

# DA-GCN: A Domain-aware Attentive Graph Convolution Network for Shared-account Cross-domain Sequential Recommendation

Lei Guo<sup>1</sup>, Li Tang<sup>1</sup>, Tong Chen<sup>2</sup>, Lei Zhu<sup>1</sup>, Quoc Viet Hung Nguyen<sup>3</sup>, Hongzhi Yin<sup>2\*</sup>

<sup>1</sup>Shandong Normal University, China

<sup>2</sup>The University of Queensland, Australia

<sup>3</sup>Griffith University, Australia

leiguoc.cs@gmail.com, litang96@126.com, tong.chen@uq.edu.au, {leizhu0608, quocviethung1}@gmail.com, h.yin1@uq.edu.au

## Abstract

Shared-account Cross-domain Sequential Recommendation (SCSR) is the task of recommending the next item based on a sequence of recorded user behaviors, where multiple users share a single account, and their behaviours are available in multiple domains. Existing work on solving SCSR mainly relies on mining sequential patterns via RNN-based models, which are not expressive enough to capture the relationships among multiple entities. Moreover, all existing algorithms try to bridge two domains via knowledge transfer in the latent space, and the explicit cross-domain graph structure is unexploited. In this work, we propose a novel graph-based solution, namely DA-GCN, to address the above challenges. Specifically, we first link users and items in each domain as a graph. Then, we devise a domain-aware graph convolution network to learn user-specific node representations. To fully account for users' domain-specific preferences on items, two novel attention mechanisms are further developed to selectively guide the message passing process. Extensive experiments on two real-world datasets are conducted to demonstrate the superiority of our DA-GCN method.

## 1 Introduction

Cross-domain Sequential Recommendation (CSR) is a recommendation task that aims at recommending the next item via leveraging a user's historical interactions from multiple domains. CSR is gaining immense research attention nowadays as users need to sign up for different platforms to access domain-specific services, e.g., music subscription and food delivery. In this work, we study CSR in an emerging yet challenging scenario, SCSR, where multiple individual users share a single account and their interaction behaviors are recorded in multiple domains [Lu *et al.*, 2019]. We consider the shared-account scenario because it is increasingly common that people share accounts with others in many applications. For example, members of a family tend to share an account for watching movies (e.g., Netflix) and

online shopping (e.g., Amazon). Consequently, generating accurate recommendations is much more challenging in this case because of the mixture of diverse users' interests within an interaction sequence [Yin *et al.*, 2020; Yin *et al.*, 2019; Guo *et al.*, 2020]. Furthermore, though in a single-user scenario, data from two domains might collaboratively help uncover her/his preferences, the existence of shared accounts will instead amplify the noise in the interaction data and impede the sequential recommendation accuracy.

Recently, several studies have been focused on recommendation with either shared-account or cross-domain settings, very few of them address SCSR that simultaneously considers both aspects. In prior work on shared accounts [Bajaj and Shekhar, 2016], a common approach is to capture the user relationships under the same account with latent representations, but none of them consider the cross-domain context, and are hence inapplicable to SCSR. Despite that cross-domain recommenders [Zhuang *et al.*, 2018], partially fit this scenario, the cross-domain knowledge is implicitly transferred in the latent space, the explicit structural information bridging two domains are largely unexplored. Zhao *et al.* [2019] proposed a graph-based model as a solution, but their work ignores the important sequential information and relies on explicit user ratings that are not usually available in both domains. However, those cross-platform methods are not designed to cope with entangled user preferences on shared accounts. One prior work on SCSR is the  $\pi$ -net method [Ma *et al.*, 2019], which formulates SCSR as a parallel sequential recommendation problem that is solved by an information-sharing network. Another related work is the PSJNet method [Ren *et al.*, 2019], which improves the  $\pi$ -net using a split-join strategy. However, these RNN-based methods overwhelmingly target on discovering sequential dependencies, and have limited capability for capturing the complex relationships among associated entities (i.e., users and items) in both domains. As a result, this limits the expressiveness of learned user and item representations, and also overlooks the explicit structural information (e.g., item-user-item paths) linking two domains.

To address the above challenges, we propose a novel graph-based solution, namely Domain-Aware Graph Convolutional Network (DA-GCN), for SCSR. Specifically, to model the complicated interaction relationships, we first construct a

\*Corresponding author

Cross-Domain Sequence (CDS) graph to link different domains, where users and items in each domain are nodes and their associations are edges. Thereafter, to adapt to the scattered user preferences on shared accounts, we suppose there are  $H$  latent users under each account and leverage the message-passing strategy in a domain-aware graph convolution network to aggregate the information passed from directly linked neighbors. Then, node embeddings of users and items are learned by passing information from connected neighbor nodes. Though interactions with items are recorded at the account level, items in different domains will have different attraction to different users under the same account. In a similar vein, different neighbor users/items may have varied importance when learning the properties of an item. Hence, we further design two specialized attention mechanisms to discriminatively select relevant information during the message passing. Consequently, we are capable of modeling the multifaceted interactions as well as transferring fine-grained domain knowledge by considering the structure information.

The main contributions of this work are summarized as:

- We investigate an emerging yet challenging recommendation task, namely SCSR. After pointing out the defects of existing RNN-based solutions, we bring a new take on the SCSR problem with a graph-based approach.
- We firstly construct an innovative CDS graph that explicitly links accounts and items from two domains, and build a novel graph model named DA-GCN to attentively learn expressive representations for items and account-sharing users for recommendation.
- We conduct extensive experiments on two real-word datasets, and the experimental results demonstrate the superiority of DA-GCN compared with several state-of-the-art baselines.

## 2 Related Work

### 2.1 Cross-domain Recommendation

As Cross-domain Recommendation (CR) concerns data from multiple domains, it has been proven useful in dealing with cold-start [Abel *et al.*, 2013] and data sparsity issues [Pan *et al.*, 2010]. Existing studies can be categorized into traditional methods and deep learning-based methods. In traditional methods, there are two main ways in dealing with CR. One is to aggregate knowledge between two domains [Hu *et al.*, 2018]. Another is to transfer knowledge from the source domain to the target domain [Tang *et al.*, 2012]. Deep learning-based methods are well suited to transfer learning, as they can learn high-level abstractions among different domains [Zhao *et al.*, 2019; Liu *et al.*, 2020]. For example, Zhao *et al.* [2019] and Liu *et al.* [2020] studied CR by leveraging the Graph Convolution Network (GCN)-based techniques, but their methods all rely on explicit user ratings, and can not be directly applied to SCSR.  $\pi$ -net [Ma *et al.*, 2019] and PSJNet [Ren *et al.*, 2019] are two recently proposed methods for SCSR, but their methods are all based on RNNs, which are neither expressive enough to capture the multiple associations nor can model the structure information that bridges two domains.

### 2.2 Shared-account Recommendation

The task of recommending items for a shared-account is to make recommendations by modeling the mixture of user behaviours, which usually performs user identification first, and then makes recommendations [Zhao *et al.*, 2016; Jiang *et al.*, 2018]. For example, Wang *et al.* [2014] are the first to study user identification in a novel perspective, in which they employ user preference and consumption time to indicate user’s identity. Jiang *et al.* [2018] introduced an unsupervised framework to judge users within a same account and then learned the preferences for each of them. In their method, a heterogeneous graph is exploited to learn the relationship between items and the metadata. Ma *et al.* [2019] and Ren *et al.* [2019] argued that this task can be treated in an end-to-end fashion, and can also be improved by simultaneously considering the cross-domain information.

### 2.3 GCN-based Recommendation

The development of the graph neural networks [Wang *et al.*, 2019b; Wang *et al.*, 2019c] has attracted a lot of attention to exploring graph-based solutions for recommender systems [Guo *et al.*, 2021; Wang *et al.*, 2021]. E.g., PinSage [Ying *et al.*, 2018] combines random walks with multiple graph convolutional layers on the item-item graph for Pinterest image recommendation. NGCF [Wang *et al.*, 2019b] exploits high-order proximity by propagating embeddings on the user-item interaction graph to simultaneously update the representations for all users and items in an efficient way by implementing the matrix-form rule. Qiu *et al.* [2020] exploits the session-based recommendation problem via proposing a novel full graph neural network to the item dependencies between different sessions. However, none of them can be directly applied to SCSR as they either fail to capture the important sequential information [Wang *et al.*, 2020] or focus on the recommendation problem in a single domain.

## 3 Methodologies

### 3.1 Preliminaries

Let  $S_A = \{A_1, A_2, \dots, A_i\}$  and  $S_B = \{B_1, B_2, \dots, B_j\}$  be the behaviour sequences from domain A and B respectively, where  $A_i \in \mathcal{A} (1 \leq i \leq p)$  is the item index in domain A;  $\mathcal{A}$  is the item set in domain A;  $B_j \in \mathcal{B} (1 \leq j \leq q)$  is the item index in domain B; and  $\mathcal{B}$  is the item set in domain B.  $U = \{U_1, U_2, \dots, U_k, \dots, U_{|U|}\}$  is a subset of accounts, where  $U_k \in \mathcal{U} (1 \leq k \leq n)$  denotes the account index and  $\mathcal{U}$  represents the whole account set. Given  $S_A$  and  $S_B$ , the task of SCSR is to recommend the next item based on a user’s historical behaviours. The recommendation probabilities for all candidate items in domain A and B are:

$$P(A_{i+1}|S_A, S_B) \sim f_A(S_A, S_B); \tag{1}$$

$$P(B_{j+1}|S_B, S_A) \sim f_B(S_B, S_A), \tag{2}$$

where  $P(A_{i+1}|S_A, S_B)$  is the probability of recommending the next item  $A_{i+1}$  in domain A, given  $S_A$  and  $S_B$ .  $f_A(S_A, S_B)$  is the learned function to estimate  $P(A_{i+1}|S_A, S_B)$ . Similar definitions are applied to  $P(B_{j+1}|S_B, S_A)$  and  $f_B(S_B, S_A)$ .

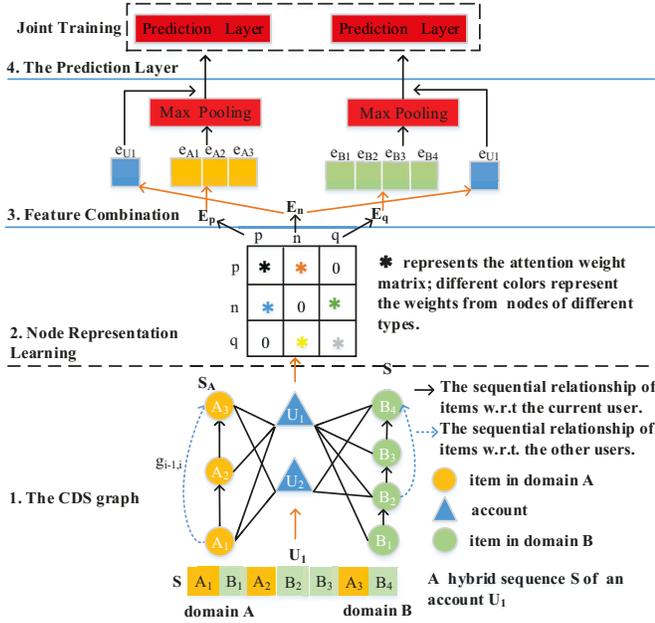


Figure 1: An overview of DA-GCN. The directed edge  $g_{i-1,i}$  denotes the click of item  $A_i$  after item  $A_{i-1}$ .

### 3.2 An Overview of DA-GCN

DA-GCN tries to learn a graph-based recommendation framework that can model the multiple associations and the structure-aware domain knowledge via recursively aggregating features from local graph neighbors in different domains, so as to better capture the diverse user preferences in a shared account. Fig. 1 presents an overview of DA-GCN. Specifically, to model the sophisticated associations among users and items, a Cross-Domain Sequence (CDS) graph is first constructed. Four types of associations are considered: a) user-item interactions in domain A; b) user-item interactions in domain B; c) sequential transitions among items in domain A; d) sequential transitions among items in domain B. Then, we subsume  $H$  latent users under the same account and leverage GCN to learn node representation for each latent user, which collects information from the connected neighbors in different domains. In our case, as multiple users interact with different items with a joint account, an interacted item is actually of interest to only some specific account users. Hence, a domain-aware attention mechanism is further proposed to weight the passed messages from different domains. Analogously, we also design another attention scheme on the item side during information aggregation. Thereafter, we generate the sequence-level embedding of each domain via concatenation and pooling operations, which facilitates prediction of the next item. Finally, to mutually maximize the knowledge, a joint training paradigm is leveraged.

### 3.3 The Proposed Method

Motivated by Wang et al. [2019a] and Zhao et al. [2019], in this work we employ the message-passing strategy to learn the node representations from the CDS graph.

### Representation Learning with Latent Users

Since in SCSR, an account is usually shared by multiple users, behaviors under the same account are usually generated by different users. Moreover, as users under the same account often have different interests, uniformly treating an account as a virtual user is inappropriate. Because the number and identity of all users under an account are both unknown, we assume there are  $H$  latent users ( $U_{k,1}, U_{k,2}, \dots, U_{k,h}, \dots, U_{k,H}$ ) under each account ( $U_k$ ), and the embedding of user  $U_{k,h}$  is denoted by  $e_{U_{k,h}} \in \mathbb{R}^d$ , which is learned by accumulating all the information passed from connected items in domain A and B. Intuitively, by defining latent users under each account, we can learn diversified user representations from their cross-domain interaction sequences to encode multifaceted personalities and preferences.

**Message Passing.** Suppose  $A_i \in \mathcal{N}_{A_i}^{U_{k,h}}$  and  $B_j \in \mathcal{N}_{B_j}^{U_{k,h}}$  are the item neighbors of  $U_{k,h}$  in domain A and B respectively, where  $\mathcal{N}_{A_i}^{U_{k,h}}$  is the item neighbor set of  $U_{k,h}$  in domain A, and  $\mathcal{N}_{B_j}^{U_{k,h}}$  is the item neighbor set in domain B. Specifically, we define the message passed from  $A_i \in \mathcal{N}_{A_i}^{U_{k,h}}$  to  $U_{k,h}$  as:

$$\mathbf{m}_{U_{k,h} \leftarrow A_i} = \gamma_{A_i}^{U_{k,h}} (\mathbf{W}_1 e_{A_i} + \mathbf{W}_2 (e_{A_i} \odot e_{U_{k,h}})), \quad (3)$$

where  $\mathbf{m}_{U_{k,h} \leftarrow A_i}$  is the message representation;  $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d' \times d}$  are the trainable weight matrices. The information interaction between  $A_i$  and  $U_{k,h}$  is represented by the element-wise product  $e_{A_i} \odot e_{U_{k,h}}$ , where  $e_{A_i} \in \mathbb{R}^d$  and  $e_{U_{k,h}}$  are the embedding vectors of item  $A_i$  and user  $U_{k,h}$  respectively.  $\gamma_{A_i}^{U_{k,h}}$  is a learnable parameter that controls how much information can be passed from item  $A_i$  to user  $U_{k,h}$ . We will detail the learning process of this attentive weight in the subsequent section. Similar to domain A, the message passed from  $B_j$  to  $U_{k,h}$  in domain B can be defined as:

$$\mathbf{m}_{U_{k,h} \leftarrow B_j} = \gamma_{B_j}^{U_{k,h}} (\mathbf{W}_1 e_{B_j} + \mathbf{W}_2 (e_{B_j} \odot e_{U_{k,h}})). \quad (4)$$

Besides, to retain the information carried by the target user, a self-connection is further added to her/him via:

$$\mathbf{m}_{U_{k,h} \leftarrow U_{k,h}} = \gamma_{U_{k,h}}^{U_{k,h}} (\mathbf{W}_1 e_{U_{k,h}}), \quad (5)$$

where  $\gamma_{B_j}^{U_{k,h}}$  and  $\gamma_{U_{k,h}}^{U_{k,h}}$  are attentive weights guiding the strength of each message to be passed, which are also computed by an attention mechanism.

**Domain-aware Attention Mechanism.** On both domains, to fully capture each latent user's distinct preferences over different interacted items, we devise a domain-aware attention mechanism to compute the importance of different item nodes to the target user. The importance of  $A_i$  to  $U_{k,h}$  in domain A is defined as:

$$s_{A_i}^{U_{k,h}} = f(e_{U_{k,h}}, e_{A_i}), \quad (6)$$

where  $f(\cdot)$  is a pairwise similarity metric. In this work, cosine similarity is applied. Similarly, the importance of  $B_j$  to  $U_{k,h}$  in domain B can be defined as:

$$s_{B_j}^{U_{k,h}} = f(e_{U_{k,h}}, e_{B_j}). \quad (7)$$

Then, the attentive weight on each item from domain A is obtained by the following normalization operation:

$$\begin{aligned} \gamma_{A_i}^{U_{k,h}} &= \exp(s_{A_i}^{U_{k,h}}) / \left( \sum_{A_i' \in \mathcal{N}_{A_i}^{U_{k,h}}} \exp(s_{A_i'}^{U_{k,h}}) \right. \\ &\quad \left. + \sum_{B_j \in \mathcal{N}_{B_j}^{U_{k,h}}} \exp(s_{B_j'}^{U_{k,h}}) + s_{U_{k,h}}^{U_{k,h}} \right), \end{aligned} \quad (8)$$

where  $s_{U_{k,h}}^{U_{k,h}}$  weighs the self-connection within user  $U_{k,h}$ . Analogously, by replacing the corresponding user/item embeddings,  $\gamma_{B_j}^{U_{k,h}}$  and  $\gamma_{U_{k,h}}^{U_{k,h}}$  can be obtained in a similar way.

**Message Aggregation.** The embedding of user  $U_{k,h}$  ( $e_{U_{k,h}}$ ) is then updated via:

$$\begin{aligned} e_{U_{k,h}} &= \text{LeakyReLU}(\mathbf{m}_{U_{k,h} \leftarrow U_{k,h}} \\ &\quad + \sum_{A_i \in \mathcal{N}_{A_i}^{U_{k,h}}} \mathbf{m}_{U_{k,h} \leftarrow A_i} + \sum_{B_j \in \mathcal{N}_{B_j}^{U_{k,h}}} \mathbf{m}_{U_{k,h} \leftarrow B_j}), \end{aligned} \quad (9)$$

where all the messages passed from her/him-self and items in both domains are aggregated, followed by a LeakyReLU activation function. Eventually, we merge all latent users' representations into the account-level representation  $U_k$ :

$$e_{U_k} = \frac{1}{H} \sum_{h=1}^H e_{U_{k,h}}. \quad (10)$$

### User-specific Item Representation Learning

For an item, its representation is learned by aggregating the information from two types of nodes, i.e., the connected users and items within the same domain. As the item representations are learned in a similar way in both domains, we take domain A as an example, and the process is also applicable for domain B.

**Message Passing.** Let  $A_{i-1} \in \mathcal{N}_{A_{i-1}}^{A_i}$  and  $U_{k,h} \in \mathcal{N}_{U_{k,h}}^{A_i}$  be the connected item and user nodes respectively, where  $\mathcal{N}_{A_{i-1}}^{A_i}$  is the item neighbor set of  $A_i$  in domain A and  $\mathcal{N}_{U_{k,h}}^{A_i}$  is the user neighbor set of  $A_i$ . Then, the message passed from item  $A_{i-1}$  to  $A_i$  is formulated as:

$$\mathbf{m}_{A_i \leftarrow A_{i-1}} = \gamma_{A_{i-1}}^{A_i} (\mathbf{W}_1 e_{A_{i-1}} + \mathbf{W}_2 (e_{A_{i-1}} \odot e_{A_i})), \quad (11)$$

where the parameter  $\gamma_{A_{i-1}}^{A_i}$  controls how much information can be passed from  $A_{i-1}$  to  $A_i$ , as we will explain shortly. The message passed from user  $U_{k,h}$  to  $A_i$  is defined as:

$$\mathbf{m}_{A_i \leftarrow U_{k,h}} = \gamma_{U_{k,h}}^{A_i} (\mathbf{W}_1 e_{U_{k,h}} + \mathbf{W}_2 (e_{U_{k,h}} \odot e_{A_i})). \quad (12)$$

Similarly, the retained message of  $A_i$  is formulated as:

$$\mathbf{m}_{A_i \leftarrow A_i} = \gamma_{A_i}^{A_i} (\mathbf{W}_1 e_{A_i}), \quad (13)$$

where  $\gamma_{U_{k,h}}^{A_i}$  and  $\gamma_{A_i}^{A_i}$  are both learnable weights.

**Sequence-aware Attention Mechanism.** Due to varied relevance of linked users/items to the target item, we develop another attention method to assess the importance of the connected users and the items that have sequential relationships to the target item. The importance of item  $A_{i-1}$  to  $A_i$  is formulated as:

$$s_{A_{i-1}}^{A_i} = f(e_{A_{i-1}}, e_{A_i}). \quad (14)$$

The importance of user  $U_{k,h}$  on item  $A_i$  is defined as:

$$s_{U_{k,h}}^{A_i} = f(e_{A_i}, e_{U_{k,h}}). \quad (15)$$

Then, we normalize  $s_{A_{i-1}}^{A_i}$  by the following operation:

$$\begin{aligned} \gamma_{A_{i-1}}^{A_i} &= \exp(s_{A_{i-1}}^{A_i}) / \left( \sum_{A_i' \in \mathcal{N}_{A_{i-1}}^{A_i}} \exp(s_{A_i'}^{A_i}) \right. \\ &\quad \left. + \sum_{U_{k,h}' \in \mathcal{N}_{U_{k,h}}^{A_i}} \exp(s_{U_{k,h}'}^{A_i}) + s_{A_i}^{A_i} \right), \end{aligned} \quad (16)$$

where  $s_{A_i}^{A_i}$  is the importance of the self-connection within item  $A_i$ .  $\gamma_{U_{k,h}}^{A_i}$  and  $\gamma_{A_i}^{A_i}$  are obtained in the same way.

**Message Aggregation.** The user-specific item representation is updated by aggregating all messages passed from the neighbor user/item nodes within domain A:

$$\begin{aligned} e_{A_i} &= \text{LeakyReLU}(\mathbf{m}_{A_i \leftarrow A_i} + \sum_{A_{i-1} \in \mathcal{N}_{A_{i-1}}^{A_i}} \mathbf{m}_{A_i \leftarrow A_{i-1}} \\ &\quad + \sum_{U_{k,h} \in \mathcal{N}_{U_{k,h}}^{A_i}} \mathbf{m}_{A_i \leftarrow U_{k,h}}). \end{aligned} \quad (17)$$

For better expression, we use  $\mathbf{g}_{A_i}^h = e_{A_i}$  to denote the embedding of item  $A_i$  w.r.t the  $h$ -th user. Then, the final representation of item  $A_i$  ( $\mathbf{g}_{A_i}$ ) can be defined as:

$$\mathbf{g}_{A_i} = \frac{1}{H} \sum_{h=1}^H \mathbf{g}_{A_i}^h. \quad (18)$$

### Matrix-form Propagation Rule

To efficiently update the representations for all users and items, we formulate the propagation rule in a layer-wise matrix-form, which is defined as:

$$\mathbf{E}_l = \sigma((\mathcal{L} + \mathbf{I})\mathbf{E}_{l-1}\mathbf{W}_1 + \mathcal{L}\mathbf{E}_{l-1} \odot \mathbf{E}_{l-1}\mathbf{W}_2), \quad (19)$$

where  $\mathbf{I}$  denotes an identity matrix; and  $\mathbf{E}_l \in \mathbb{R}^{(p+n+q) \times d}$  is the representation of all user and item nodes in domain A and B.  $\mathcal{L} \in \mathbb{R}^{(p+n+q) \times (p+n+q)}$  is the Laplacian matrix for the CDS graph, which is defined as:

$$\mathcal{L} = \begin{bmatrix} \mathbf{Y}_{A_i A_{i-1}} & \mathbf{Y}_{A_i U_k} & \mathbf{0} \\ \mathbf{Y}_{U_k A_i} & \mathbf{0} & \mathbf{Y}_{U_k B_j} \\ \mathbf{0} & \mathbf{Y}_{B_j U_k} & \mathbf{Y}_{B_j B_{j-1}} \end{bmatrix}, \quad (20)$$

where  $\mathbf{Y}_{A_i A_{i-1}} \in \mathbb{R}^{p \times p}$  and  $\mathbf{Y}_{A_i U_k} \in \mathbb{R}^{p \times n}$  are the attention matrices carrying the weights from item and user neighbors to the target item in domain A, respectively;  $\mathbf{Y}_{U_k A_i} \in \mathbb{R}^{n \times p}$  represents the weights from item neighbors in domain A to user nodes;  $\mathbf{Y}_{U_k B_j} \in \mathbb{R}^{n \times q}$  denotes the weights from item neighbors in domain B to the user nodes;  $\mathbf{Y}_{B_j U_k} \in \mathbb{R}^{q \times n}$  and  $\mathbf{Y}_{B_j B_{j-1}} \in \mathbb{R}^{q \times q}$  represent the weights from user nodes and item nodes to the target item in domain B, respectively.

Methods	HVIDEO								HAMAZON							
	E-domain (%)				V-domain (%)				M-domain (%)				B-domain (%)			
	MRR		Recall		MRR		Recall		MRR		Recall		MRR		Recall	
	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20
POP	1.71	2.24	2.21	6.58	2.66	3.27	5.01	10.49	0.36	0.49	0.73	2.02	0.14	0.22	0.42	1.22
Item-KNN	2.11	2.90	3.01	12.11	4.43	2.93	10.48	23.93	1.28	1.86	2.58	9.00	3.23	4.55	6.65	20.94
BPR-MF	1.34	1.64	2.74	5.83	1.21	1.36	1.88	3.38	2.90	3.06	3.90	5.50	0.88	0.96	1.23	2.15
VUI-KNN	2.03	3.48	6.36	24.27	3.44	2.87	16.46	34.76	-	-	-	-	-	-	-	-
NCF-MLP++	3.92	5.14	7.36	20.81	16.25	17.90	26.10	43.04	13.68	14.21	18.44	24.31	13.67	14.05	18.14	22.08
Conet	5.01	6.21	9.26	22.71	21.25	23.28	32.94	52.72	14.64	15.12	19.25	24.46	15.85	16.28	20.98	25.56
GRU4REC	12.27	13.70	16.24	32.16	78.27	78.27	80.15	83.04	82.01	82.11	83.10	84.06	81.34	81.44	82.77	83.76
HGRU4REC	14.47	16.11	19.79	37.52	80.37	80.62	81.92	84.43	83.07	83.14	84.24	84.91	82.15	82.31	83.46	84.91
$\pi$ -net	14.63	16.88	20.41	45.19	80.51	80.95	83.22	87.48	83.91	83.95	84.91	85.33	84.93	84.93	85.33	85.38
PSJNet	16.63	18.46	22.12	42.20	81.97	82.32	84.32	87.75	84.01	84.05	84.88	85.28	<b>85.10</b>	<b>85.11</b>	85.32	85.38
DA-GCN	<b>19.24</b>	<b>21.24</b>	<b>26.65</b>	<b>47.78</b>	<b>83.13</b>	<b>83.42</b>	<b>85.46</b>	<b>88.30</b>	<b>84.69</b>	<b>84.71</b>	<b>85.13</b>	<b>85.34</b>	84.81	84.81	<b>85.32</b>	<b>85.38</b>

Table 1: Experimental results on HVIDEO and HAMAZON. VUI-KNN does not work on this dataset because it needs specific time in a day which is not available on HAMAZON dataset.

### The Prediction Layer

Through the max pooling operation on the item embeddings within the sequence  $(g_{A_1}, g_{A_2}, \dots, g_{A_i})$  (or  $(g_{B_1}, g_{B_2}, \dots, g_{B_j})$ ), we can get the sequence-level embedding  $h'_{S_A}$  (or  $h'_{S_B}$ ) for  $S_A$  (or  $S_B$ ). Then, we feed the concatenation of  $h'_{S_A}$  (or  $h'_{S_B}$ ) and  $e_{U_k}$  into the following prediction layer:

$$P(A_{i+1}|S_A, S_B) = softmax(W_A \cdot [h'_{S_A}, e_{U_k}]^T + b_A);$$

$$P(B_{j+1}|S_A, S_B) = softmax(W_B \cdot [h'_{S_B}, e_{U_k}]^T + b_B),$$

where  $W_A, W_B$  are the embedding matrices of all items in domain A and B, respectively;  $b_A, b_B$  are the bias terms. Then, we leverage a negative log-likelihood loss function to train DA-GCN in each domain:

$$L_A(\theta) = -\frac{1}{|\mathbb{S}|} \sum_{S_A, S_B \in \mathbb{S}} \sum_{A_i \in S_A} \log P(A_{i+1}|S_A, S_B), \quad (21)$$

$$L_B(\theta) = -\frac{1}{|\mathbb{S}|} \sum_{S_B, S_A \in \mathbb{S}} \sum_{B_j \in S_B} \log P(B_{j+1}|S_A, S_B), \quad (22)$$

where  $\theta$  represents all the parameters of DA-GCN and  $\mathbb{S}$  denotes the training sequences in both domain A and B. Both objectives are joined via a multi-task training scheme:

$$L(\theta) = L_A(\theta) + L_B(\theta). \quad (23)$$

All the parameters in DA-GCN are learned via gradient back-propagation algorithm in an end-to-end fashion.

## 4 Experiments

### 4.1 Experimental Setup

**Research Questions.** We intend to answer the following research questions:

**RQ1.** How does our proposed DA-GCN method perform compared with other state-of-the-art methods?

**RQ2.** Is it helpful to leverage the sequence information? Is it helpful to learn the node representations by incorporating the attention mechanism?

**RQ3.** Is it helpful to model the shared-account characteristic? How does the hyper-parameter  $H$  (the number of users) affect the performance of DA-GCN?

**RQ4.** How is the training efficiency and scalability of DA-GCN when processing large-scale data?

**Datasets and Evaluation Protocols.** We evaluate DA-GCN on two real-world datasets that are released by Ren et al. [2019], i.e., HVIDEO and HAMAZON. HVIDEO is a smart TV dataset that contains the watching logs from two platforms, i.e., the V-domain (the videos of TV series, movies, etc.) and E-domain (the educational videos based on textbooks and instructional videos on sports food, medical, etc.). HAMAZON is the data from two Amazon domains, that is, the movie watching and rating records (M-domain) and the book reading and rating behaviors (B-domain). We randomly select 75% of all the sequences as the training data, 15% as the validation set and the remaining 10% as the test set. For evaluation, we use the last watched item in each sequence for each domain as the ground truth and report the results measured by the commonly used metrics MRR@5, Recall@5, MRR@20 and Recall@20.

**Baselines.** We compare DA-GCN with the following baseline methods: 1) Traditional recommendations: POP [He et al., 2017], Item-KNN [Linden et al., 2003], and BPR-MF [Hidasi et al., 2016]. 2) Shared-account recommendations: VUI-KNN [Wang et al., 2014]. 3) Cross-domain recommendations: NCF-MLP++ [He et al., 2017], and Conet [Hu et al., 2018]. 4) Sequential recommendations: GRU4REC [Hidasi et al., 2016], HGRU4REC [Quadana et

Variants	E-domain (%)				V-domain (%)			
	MRR		Recall		MRR		Recall	
	@5	@20	@5	@20	@5	@20	@5	@20
GCN <sub>OSA</sub>	19.24	20.86	26.24	43.40	82.99	83.23	85.23	87.66
GCN <sub>OA</sub>	19.11	20.81	26.07	43.78	83.09	83.35	85.43	87.88
GCN <sub>OS</sub>	19.21	21.19	26.37	47.04	83.12	83.42	85.45	88.27
DA-GCN	<b>19.24</b>	<b>21.24</b>	<b>26.65</b>	<b>47.78</b>	<b>83.13</b>	<b>83.42</b>	<b>85.46</b>	<b>88.30</b>

Table 2: Ablations studies on the HVIDEO dataset.

al., 2017]. 5) Shared-account cross-domain sequential recommendations:  $\pi$ -net [Ma et al., 2019], and PSJNet [Ren et al., 2019].

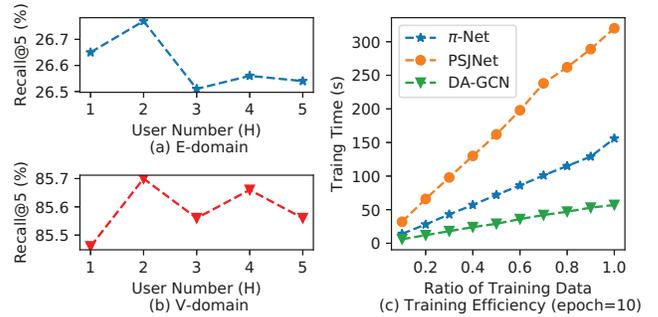
**Implementation Details.** We implement DA-GCN using the Tensorflow framework accelerated by NVidia RTX 2080 Ti GPU. For the initialization strategy, we randomly initialize the model parameters by the Xavier method [Glorot and Bengio, 2010]. We take Adam as our optimizing algorithm and also apply gradient clipping with range [-5,5] during the training period. For hyperparameters, the learning rate is set to 0.001. The embedding size is set to 100 on HVIDEO and 200 on HAMAZON. To speed up the training process, the batch size are both set to 128. The latent user number  $H$  is searched in [1-5] with a step size of 1. All these parameters are tuned on the validation set.

## 4.2 Experimental Results (RQ1)

Table 1 shows the comparison results of DA-GCN over other baselines on HVIDEO and HAMAZON. We have the following observations: 1) Our DA-GCN method achieves the best performance on both domains of HVIDEO and outperforms other baselines on HAMAZON in most metrics, which demonstrates the capability of DA-GCN in modeling the multiple associations and the user-specific representations, as well as the structure information of the domain knowledge. 2) The methods developed for SCSR are significantly better than other baselines, demonstrating the importance of considering the shared-account and cross-domain characteristics simultaneously. 3) DA-GCN outperforms the shared account and cross-domain baselines (i.e., VUI-KNN, NCF-MLP++ and Conet) indicating the importance of the sequential information and the effectiveness of our method in modeling users' sequential patterns. 4) From Table 1, we also observe that our GCN-based solution performs better than RNN-based methods ( $\pi$ -net and PSJNet). This result demonstrates the capability of our GCN-based method in modeling the sequential information and the complicated interaction relationships.

## 4.3 Ablation Studies of DA-GCN (RQ2)

To further illustrate the impact of different components to DA-GCN, we conduct ablation studies on both datasets. Due to space limitation, only results on HVIDEO are presented and similar results are obtained on HAMAZON. The experimental results are shown in Table 2, where GCN<sub>OS</sub> is the method that disables the sequential information when constructing the CDS graph. GCN<sub>OA</sub> is the method that disables


 Figure 2: Impact of  $H$  and Model Training Efficiency on HVIDEO.

the attention mechanisms when aggregating the passed messages. GCN<sub>OSA</sub> is a variant of DA-GCN that removes both components. From Table 2 we can find that: 1) DA-GCN outperforms GCN<sub>OSA</sub> and GCN<sub>OS</sub> demonstrating the importance of leveraging the sequential information in modeling users' preferences. 2) DA-GCN outperforms GCN<sub>OA</sub> indicating the importance of weighting the passed messages differently and our model can learn a better node representation via the proposed attention mechanisms.

## 4.4 Impact of $H$ and Model Training Efficiency

**Impact of  $H$  (RQ3).** To explore the importance of modeling the shared-account characteristic, we further conduct experiments on HVIDEO to show the impact of  $H$ . The experimental results shown in Fig. 2 demonstrate that viewing an account as a virtual user can not get better results than modeling it as  $H$  latent users, which is more in line with reality.

**Training Efficiency (RQ4).** To investigate the model training efficiency and scalability of our graph-based solution, we further conduct experiments via measuring the time cost for the model training with varying the training ratio of the HVIDEO data in [0.1, 1.0] with step size 0.1. From the experimental results shown in Fig. 2, we can find that DA-GCN has lower training time cost than  $\pi$ -net and PSJNet, demonstrating its better scalability to large scale datasets.

## 4.5 Conclusions

In this work, we propose DA-GCN for SCSR to fully learn user and item representations from their interactions, as well as the explicit structural information. Specifically, to model the multiple associations among users and items, we first link them in a CDS graph. To model the structure information of the transferred knowledge, we then develop a domain-ware GCN to learn user-specific node representations, where two attention mechanisms are devised to weight the local neighbors of the target. The experimental results on two real-world datasets demonstrate the superiority of our graph-based solution.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (Nos. 61602282, 61802236), ARC Discovery Project (No. DP190101985) and ARC Training Centre for Information Resilience (No. IC200100022).

## References

- [Abel *et al.*, 2013] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the social web. In *UMUAI*, 2013.
- [Bajaj and Shekhar, 2016] Payal Bajaj and Sumit Shekhar. Experience individualization on online tv platforms through persona-based account decomposition. In *MM*, 2016.
- [Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In *AISTATS*, 2010.
- [Guo *et al.*, 2020] Lei Guo, Hongzhi Yin, Qinyong Wang, Bin Cui, Zi Huang, and Lizhen Cui. Group recommendation with latent voting mechanism. In *ICDE*, 2020.
- [Guo *et al.*, 2021] Lei Guo, Hongzhi Yin, Tong Chen, Xiangliang Zhang, and Kai Zheng. Hierarchical hyperedge embedding-based representation learning for group recommendation. *TOIS*, 2021.
- [He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, 2017.
- [Hidasi *et al.*, 2016] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *ICLR*, 2016.
- [Hu *et al.*, 2018] Guangneng Hu, Yu Zhang, and Qiang Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *CIKM*, 2018.
- [Jiang *et al.*, 2018] Jyun-Yu Jiang, Cheng-Te Li, Yian Chen, and Wei Wang. Identifying users behind shared accounts in online streaming services. In *SIGIR*, 2018.
- [Linden *et al.*, 2003] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003.
- [Liu *et al.*, 2020] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In *CIKM*, 2020.
- [Lu *et al.*, 2019] Xu Lu, Lei Zhu, Zhiyong Cheng, Liqiang Nie, and Huaxiang Zhang. Online multi-modal hashing with dynamic query-adaption. In *SIGIR*, 2019.
- [Ma *et al.*, 2019] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke.  $\pi$ -net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *SIGIR*, 2019.
- [Pan *et al.*, 2010] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, 2010.
- [Qiu *et al.*, 2020] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. Exploiting cross-session information for session-based recommendation with graph neural networks. *TOIS*, 2020.
- [Quadrana *et al.*, 2017] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys*, 2017.
- [Ren *et al.*, 2019] Pengjie Ren, Yujie Lin, Muyang Ma, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Parallel split-join networks for shared-account cross-domain sequential recommendation. *TOIS*, 2019.
- [Tang *et al.*, 2012] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *KDD*, 2012.
- [Wang *et al.*, 2014] Zhijin Wang, Yan Yang, Liang He, and Junzhong Gu. User identification within a shared account: Improving ip-tv recommender performance. In *ADBIS*, 2014.
- [Wang *et al.*, 2019a] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *KDD*, 2019.
- [Wang *et al.*, 2019b] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, 2019.
- [Wang *et al.*, 2019c] Yuandong Wang, Hongzhi Yin, Hongxu Chen, Tianyu Wo, Jie Xu, and Kai Zheng. Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling. In *KDD*, 2019.
- [Wang *et al.*, 2020] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. Intention nets: psychology-inspired user choice behavior modeling for next-basket prediction. In *AAAI*, 2020.
- [Wang *et al.*, 2021] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z Sheng, Mehmet Orgun, Longbing Cao, Francesco Ricci, and Philip S Yu. Graph learning based recommender systems: A review. 2021.
- [Yin *et al.*, 2019] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. Social influence-based group representation learning for group recommendation. In *ICDE*, 2019.
- [Yin *et al.*, 2020] H. Yin, Q. Wang, K. Zheng, Z. Li, and X. Zhou. Overcoming data sparsity in group recommendation. *TKDE*, 2020.
- [Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, and Pong Eksombatchai. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, 2018.
- [Zhao *et al.*, 2016] Yafeng Zhao, Jian Cao, and Yudong Tan. Passenger prediction in shared accounts for flight service recommendation. In *APSCC*, 2016.
- [Zhao *et al.*, 2019] Cheng Zhao, Chenliang Li, and Cong Fu. Cross-domain recommendation via preference propagation graphnet. In *CIKM*, 2019.
- [Zhuang *et al.*, 2018] Fuzhen Zhuang, Yingmin Zhou, Fuzheng Zhang, Xiang Ao, Xing Xie, and Qing He. Cross-domain novelty seeking trait mining for sequential recommendation. *arXiv preprint arXiv:1803.01542*, 2018.