075	.1903	.2405	.2128	.1886	.2616	.2247	.2177
PSJNet	.2862	.2429	.2347	.2516	.2161	.2035	.2512	.2233	.1964	.2682	.2412	.2326
DASL	.2940	.2587	.2412	.2586	.2248	.2159	.2620	.2318	.2174	.2812	.2540	.2493
DA-GCN	.2925	.2624	.2477	.2543	.2294	.2198	.2589	.2294	.2138	.2856	.2617	.2533
DDGHM-L	.2786	.2425	.2297	.2468	.2183	.2099	.2415	.2187	.1934	.2655	.2410	.2330
DDGHM-G	.2495	.2133	.1862	.2144	.1701	.1547	.2275	.2089	.1749	.2488	.2208	.2062
DDGHM-GA	.2957	.2688	.2534	.2590	.2316	.2239	.2658	.2378	.2204	.2781	.2599	.2423
DDGHM	**.3148**	**.2745**	**.2630**	**.2754**	**.2418**	**.2343**	**.2762**	**.2515**	**.2284**	**.2931**	**.2689**	**.2642**

the varying importance of previous items and then aggregate them as a whole:

$$\alpha_{al,k}^t = p_{al,t}^\top \sigma(\mathbf{W}_{a1}^t \mathbf{h}_{al,i}^t + \mathbf{W}_{a2}^t \mathbf{h}_{al,k}^t + \mathbf{c}^t),$$

$$\mathbf{SE}_{ac}^t = \sum_{k=1}^{i} \alpha_{a,k}^t \mathbf{h}_{al,k}^t, \qquad (6)$$

where parameters $p_{a,t}^\top \in \mathbb{R}^D$ and $\mathbf{W}_{a1}^t, \mathbf{W}_{a2}^t \in \mathbb{R}^{D \times D}$ control the weights of item embedding vectors. As for *short-term preference*, we concatenate the last item embedding which represents the current interest of the user with the above sequence embedding after aggregation and then take a linear transformation over them to generate the final sequence embedding:

$$\mathbf{SE}_{al}^t = \mathbf{W}_{a3}^t [\mathbf{SE}_{ac}^t; \mathbf{h}_{al,i}^t]. \qquad (7)$$

B DUAL DYNAMIC GRAPH MODELING

We represent the thorough algorithm of the dual dynamic graph modeling module in Algorithm 1. The input of this module includes single-domain and cross-domain behavior sequences, and the output consists of item embeddings and sequence embeddings. The whole procedure of dual dynamic graph modeling can be divided into two parts, i.e., 1) dual dynamic graphs and 2) fuse attentive gate. We describe how to construct dual dynamic graphs in line 14-20, and introduce how to transfer cross-domain information by fuse attentive gating mechanism in line 5-13.

C EXPERIMENTAL RESULTS

Here, we additionally report the experimental results on two Amazon datasets when the cut-off of the ranked list is 10. The results

show that: (1) **DDGHM** outperforms all the baselines of SR, CDR, and CDSR. (2) **DDGHM** also shows the superiority over its variants, i.e., **DDGHM**-L, **DDGHM**-G, and **DDGHM**-GA, indicating the effectiveness of each component in the dual dynamic graph modeling module.