A Hierarchical Attention Network for Cross-Domain Group Recommendation

Ruxia Liang, Qian Zhang, *Member, IEEE*, Jianqiang Wang, and Jie Lu, *Fellow, IEEE*

***Abstract*— Many online services allow users to participate in various group activities such as online meeting or group buying, and thus need to provide user groups with services that they are interested. The group recommender systems (GRSs) emerge as required and provide personalized services for various online user groups. Data sparsity is an important issue in GRSs, since even fewer group–item interactions are observed. Moreover, the group and the group members have complex and mutual relationships with each other, which exacerbates the difficulty in modeling the preferences of both a group and its members for recommendation. The cross-domain recommender system (CDRS) is a solution to alleviate data sparsity and assist preference modeling by transferring knowledge from a source domain which has relatively dense data to another. The existing CDRSs are usually developed for individual users and cannot be directly applied for group recommendation. To alleviate the data sparsity issue in GRSs, we first study the cross-domain group recommendation problem and propose a hierarchical attention network-based cross-domain group recommendation method, called HAN-CDGR. HAN-CDGR takes the advantage of data from a source domain to benefit recommendation generation for both the individual users and groups in the target domain which has data sparsity and cannot generate accurate recommendation. In HAN-CDGR, a hierarchical attention network is constructed to learn and model individual and group preferences, with consideration of both group members' interactions and dynamic weights and the complex relationships between individuals and groups. Adversarial learning is used to effectively transfer knowledge from a source domain to the target domain. Extensive experiments, which demonstrate the effectiveness and superiority of our proposal, providing accurate recommendation for both individual users and groups, are conducted on three tasks.**

***Index Terms*— Cross-domain recommender systems (CDRSs), group recommender systems (GRSs), hierarchical attention network, recommender systems.**

Manuscript received 1 November 2020; revised 19 October 2021 and 20 May 2022; accepted 15 August 2022. This work was supported in part by the Australian Research Council (ARC) under the Australian Laureate Fellowship under Grant FL190100149, in part by the National Natural Science Foundation of China under Grant 71871228, and in part by the China Scholarship Council. *(Ruxia Liang and Qian Zhang contributed equally to this work.) (Corresponding author: Jie Lu.)*

Ruxia Liang was with the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Faculty of Engineering and Information Technology, Australian Artificial Intelligence Institute (AAII), University of Technology Sydney, Ultimo, NSW 2007, Australia. She is now with the National Engineering Research Center for Educational Big Data and the Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China (e-mail: rxliang@ccnu.edu.cn).

Qian Zhang and Jie Lu are with the Faulty of Engineering and Information Technology, Australian Artificial Intelligence Institute, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: qian.zhang-1@uts.edu.au; jie.lu@uts.edu.au).

Jianqiang Wang is with the School of Business, Central South University, Changsha 410083, China (e-mail: jqwang@csu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3200480.

Digital Object Identifier 10.1109/TNNLS.2022.3200480

Nomenclature

$\mathcal{T} = \{\mathcal{G}, \mathcal{U}^t, \mathcal{V}, \boldsymbol{G}, \boldsymbol{R}_t\}$	Target domain.
$\mathcal{S} = \{\mathcal{U}^s, \mathcal{V}^s, \boldsymbol{R}_s\}$	Source domain.
$\mathcal{K}_l$	Set of user IDs in group g_l.
$\mathcal{H}_i$	Set of group IDs that user u_i has joined.
$\boldsymbol{u}_i$, $\boldsymbol{v}_j$	Embedding vectors of target user $u_i \in \mathcal{U}^t$ and item $\boldsymbol{v}_j \in \mathcal{V}$.
$\boldsymbol{u}^s_{i'}$, $\boldsymbol{v}^s_{j'}$	Embedding vectors of source user $u^s_{i'} \in \mathcal{U}^s$ and item $v^s_{j'} \in \mathcal{V}^s$.
$\boldsymbol{q}_l$	Embedding vector of group $g_l \in \mathcal{G}$.
d	Dimension of embedding vectors.
$\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$	Vectors of queries, keys, and values of all members in a group.
d_k	Dimension of query and key vector.
$\boldsymbol{P}_i$, $\boldsymbol{P}_j$, $\boldsymbol{W}_i$	Weight matrices of neural networks.
$\boldsymbol{h}_o$, $\boldsymbol{h}_\tau$, $\boldsymbol{b}$	Weigh and bias vectors for neural networks.
$\hat{\mathbf{u}}_{l,i}$	User u_i's group-specific embedding in group g_l.
$a(j,i)$	u_i's contribution score in group decision on item v_j.
$\alpha_{l,i}$	u_i's group-specific attention on group g_l.
O^s, O^t, and O^g	Sets of training examples in source and target domains.
$\varphi_{\text{pooling}}(\cdot)$	Pooling function.
$\boldsymbol{e}^s$, $\boldsymbol{e}^t$	Pooling results in source and target domains.
λ, β, and γ	Parameters in gradient reversal layer and total user loss function.
D	Domain discriminator.
$\{\theta^s_p, \theta^u_p, \theta^g_p, \theta_c\}$	Parameter sets of HAN-CDGR.
$\hat{y}^s_{i'j'}, \hat{y}^t_{ij}, \hat{y}^g_{lj}$	Predicted preference scores in both domains.

I. Introduction

RECOMMENDER systems are developed to alleviate the information overload problem to help users retrieve valuable information and find the most suitable services. They are widely applied in many web applications such as video platforms, online stores, or social media [1]. With the rapid growth of digital social networking, many web applications now allow people to participate in various activities in groups, e.g., joining a video channel on YouTube, team buying for discount price, and hiking or traveling with families or friends. Under these circumstances, the products/services that consumers adopted are targeted for a group of users rather than for individuals. The demand of personalized items or services for groups has prompted the development of group recommender systems (GRSs) and its applications in various fields such as tourism [2], entertainment [3], and catering services [4].

Modeling preferences of group members and how they contribute to the group preference act as an essential role in GRSs. Data sparsity, however, is severe in GRSs due to the limited observations of user and group interactions compared with the large number of items [5], [6], which severely impairs the accuracy of user/group preference modeling and recommendation performance. To alleviate the data sparsity issue, some methods are developed to model groups' preferences with various side information of users or groups such as social relations [7], [8], [9], [10]. However, such side information is often unavailable for user/group preference modeling. Another solution is to use transfer learning to borrow data from a related source domain to assist the recommendation tasks in a sparse target domain, which is also known as cross-domain recommender systems (CDRSs) [11]. The core assumption of CDRSs is that similar users in different domains also have similar preferences. When the users change from individuals to groups, however, it is more difficult to extract shared knowledge between two domains due to the complex relationship between members in a group. Hence, CDRSs for individual users cannot be directly applied to provide recommendations for groups.

To deal with the complex member–group relationship and develop cross-domain GRSs, there are three challenges.

1) Group decisions are structured collaboratively so that group members are mutually dependent on each other yet personally impact differently on different items. Thus, a comprehensive strategy to model group preference is very challenging.
2) One user may take part in more than one group and the user preference will adaptively change inside different groups [12], [13]. The different relationships each user has across the different groups to which they belong can be named as containment relationships which is hard to capture and seldom considered in group member preference modeling.
3) Different from the CDRSs for individual users, the knowledge transferred from the source domain in cross-domain group recommendation needs to benefit preference modeling and recommendation for both individual users and groups. Whether the knowledge transfer is effective and how to balance the effectiveness on both sides remains an unsolved problem.

To address the above three challenges, we propose a **h**ierarchical **a**ttention **n**etwork-based **c**ross-**d**omain **g**roup **r**ecommendation method, named HAN-CDGR, to boost group recommendation performance. First, we construct a hierarchical attention neural network to model the preferences of both individual users and groups in a common feature space to allow them to reinforce each other. The first level of attention is used to consider the inner relationships between group members; the other level of attention handles complex interdependencies between the users and groups. This hierarchical attention neural network solves challenges 1) and 2). Second, to solve challenge 3), we apply adversarial learning and domain adaptation methods to learn transferable latent representations of a user–item pair from both the source and target domains and use the inner product module to generate user–/group–item predictions. In summary, the main contributions of this article are as follows.

1) *Group Preference Module to Represent Group Preferences With Knowledge of Both Individual– and Group–Item Interactions:* It enables more precisely group preference modeling by comprehensively considering the influence between group members and adaptive weights of each member in this group.
2) *Individual User Preference Module That Is Able to Deal With a User's Containment Relationship in Multiple Groups:* This module relaxes the constraints that a user has fixed preferences and represents one user with different preferences when he/she participates in different groups, namely, group-specific user embedding, which is more suitable and flexible for real-world situations.
3) *Adversarial Learning-Based Knowledge Transfer Method for Cross-Domain Group Recommendation:* This method can alleviate the data sparsity issue in GRSs and improve the recommendation for both groups and individual users by extracting knowledge from a source domain. To the best of our knowledge, it is the first work that considers cross-domain knowledge transfer for group recommendation tasks.
4) *Novel **H**ierarchical **A**ttention **N**etwork-based **C**ross-**D**omain **G**roup **R**ecommendation Called HAN-CDGR:* Extensive experiments are performed on three tasks with six real-world datasets to verify the effectiveness of HAN-CDGR, which significantly improves the recommendation performance of both individual users and groups and alleviates the data sparsity issue.

In the rest of this article, Section II provides a review of related work. In Section III, we formally define the research problem and describe our proposed method HAN-CDGR. In Section IV, we conduct extensive experiments and provide analysis on the experimental results. Finally, we summarize this article with conclusions and future research directions in Section V.

II. Related Work

This study has three highly related research topics: group recommendation, hierarchical neural networks in recommender systems, and CDRSs. This section will review these three areas of research, respectively.

A. Group Recommender System

GRSs aim to generate satisfactory recommendations for groups from various items or services, where group preference modeling plays a vital role in recommendation performance.

Different GRSs have different group preference modeling components, which is developed according to the natural characteristics of groups. According to the group member mobility, groups are divided into two types: persistent groups and occasional groups. GRSs for different group types are heavily sensitive to data [14], and thus they are often verified on different datasets to meet their specific group characteristics. Usually, a persistent group has existed for a long period, so that group members become closely correlated and have some interactions with items within this group. In contrast, occasional groups are usually passively formed where the members do not know each other before and cannot negotiate a consensus preference for group decision-making. Therefore, there exist extremely weaker inner relationships among group members and even fewer group–item interactions in occasional groups than the persistent ones [2], [15].

Two categories of GRSs have been developed: the memory-based and the model-based group recommendation methods. Fixed aggregation strategies, such as AVeraGe (AVG) [16], [17], least misery (LM) [18], and maximum satisfaction (MS) [19], or some improved weighted aggregation strategies [2], [20], are widely used for fusing group member preferences in previous memory-based group recommendation methods that lack flexibility and rationality. Compared with these, the proposed HAN-CDGR models both user and group preferences from data by designing a deep neural network with a better flexibility.

Compared with the memory-based types, the model-based group recommendation methods have received more attention, including shallow methods [4], [21], [22] and deep learning methods [7], [14], [23], [24], [25]. The singular-value-decomposition-based group recommendation (SVD-based GR) methods [4] were applied to generate group recommendations by integrating diverse fixed aggregation strategies. Some probabilistic models, such as personal impact topic (PIT) [21] and consensus model (COM) [26], were proposed to model group profiles with consideration of group members' personal impacts and related topics.

The existing deep leaning group recommendation methods are mainly divided into persistent group recommendations [7], [14], [25], [27] and occasional group recommendations [5], [6], [8], [10], [28], [29], [30]. For the first persistent group recommendation type, deep learning group recommender (DLGR) was the first deep architecture model to learn groups' high-level features [14]. Attentive group recommendation (AGREE) and social-enhanced AGREE (SoAGREE) used attention mechanisms to learn groups' preferences adaptively with respect to the specific items under consideration [7], [25]. However, they overlooked the complex inner relationships among persistent group members and modeled group preferences by fusing fixed group members' preferences. To tackle this issue, Vinh Tran *et al.* [27] represented each group member using a single subattention network to model the interactions between the group member and all the other members in the group. HAN-CDGR is for persistent group recommendation. It is different from the existing persistent group recommendation methods by designing a hierarchical neural network to learn both user and group adaptive preferences.

The occasional group recommendation methods focus on alleviating the data sparsity issue through better modeling user–/group–item interactions [5], [6] or borrowing knowledge from auxiliary information, such as social network [8], [10], [28], [30]. Graphical and attentive multiview embeddings (GAME) and group recommendation using attentive dual influences (GRADI) represented user–/group–item interactions as various graph data and learned the latent representations of the groups, users, and items from multiple independent views [5], [6]. Social-influence-based group recommender (SIGR), centrality-aware group recommender (CAGR), and group self-attention (GroupSA) leveraged the user social networks to enhance user/group preferences [8], [10], [28]. Furthermore, hierarchical hyperedge embedding-based group recommender (HyperGroup) improved group preference modeling by exploiting the group similarity [30]. The group preference modeling component of GroupSA is most similar to our proposed method HAN-CDGR, which both used a self-attention network to capture inner relationships among group members. Since GroupSA tackled occasional group recommendation, it only considered inner relationships among group members that have direct social connections. Different from these, HAN-CDGR caters to persistent groups and assumes all the group members have inner relationships. Moreover, it assumes that no auxiliary information is available and tries to alleviate the data sparsity issue by incorporating information from other domain knowledge. Table I summarizes the key characteristics of the aforementioned group recommendation methods.

B. Hierarchical Neural Networks in Recommender Systems

Hierarchical classification and hierarchical clustering are the two main paradigms in machine learning [32], [33], which have been used in the field of recommender systems [7], [34]. Park and Kim [33] proposed an adaptive resonance theory-supervised predictive mapping for the hierarchical classification network and applied it to a multimedia recommendation systems for digital storytelling. Zhong *et al.* [35] proposed a novel hierarchical and interactive gate network for rating prediction, which modeled local word informativeness and global review semantics in the reviews of users/items in a hierarchical manner.

For group recommendation, SoAGREE [7] used users' social connections to construct a hierarchical attention network for group recommendation. In [34], a Facebook application HappyMovie was developed to recommend movies to groups. Furthermore, Quijano-Sánchez *et al.* [13] exploited the hierarchical relationships within a group based on the influence of social relationships between individuals. When groups contain thousands of group members, Qin *et al.* [36] divided big groups into many common-interest user subgroups using member clustering and generated big group recommendations by aggregating subgroup recommendation lists. When the groups have not been built in advance, the Leuven algorithm was applied to GRSs to identify user groups automatically

TABLE I
SUMMARY OF EXISTING GROUP RECOMMENDATION METHODS

Method	Input			User/Item feature		Aggre strategy		Group type	
	explicit	implicit	other infor	fixed	dynamic	fixed	dynamic	persistent	occasional
MC-GR [2]	×			×			×		×
Groupality [20]	×			×		×		×	
TruGRC [31]	×		×	×		×		×	
SVD-based GR [4]	×			×		×		×	
PIT [21]		×	×	×			×		×
COM [26]		×	×		×	×			×
CVTM, GERF [3], [22]		×	×	×		×			×
DLGR [14]	×	×			×	×		×	
AGREE [25]		×		×			×	×	
MoSAN [27]		×		×			×	×	
GAME, GRADI [5], [6]		×			×		×		×
SIGR, CAGR, GroupSA, and HyperGroup [8], [10], [28], [30]		×	×		×		×		×
HAN-CDGR		×			×		×	×	

in [37]. Previous hierarchical neural-network-based group recommenders usually relied on users' auxiliary information, while our method focuses on alleviating data sparsity issue with no auxiliary information available.

C. Cross-Domain Recommender Systems

Data sparsity is acute for recommender systems, especially when the recommender system is newly launched. CDRS is one effective method to alleviate the data sparsity issue, where deep learning methods have been the mainstream in the existing CDRSs research. One category of CDRSs investigates how to apply additional information, such as reviews, user/item profiles, and tags, to improve the recommendation effectiveness on either the target domain or both the source and target domains. Discriminative adversarial networks for cross-domain recommendation (RecSys-DAN) [38] transfers the latent representations from a source domain to a target domain in an adversarial way. The dual-target CDR [39] and deep dual-transfer cross-domain recommendation [40] are two dual-target CDRs that adapt rating and multisource content information to generate user/item latent features. Another category tackles the data sparsity issue through the transfer of knowledge from one or more relevant source domains, which have relatively rich user–item interaction data, to a target domain [41], [42], [43]. The collaborative cross networks [41] introduced cross connections between two networks to allow dual knowledge transfer across domains with implicit feedback. This method assumed user fully overlapping across domains. Neural attentive transfer recommendation (NATR) [42] transfers item embeddings across domains through attention networks. Taking the merits of adversarial learning, the deep domain adaptation recommendation model [44] extracts and transfers rating patterns from rating matrices from the user or item side. On this basis, a deep dual adversarial network is proposed to transfer both the user side and item side domain shared information [43]. However, none of these cross-domain recommendation methods has been applied to group recommendation. Aiming at the group recommendation challenge issue, this study proposes a cross-domain group recommendation method that applies representation learning, adversarial learning, and domain adaptation methods to learn transferable latent representations for users, items, and their interactions across domains to contribute to both user and group recommendations.

III. HIERARCHICAL ATTENTION NETWORK-BASED CROSS-DOMAIN GROUP RECOMMENDATION METHOD

This section first introduces the notations and problem formulation, then introduces the motivations for developing the HAN-CDGR method through a concrete example for cross-domain group recommendation, gives an overview of the entire method, and finally explains in detail each module of the method.

A. Notations and Problem Definition

In this work, we aim to tackle a cross-domain group recommendation problem which involves two domains: a target domain and a source user domain. The target domain involves two interaction matrices with implicit feedback: the group–item interaction matrix and the user–item interaction matrix. Suppose we have S groups $\mathcal{G} = \{g_1, g_2, \ldots, g_S\}$ and N^t items $\mathcal{V} = \{v_1, v_2, \ldots, v_{N^t}\}$; the groups' implicit feedback on items is represented as $\boldsymbol{G} \in \mathbb{R}^{S \times N^t}$ (bold uppercase letter represents a matrix), where an element of $\boldsymbol{G}$ is 1 if the corresponding group–item interaction is observed (i.e., a group purchases an item) and empty otherwise. Since each group in the group set $\mathcal{G}$ is formulated by individual users, the user–item interaction matrix containing M^t users $\mathcal{U}^t = \{u_1, u_2, \ldots, u_{M^t}\}$ can be denoted as $\boldsymbol{R}_t \in \mathbb{R}^{M^t \times N^t}$. Considering user–/group–item interactions in the target domain are usually sparse, an auxiliary source domain, denoted as $\boldsymbol{R}_s$, is introduced to help group (or the user $u_i \in \mathcal{U}^t$) recommendation in the target domain. Suppose the source domain $\boldsymbol{R}_s$ has M^s users $\mathcal{U}^s = \{u_1^s, u_2^s, \ldots, u_{M^s}^s\}$ and N^s items $\mathcal{V}^s = \{v_1^s, v_2^s, \ldots, v_{N^s}^s\}$, the source user–item implicit interaction matrix is denoted as $\boldsymbol{R}_s \in \mathbb{R}^{M^s \times N^s}$. Particularly, the user–item interaction matrices $\boldsymbol{R}_s$ and $\boldsymbol{R}_t$ are from the same or similar product domains. For example, people who like romance novels may also enjoy watching romantic movies. Therefore, a book dataset with relatively rich data can be regarded as a source domain of

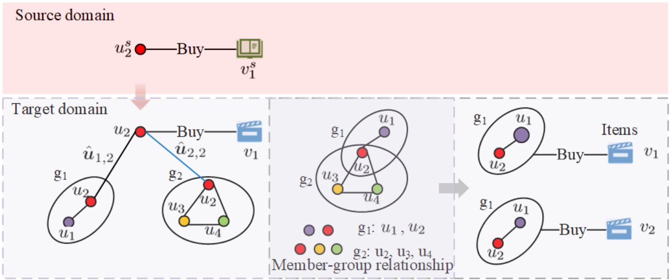


Fig. 1. Concrete example for cross-domain group recommendation. There is no group label in the source domain data, while there are both user-/group–item interactions and member–group relationship information in the target domain. The member–group relationships reflect both the inner relationships among group members and containment relationship between users and various groups. Lines between users in each group indicate the inner relationships among group members, while the lines between users and groups represent containment relationships between them. In a group, the circles with different colors represent different users, and the circle size of each group member means its decision power. The bigger the circle, the bigger the decision power the user has within this group. The knowledge from the source user domain is first transferred to the target user domain and then contributes to group recommendation in the target domain.

a movie dataset which is sparser than the domain of books. We aim to learn some shareable and transferable information on users, items, and their interactions across the source user and target user domains in an adversarial way and boost the recommendation performance on both individual users and groups in the target domain. Then, our cross-domain group recommendation task is defined as recommending a list of items that group $g_l \in \mathcal{G}$ (or target user $u_i \in \mathcal{U}^t$) may be interested in through using information from both the source and target domains.

B. Example for Cross-Domain Group Recommendation

We intend to propose a cross-domain group recommendation method that can be applied to generate recommendations for small group activities, such as student group study, family TV program watching, and friends' travel decisions. Fig. 1 illustrates three challenging scenarios.

1) *Group Preference Modeling:* In each social group, there exist complex inner relationships among group members due to the social nature of human tourists intend. Moreover, when the group is faced with different types of items, the group's decision power distribution is dynamic due to the differences in group members' roles and expertise. For example, when a group of tourists intends to engage in activities such as rock climbing or hiking, the more experienced hikers in the group may contribute more than those inexperienced ones to the group decision. However, when the tour group chooses to visit cultural sites, the group members who are more familiar with various cultural sites often play a bigger role in the group decision. As shown in Fig. 1 (right), the decision power of group member u_1 and u_2 varies when the group is faced with items v_1 and v_2.
2) *Group Member Preference Modeling:* Due to the social nature of humans, each user is likely to be influenced by other group members within diverse groups. For example, a user who likes reading books inherently may prefer hiking and rock climbing when he becomes a member of a group for sports, whereas he is more likely to watch movies and play games when he attends an entertainment group. As shown in Fig. 1 (left), user u_2 has taken part in groups g_1 and g_2 simultaneously, and u_2's group-specific preference toward g_1 and g_2 can be denoted as $\hat{\boldsymbol{u}}_{1,2}$ and $\hat{\boldsymbol{u}}_{2,2}$, respectively.
3) *Cross-Domain Group Recommendation:* As shown in Fig. 1 (above), there is a source domain containing similar users with those in the target user domain. Users in the target domain can act as a bridge between the source user and the target group. The source domain data first contribute to the user preference modeling in the target domain in an adversarial manner. Then, to efficiently transfer knowledge from the source domain to the target group domain, we need to properly model complex relationships between users and groups in the target domain. In this way, knowledge from the source domain can be transferred to the group preference modeling module by aggregating group member preferences.

The existing GRSs fail to model complex inner relationships among group members and the containment relationships between users and various groups. CDGRs for individual users are unable to deal with the data sparsity issue in GRSs. The HAN-CDGR method, described in Section III-C, helps solve these challenges.

C. HAN-CDGR Method Overview

As shown in Fig. 2(a), the HAN-CDGR method consists of four main modules: Module 1) group preference modeling (M1); Module 2) user preference modeling (M2); Module 3) adversarial-learning-based knowledge transfer (M3); and Module 4) top-N item recommendation (M4). The first two modules aim to learn latent representations of users and groups through a hierarchical attention neural network. It should be noted that both user and group preferences can be augmented with multimodule features, such as images and texts, via customized encoders [45]. The adversarial-learning-based knowledge transfer module aims to learn transferable user–item latent features between the source and target domains to alleviate the data sparsity issue of target domain. Finally, both the user– and group–item prediction scores are generated through an inner product module. We list some important notations associated with HAN-CDGR in Nomenclature section. Next, we explain each module of the HAN-CDGR method in detail.

D. M1: Group Preference Modeling

Let $g_l \in \mathcal{G}$ be the target group containing a set of users. The set of user IDs in group g_l is represented as $\mathcal{K}_l$. Let $\boldsymbol{v}_j$ be the embedding vector of item v_j. The target group g_l's overall embedding on item v_j is formulated as

$$\boldsymbol{g}_l(j) = \sum_{i \in \mathcal{K}_l} a(j,i)\hat{\boldsymbol{u}}_{l,i} + \boldsymbol{q}_l \quad (1)$$

where $\hat{\boldsymbol{u}}_{l,i}$ is the group-specific embedding vector of member u_i in group g_l. $a(j,i)$ means the weight of u_i in the group

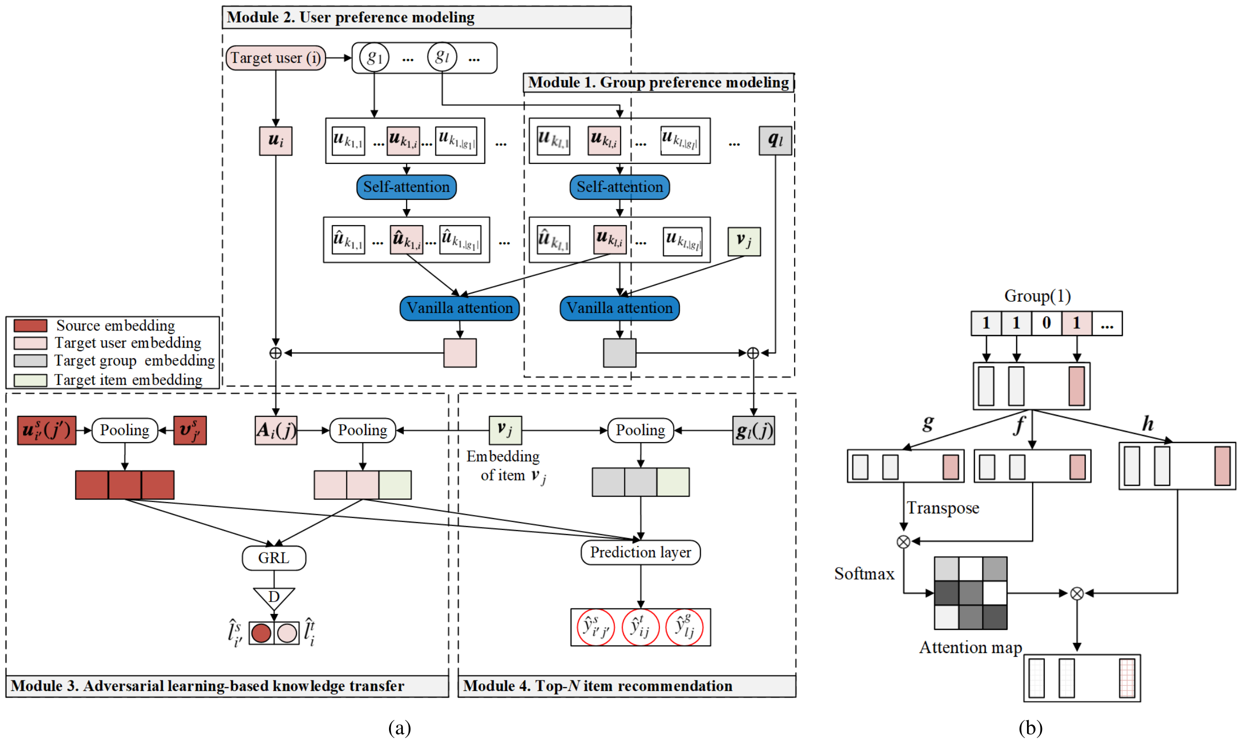


Fig. 2. (a) Overall framework of our proposed HAN-CDGR method. It contains four modules: group preference modeling, user preference modeling, adversarial-learning-based knowledge transfer, and top-N recommendation. (b) Details about the self-attention module.

decision on item v_j, and $\boldsymbol{q}_l$ denotes the target group g_l's inherent embedding of dimension d. The two components of (1) are denoted as M1-a and M1-b, respectively. We elaborate on these two components in the following.

1) M1-a: Group Member Embedding Aggregation: A group can be regarded as a global input that is composed of several interacting parts. In (1), we first use a self-attention neural network to improve the group members' representations with consideration of the group members' inner relationships to get $\hat{\boldsymbol{u}}_{l,i}$; next use a vanilla attention network to learn group members' dynamic weights on different items and aggregate the group members' improved representations as a whole.

a) Self-attention neural network: The self-attention network aims to model user u_i's group-specific embedding $\hat{\boldsymbol{u}}_{l,i}$. It is intuited that one group member's preference for an item could be affected by other members, such that the attention for some members could be distracted and others could be strengthened. Along this view, the member user-improved embedding $\hat{\boldsymbol{u}}_{l,i}$ can be partly determined as a weighted combination of features from all the members in the group

$$\begin{aligned}\boldsymbol{O} &= \text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\\ &= \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \quad (2)\end{aligned}$$

$$\hat{\boldsymbol{u}}_{l,i} = \boldsymbol{m}^T\, \boldsymbol{O} + \boldsymbol{u}_i \quad (3)$$

where $.^T$ indicates the transposition operation, matrices $\boldsymbol{Q} \in \mathbb{R}^{|\mathcal{K}_l| \times d_k}$, $\boldsymbol{K} \in \mathbb{R}^{|\mathcal{K}_l| \times d_k}$, and $\boldsymbol{V} \in \mathbb{R}^{|\mathcal{K}_l| \times d}$ are packed by the queries, keys, and values of all the members in a group, and the query, key, and value vectors of each member in this group are obtained by multiplying the member embedding of dimension d with three corresponding trainable embedding matrices. Since $\mathcal{K}_l$ represents the set of user IDs in group g_l, the cardinality $|\mathcal{K}_l|$ means the group size of g_l. As shown in Fig. 2(b), three mapping functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ represent three multilayer perceptron (MLP) neural networks which map the group embedding matrix $\boldsymbol{G}_l$ into matrices $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, respectively. Specifically, $\boldsymbol{G}_l \in \mathbb{R}^{|\mathcal{K}_l| \times d}$ is the input embedding layer of the target group g_l, and each row of $\boldsymbol{G}_l$ represents the embedding of each member within this group. $(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d_k})$ is calculated as an input of a $softmax$ layer to determine a normalized weight (or attention) matrix of size $|\mathcal{K}_l| \times |\mathcal{K}_l|$, where $(1/\sqrt{d_k})$ is a scaling factor to alleviate the effect of extremely small gradients when the value of d_k is too large [46]. Each element of the weight matrix indicates the extent to which the group attends to one member when synthesizing another member in this group. Then, the output matrix $\boldsymbol{O} \in \mathbb{R}^{|\mathcal{K}_l| \times d}$ of the self-attention layer can be obtained by multiplying the normalized weight matrix with $\boldsymbol{V}$. Each row of the output matrix $\boldsymbol{O}$ represents the hidden representation of each member in g_l, by attending over other group members in the group. In addition, $\boldsymbol{m}$ is a group member indicator vector, $m_i \in \{0, 1\}$ and $i = \{1, 2, \ldots, |\mathcal{K}_l|\}$. $m_i = 1$ indicates the group member u_i; otherwise, $m_i = 0$. To obtain u_i's hidden representation in $\boldsymbol{O}$, we multiply the group member indicator vector with the output matrix $\boldsymbol{O}$. We further associate a member user u_i's hidden representation with its own dedicated embedding $\boldsymbol{u}_i$ to determine u_i's group-specific embedding $\hat{\boldsymbol{u}}_{l,i}$.

b) Vanilla attention neural network: The vanilla attention neural network aims to learn the group members' adaptive weights for different unrated items, which are used to compute a linear combination of group members' improved embedding vectors corresponding to them, to serve as the output preference for group member embedding aggregation

$$o(j,i) = \boldsymbol{h}_o^T \mathrm{ReLU}(\boldsymbol{P}_j \boldsymbol{v}_j + \boldsymbol{P}_i \hat{\boldsymbol{u}}_{l,i} + \boldsymbol{b}) \tag{4}$$

$$a(j,i) = \frac{\exp o(j,i)}{\sum_{t' \in \mathcal{K}_l} \exp o(j,t')} \tag{5}$$

where $\boldsymbol{P}_j$ and $\boldsymbol{P}_i$ represent the weight matrices of the vanilla attention network that project the item embedding $\boldsymbol{v}_j$ and member user improved embedding $\hat{\boldsymbol{u}}_{l,i}$ into a shared latent feature space, respectively, and $\boldsymbol{b}$ is the bias vector of the hidden layer. Then the rectified linear unit (ReLU) function is used as the activation function of the hidden layer, and the attention score $o(j,i)$ is obtained by multiplying a weight vector $\boldsymbol{h}_o$ with the output of hidden layer. Specifically, the output of the hidden layer is transformed by a fully connected layer and normalized by a $softmax$ function to produce the attention weight $a(j,i)$ of member user u_i on item v_j in group g_l. The member weight represents the overall contribution score of the member user in a group's overall decision on an item after negotiating with other group members.

2) M1-b: Group Inherent Embedding: In (1), the group inherent embedding of g_l, defined as $\boldsymbol{q}_l$, represents the general preference of the group, which is an indispensable part of the group's overall preference, especially when there is inconsistency between these two parts. For example, several hiking enthusiasts are still likely to sign up for a group tour of a cultural landscape due to corporate team-building activities. Therefore, the final output of group–item preference is given by (1) which adds the group's inherent embedding to the group members' aggregated embedding.

E. M2: User Preference Modeling

Let u_i be the target user in $\mathcal{U}^t$ which has simultaneously participated in a set of groups with index set $\mathcal{H}_i$. All the groups that u_i has joined come from the group set $\mathcal{G}$. To model the user preference with consideration of the complex containment relationships between the user and diverse groups, the target user u_i's overall embedding on item v_j can be formulated as

$$\boldsymbol{A}_i(j) = \sum_{l \in \mathcal{H}_i} \alpha_{l,i} \hat{\boldsymbol{u}}_{l,i} + \boldsymbol{u}_i \tag{6}$$

where the first component is obtained by aggregating u_i's group-specific embedding $\hat{\boldsymbol{u}}_{l,i}$ from diverse groups that u_i has joined. $\alpha_{l,i}$ denotes u_i's group-specific attention on group g_l, and it also reflects how much group g_l contributes to u_i's embedding. The second component is the user u_i's inherent embedding. We will illustrate these two components below.

1) M2-c: Group-Specific User Embedding Aggregation: To determine the aggregated embedding of target user u_i, we first use the self-attention mechanism to capture the inner relationships among different members in one group and get the self-attentive output. The same process is conducted on each group that the user u_i belongs to, and $|\mathcal{H}_i|$ self-attentive outputs are obtained, where cardinality $|\mathcal{H}_i|$ means the total number of groups that u_i has joined. The calculation processes are the same as (2) and (3), thus are omitted here. Then, an appropriate aggregation strategy is required to fuse all these group-specific embedding vectors $\hat{\boldsymbol{u}}_{l,i} (l \in \mathcal{H}_i)$. To capture the dynamic weights of one member in diverse groups, we use a vanilla attention network to calculate the attention scores of the target user belonging to different groups

$$\tau_{l,i} = \boldsymbol{h}_\tau^T \mathrm{ReLU}(\boldsymbol{W}_i \hat{\boldsymbol{u}}_{l,i} + \boldsymbol{b}) \tag{7}$$

$$\alpha_{l,i} = \frac{\exp(\tau_{l,i})}{\sum_{l' \in \mathcal{H}_i} \exp(\tau_{l',i})} \tag{8}$$

where $\alpha_{l,i}$ means the normalized weight of target user u_i in group g_l, and $\boldsymbol{W}_i$ and $\boldsymbol{b}$ are the learned weight matrix and bias vector, respectively. $\boldsymbol{h}_\tau$ is a global vector, and ReLU function is our chosen activation function. Finally, the $softmax$ function is used to normalize the weights across all the groups that u_i has joined.

2) M2-d: User Inherent Embedding: In (6), the user inherent embedding is denoted by $\boldsymbol{u}_i$ which represents an independent preference of the target user u_i in $\mathcal{R}^t$. Finally, we calculate the overall user–item preference by (6) with consideration of both the user's initial preference embedding and group-specific embedding integration.

F. M3: Adversarial-Learning-Based Knowledge Transfer

In this section, we aim to transfer knowledge from a source domain ($\boldsymbol{R}_s$) to the target domain ($\boldsymbol{R}_t$ and $\boldsymbol{G}$) through adversarial training to assist the recommendation for individual users in $\mathcal{R}^t$, and further contribute to the recommendation for groups in $\mathcal{G}$. As depicted in Fig. 2(a), the adversarial-learning-based knowledge transfer module is composed of two components: a pooling layer and adversarial learning. Specifically, the pooling layer aims to get a comprehensive representation of the user–item pair from both the source and target domains, which are trained to be as similar as possible so that a classifier cannot reliably decide which domain the feature vector is from. Meanwhile, the domain discriminator D in the adversarial learning layer aims to make accurate discrimination between the source and target domains.

1) Pooling Layer: For the source domain, following [25], the source user preference and item property are extracted as embedding vectors under the representation learning framework. The pooling layer first performs elementwise product on the source user–item embedding vector pair $(\boldsymbol{u}^s_{i'}(j'), \boldsymbol{v}^s_{j'})$, and then concatenates it with the original embedding $\boldsymbol{u}^s_{i'}(j')$ and $\boldsymbol{v}^s_{j'}$. For the target domain, the inputs are user feature vector $\boldsymbol{A}_i(j)$ and item embedding $\boldsymbol{v}_j$ for each target user–item pair (u_i, v_j). The user feature vector $\boldsymbol{A}_i(j)$ is obtained by the user preference modeling (for details, see Section III-E). The pooling results of the source domain are

$$\boldsymbol{e}^s_{i'j'} = \varphi_{\text{pooling}}(\boldsymbol{u}^s_{i'}(j'), \boldsymbol{v}^s_{j'}) = \begin{bmatrix} \boldsymbol{u}^s_{i'}(j') \odot \boldsymbol{v}^s_{j'} \\ \boldsymbol{u}^s_{i'}(j') \\ \boldsymbol{v}^s_{j'} \end{bmatrix} \tag{9}$$

where $\varphi_{\text{pooling}}(\cdot)$ means the pooling function, and $\odot$ denotes the elementwise product of two vectors. $[\cdot]$ represents the concatenation of feature vectors. Similarly, the pooling result of (user, item) pair in the target domain is denoted as $\boldsymbol{e}^t_{ij}$.

2) Domain Discriminator D: The domain discriminator D with parameter set θ_c contains one domain classifier and one gradient reverse layer (GRL), which aims to match the latent features of the source user–item pair and the target user-=item pair and facilitate positive knowledge transfer between both the domains through adversarial training. Specifically, the GRL is inserted between the pooling layer and the domain classifier [47]. It behaves differently for the forward and back propagation processes, which acts as an identity transform for forward propagation, while it reverses the gradient direction by multiplying the gradient by $-\lambda$ in the backward propagation

$$\begin{aligned} G(\boldsymbol{x}) &= x \ \ (\text{forward propagation}) \\ \frac{dG(\boldsymbol{x})}{d\boldsymbol{x}} &= -\lambda \boldsymbol{I} \ \ (\text{backward propagation}) \end{aligned} \tag{10}$$

where $\boldsymbol{I}$ is an identity matrix, and λ is a hyperparameter.

We use domain classifier to predict the domain label $\hat{l}_i \in \{0, 1\}$ of the ith example, which comes from both the source and target user domains, where 0 indicates the source domain and 1 for the target domain, respectively. Binary cross-entropy loss is used to measure the domain classifier loss

$$L_c(\theta_c) = -\sum_{i=1}^{|O^s|+|O^t|} l_i \log \hat{l}_i + (1 - l_i) \log (1 - \hat{l}_i) \tag{11}$$

where $|*|$ means the number of training examples, and l_i and $\hat{l}_i$ are the real and predicted domain labels of the ith training example from both the domains.

G. M4: Top-N Item Recommendation

1) Prediction Layer: The interaction prediction for a given pair of user u_i (or source user $u^s_{i'}$ and group g_l) and item v_j(or source item $v^s_{j'}$) is achieved by an inner product module with parameter sets θ^u_p, θ^s_p, and θ^g_p, respectively. Take an target user–item pair (u_i, v_j) as an example, the user feature vector $\boldsymbol{A}_i(j)$ obtained by (6) and item feature vector $\boldsymbol{v}_j$ are fed into this inner product module and output the interaction probability $\hat{y}^t_{ij}$ between the pair of (u_i, v_j). The estimated value $\hat{y}^t_{ij}$ is calculated by

$$\hat{y}^t_{ij} = \sigma(\boldsymbol{A}^T_i(j) \cdot \boldsymbol{v}_j) \tag{12}$$

where σ represents the *sigmoid* function, and $.^T$ means the transposition operation. The prediction loss for the target user domain is as below

$$\begin{aligned} L^t_p(\theta^u_p) &= \sum_{(i,j,n)\in O^t} \left(y^t_{ijn} - \hat{y}^t_{ijn}\right)^2 \\ &= \sum_{(i,j,n)\in O^t} \left(\hat{y}^t_{ij} - \hat{y}^t_{in} - 1\right)^2 \end{aligned} \tag{13}$$

where O^t denotes the training set of the target user domain $\boldsymbol{R}_t$. Each instance in O^t is a triplet (i, j, n), meaning that the target user u_i has interacted with item v_j, but not with item v_n. Similarly, the other two prediction losses for the source user–item and group–item interactions in both the domains are defined as $L^s_p(\theta^s_p)$ and $L^g_p(\theta^g_p)$, respectively.

2) Objective Function: The total loss function with parameter set $\{\theta^s_p, \theta^u_p, \theta_c\}$ for the recommendation task for individual users is obtained as

$$L_{\text{user}}\left(\theta^s_p, \theta^u_p, \theta_c\right) = L^t_p\left(\theta^u_p\right) + \beta \cdot L^s_p\left(\theta^s_p\right) - \gamma \cdot L_c(\theta_c) \tag{14}$$

where β and γ are two trade-off parameters to balance the contributions of the source preference prediction loss and domain loss. The optimization objective can be further expressed as

$$\begin{aligned} \left(\hat{\theta}^s_p, \hat{\theta}^u_p\right) &= \arg \min_{\theta^s_p, \theta^u_p} L_{\text{user}}\left(\theta^s_p, \theta^u_p, \theta_c\right) \\ \left(\hat{\theta}_c\right) &= \arg \max_{\theta_c} L_{\text{user}}\left(\theta^s_p, \theta^u_p, \theta_c\right). \end{aligned} \tag{15}$$

With the help of GRL, the parameter set can be optimized by stochastic gradient descent or its variants. After training, some domain-invariant and discriminate user–item feature vectors can be found in both the source and target domains, and such features in the source domain $\boldsymbol{R}_s$ can further facilitate the recommendations in $\boldsymbol{R}_t$ and $\boldsymbol{G}$.

After optimizing objection function (15) on both the source user–item and target user–item interactions, we further optimize the group–item interaction prediction function $L^g_p(\theta^g_p)$ on the set of group–item interaction instances O^g

$$\begin{aligned} L^g_p\left(\theta^g_p\right) &= \sum_{\left(l,j,j'\right)\in O^g} \left(y^g_{ljj'} - \hat{y}^g_{ljj'}\right)^2 \\ &= \sum_{\left(l,j,j'\right)\in O^g} \left(\hat{y}^g_{lj} - \hat{y}^g_{lj'} - 1\right)^2 \end{aligned} \tag{16}$$

where (l, j, j') indicates that group g_l has interacted with item v_j but not with item $v_{j'}$. The group's predicted score on item v_j(or $v_{j'}$) is calculated by feeding the group preference by (1) and item embedding $\boldsymbol{v}_j$ (or $\boldsymbol{v}_{j'}$) to the inner product module. The pseudo-code for optimizing HAN-CDGR is summarized as Algorithm 1.

3) Recommendation Generation: After optimizing HAN-CDGR, given a target group g_l (or target user u_i), we can obtain the prediction score of the group g_l (or the user u_i) on a certain item v_j. Then, a recommendation list of items that the group (or the user) might like can be generated.

H. Complexity Analysis

Generally, the complexity of the proposed HAN-CDGR is affected by epoch E, the size of the data in the source domain $|O^s|$ and the target domain $|O^t|$, the user number M^t and group number S in the target domain, and the number of hidden factors K. In our proposed HAN-CDGR, Module 1 contains the user preference modeling and group modeling, which contains one-layer encoder, two-layer self-attention, and two-layer vanilla attention. Thus, the complexity is $O(M^t * (2K^2+K))$. The complexity of Module 2, similar to Module 1 but with group size, is $O(S^*(2K^2 + K))$. For the adversarial part, it contains a GRL and one-layer classifier, and thus the complexity is $O((|O^s|+|O^t|)*(K^2))$. Module 4 contains one-layer prediction. As it involves negative samples so the size of data in the target domain increases to $5|O^t|$, the complexity is $O(5|O^t| * K)$. Overall, the complexity of HAN-CDGR is $O(N*((M^t+S+|O^s|+|O^t|)*2K^2+(M^t+S+5|O^t|)*K))$.

Algorithm 1 HAN-CDGR

Input: $\mathcal{T} = \{\mathcal{G}, \mathcal{U}^t, \mathcal{V}, \boldsymbol{G}, \boldsymbol{R}_t\}$, the target domain;
$\mathcal{S} = \{\mathcal{U}^s, \mathcal{V}^s, \boldsymbol{R}_s\}$, the source domain;
Member–group relationship information.
Output: The preference predictions $\hat{y}^t_{ij}$ and $\hat{y}^g_{lj}$ in target domain.
Randomly initialization
While $epoch < E$ **do**:
Randomly shuffle the training data
foreach $(u^s_{i'}, v^s_{j'}) \in O^s$ **do**
  Get $\boldsymbol{u}^s_{i'}(j')$, $\boldsymbol{v}^s_{j'}$, and $\boldsymbol{e}^s_{i'j'}$ by (9)
  $\hat{l}(u^s_{i'}, v^s_{j'}) \leftarrow D(\boldsymbol{e}^s_{i'j'})$, $\hat{y}^s_{i'j'} \leftarrow \sigma((\boldsymbol{u}^s_{i'}(j'))^T \cdot \boldsymbol{v}^s_{j'})$
end
foreach $(u_i, v_j) \in O^t$ **do**
  Get $\boldsymbol{A}_i(j)$, $\boldsymbol{v}_j$ by (6), $\boldsymbol{e}^t_{ij}$ by (9)
  $\hat{l}(u_i, v_j) \leftarrow D(\boldsymbol{e}^t_{ij})$, $\hat{y}^t_{ij} \leftarrow \sigma(\boldsymbol{A}^T_i(j) \cdot \boldsymbol{v}_j)$
end
Calculate L_c by (11)
Calculate prediction losses L^t_p and L^s_p by (13)
Calculate the loss function for individual users:
$L_{user} = L^t_p + \beta \cdot L^s_p - \gamma \cdot L_c$
Update the network parameters $\{\theta^s_p, \theta^u_p, \theta_c\}$ through back propagation
foreach $(g_l, v_j) \in O^g$ **do**
  Get $\boldsymbol{g}_l(j)$, $\boldsymbol{v}_j$ by (1), $\hat{y}^g_{lj} \leftarrow \sigma(\boldsymbol{g}^T_l(j) \cdot \boldsymbol{v}_j)$
end
Calculate the loss function L^g_p for groups by (16)
Update the network parameters θ^g_p through back propagation
End
Output: The final predictions: $\hat{y}^t_{ij}$ and $\hat{y}^g_{lj}$.

Considering that K is a much smaller number compared with the size of the data or the number of users/groups and can be treated as constant, the overall complexity of HAN-CDGR is $O(k^2n^2)$. Considering that HAN-CDGR is trained on data from both the source and target domains and it involves two-level attention neural networks, it may take more time to achieve convergence than the group recommendation methods based on a single-attention neural network such as AGREE [25], or individual recommendation methods such as Bayesian personalized ranking (BPR) [48] and neural collaborative filtering (NCF) [49]. Distributed computation may be a solution to speed up our proposed method [50].

IV. Experiments and Analysis

In this section, we conduct extensive experiments on several public datasets to verify the effectiveness of our proposed method. The source codes and datasets used in our experiments have been provided online.[1] The datasets are introduced first, followed by the evaluation protocols, baseline methods, and experimental settings. Then, the experimental results and related analysis are presented. Parameter analysis concludes this section.

[1]https://github.com/ccnu-mathits/HAN-CDGR

TABLE II
Statistical Information on the Six Original Datasets

Data_source	User No.	Item No.	Group No.	G-I/U-I sparsity
Mafengwo	5275	1513	995	99.75%/99.50%
CAMRa2011	610	7710	290	93.51%/97.49%
Yelp	25667	25815	-	-/99.89%
MovieLens1M	6040	3900	-	-/95.75%
MovieLens25M	162541	59047	-	-/99.73%
MovieLens-latest-small	602	9724	-	-/98.30%

A. Datasets

Our experiments are conducted on six real-world datasets: Mafengwo,[2] Yelp,[3] CAMRa2011,[4] and MovieLens.[5] Table II lists the statistical information about these original datasets, where the last column shows the sparsity information of user–item (U-I) interactions or group–item (G-I) interactions corresponding to different datasets. The Mafengwo dataset [25] is crawled from a tourism website Mafengwo.[6] It contains a member–group relationship file and the traveling venues of the group and each of the group members. The member–group relationship file records the member information for each group. CAMRa2011 is a dataset from the second challenge on context-aware movie recommendation, which contains movie rating records of both individual users and households as well as group member information. These two datasets have been tested for group recommendation tasks [7], [14], [25]. Yelp and MovieLens are another two popular datasets which have been widely tested for individual recommendations because they only contain individual user–item interactions. However, all the above datasets have rarely been tested for cross-domain group recommendation. In this study, we preprocess the above datasets and construct three cross-domain recommendation tasks for both individuals and groups. For Mafengwo which is the target domain for task 1, we choose Yelp as a relative dense source domain to assist the recommendation in Mafengwo because restaurant and tourism are closely related in item level. Since our proposed method does not need group labels for the source domain data, we do not process the Yelp data to generate group labels, but we filter it such that it is comparable with the size of user data in Mafengwo. Similarly, for CAMRa2011 which is the target domain for task 2, we filter a comparable source dataset from MovieLens1M to assist the recommendation in CAMRa2011. Besides, we extract another target domain dataset for task 3, named MovieLens-Simi, from the MovieLens-latest-small dataset. Since complex inner relationships are more common in real groups with common tastes, such as groups of friends, than in random groups, the groups in MovieLens-Simi are formed with high inner group similarity, which is measured by user-to-user similarity and computed using Pearson correlation coefficient (PCC). Then, similar to [16], [26], and [27], we generate synthetic groups as follows: we first add each user in MovieLens-latest-small as the first member in a group. Then, we choose

[2]https://github.com/caoda0721/SoAGREE
[3]https://www.yelp.com/dataset
[4]https://recsys.acm.org/recsys11/camra/
[5]http://grouplens.org/datasets/MovieLens/
[6]http://www.mafengwo.cn

TABLE III
DESCRIPTION OF DATA SUBSETS FOR THREE TASKS

Task	Data_name	Data_source	Domain	U/G No.	Item No.	Sparsity
Task 1	task1_s1	Yelp	source	4948	1500	99.37%
	task1_t1	MaFengwo	target user	5275	1513	99.50%
	task1_t2	MaFengwo	target group	995	1513	99.76%
Task 2	task2_s1	MovieLens1M	source	602	8000	95.68%
	task2_t1	CAMRa2011	target user	602	7710	97.49%
	task2_t2	CAMRa2011	target group	290	7710	93.51%
Task 3	task3_s1	MovieLens25M	source	610	9724	96.94%
	task3_t1	MovieLens-Simi	target user	610	9724	98.30%
	task3_t2	MovieLens-Simi	target group	610	9724	99.46%

five similar users with the highest PCC similarity with the first member as the other group members. With the group generated, we count the items rated by all the members in each group to generate group–item interactions. Given a group, if more than half of the group members like a movie, we assume that the movie is adopted by the group. Finally, the MovieLens-Simi dataset is constructed as a target domain for task 3 for which we choose MovieLens25M as its source domain. Following AGREE [25], we also filter out the users in each target domain who do not participate in any group and only retain the user–item interactions with group information. After preprocessing, three cross-domain recommendation tasks are designed for our experiments as shown in Table III.

1) *Task 1:* Yelp (restaurant) → Mafengwo (tourism).
2) *Task 2:* MovieLens1M (movie)→ CAMRa2011 (movie).
3) *Task 3:* MovieLens25M (movie) → MovieLens-Simi (movie).

B. Baselines and Experimental Settings

1) Different Types of Baselines: To verify the effectiveness and superiority of HAN-CDGR, we used the following baselines.

1) **BPR** [48] uses pairwise ranking to optimize implicit matrix factorization. This method is used on individual recommendation and group recommendation where one group is treated as one virtual user in the target domain.
2) **NCF** [49] is a matrix factorization method that uses MLP to model the nonlinear relationship between users and items. Similar to BPR, it is used for recommendations for both individual users and groups.
3) **LightGCN** [51] is a simplified graph-convolution-network-based individual recommendation method. It learns user and item factors through neighborhood aggregation. Similar to both the BPR and NCF methods, the LightGCN method also generates group recommendations by treating groups as virtual users.
4) **BPR-AVG**, **NCF-AVG**, and **LightGCN-AVG** (short for "Light-AVG") are three individual recommendation methods that combine BPR/NCF/LightGCN with the predefined average strategy. They consider the average embedding of all the members in the group and optimize the BPR/NCF/LightGCN objective to make group recommendations.
5) **AGREE** [25] is a popular group recommendation method which uses a single-attention neural network to learn feature vectors for groups and uses the NCF framework to jointly model user–/group–item interactions. Since AGREE has confirmed its superiority than the COM [26] method, which also experimentally outperformed the PIT [21] method, we did not compare our proposal with these two methods again.
6) **GRADI** [5] is a single-attention neural-network-based group recommendation method. This method explores dual influence between groups and group members when modeling groups' preferences.
7) **GAME** [6]: is a graph representation learning-based group recommendation method. It models the inherent representations of groups, users, or items from multiple independent views. Recommendation results for both individual users and groups are generated based on the NCF framework.
8) **NATR** [42] is a cross-domain recommendation method with item overlapping that transfers the overlapped item embeddings across domains. In our group recommendation task, the user–item and group–item interactions can be regarded as the source and target domains with shared item sets under the same scenario with NATR where the groups are treated as virtual users.

To verify the effectiveness of the hierarchical neural network, we design a variant of HAN-CDGR, named **HAN-GR** which removes the adversarial learning module in HAN-CDGR but retains the hierarchical attention mechanism for user/group preference modeling.

2) Parameter Settings: We implemented our proposed method and all the comparative methods using PyTorch. For all the baselines, we used the original code if available or implement them by PyTorch to get experimental results. The results were fine-tuned according to the parameter settings in the original article. For the network structure of our methods HAN-GR and HAN-CDGR, we set the negative sampling ratio as 4. We searched the results for the embedding size, dropout ratio, batch size, and learning rate in [16, 32, 64, 128, 256], [0.0, 0.2, 0.4, 0.6, 0.8], [128, 256, 512], and [2e−3, 2e−4, 2e−5, 5e−6], respectively. As HAN-CDGR involves additional parameters β and γ to balance user prediction and domain losses. We searched parameters β and γ in the range of [0.05, 0.1, 0.2, 0.5, 1] and [0, 5e−4, 2e−4, $3e-1$], respectively. Moreover, we empirically set the activation function as ReLU and used the adaptive moment estimation (Adam) optimizer for network learning. Early stop technique was used to stop the training processes. We repeated the experiments on each setting five times and reported the best average results.

We chose *leave-one-out* as our evaluation protocol, which has been widely used in top-N recommendation [7], [25], [48]. Specifically, for each user (or group) that had more than three histories, we removed the latest entry for testing and took the remaining entries for training. Hit ratio (HR) and normalized discounted cumulative gain (NDCG) were used as evaluation metrics for implicit feedback. During evaluation, we randomly sampled 100 items that have not been interacted by the test user (or group) and ranked the test item among 100 negative items, which is a common strategy to balance prediction accuracy and computation cost [49], [52], [53].

C. Experimental Results

The experimental results on three tasks are given in Tables IV–VI, and the training time of each method for an epoch on the MaFengWo dataset is shown in Table VII. The performances on HR and NDCG are provided for both the users and groups. Furthermore, we conducted the Friedman test based on the results. We can see that most of the p-values of the t-tests on HR@5 and HR@10 for both the user and group recommendation results are less than 0.05, which confirms the significance of the improvement of our HAN-CDGR method over all other comparative methods. In addition, we show the best performance results in bold. Overall, the HAN-CDGR method delivers the best performance of most baselines in all the three tasks. This verifies the conclusion that modeling user/group preferences with a hierarchical attention neural network is more effective than a single-attention or a fixed-aggregation strategy. Meanwhile, transferring knowledge from a source domain to the target domain can significantly improve the recommendation performance of both individual users and groups. We can make the following observations.

1) Overall Performance Comparison on Individual Recommendation: BPR, NCF, and LightGCN are initially designed to generate recommendations for individual users. Therefore, the user– and group–item interactions are trained separately and cannot reinforce each other when training. Our proposed HAN-CDGR and its variant HAN-GR significantly outperformed these three baselines which indicates that the preference of each member is closely related and can be mutually influenced by group preference. Specifically, in the MaFengWo dataset, HAN-CDGR achieves a 0.99%–8.76% improvements in individual user recommendation tasks and gains a 15.07%–31.20% improvement in group recommendation tasks over NCF on HR or NDCG metrics.

2) Overall Performance Comparison on Group Recommendation: The BPR-AVG, NCF-AVG, and Light-AVG methods are three single-domain learning methods used for group recommendation that consider the average embedding of all the users in the group as the group representation. Light-AVG has a more superior performance than BPR-AVG and NCF-AVG on group recommendation because it modeled both group and item latent factors from graph data. The AGREE, GRADI, and GAME methods show improved performances compared with BPR-AVG, NCF-AVG, and Light-AVG, since they used attention mechanisms to learn group members' dynamic weights and trained models from both user– and group–item interaction data. HAN-CDGR outperforms these aforementioned methods for two reasons: 1) group preference modeling considering the complex inner relationships among group members and 2) user preference modeling with containment relationship between the users and diverse groups. HAN-CDGR outperforms the single-attention-based baselines, which indicates that the above two kinds of relationships are important and should be considered in group/user preference modeling. Concretely, in the Mafengwo dataset, each user has participated in at least one group and up to 190 groups, and each user in MovieLens-Simi has participated in at least one group and up to 17 groups, while each user in CAMRa2011 has only participated in one group. The largest improvement of the three datasets is in the Mafengwo dataset, which indicates that our proposed method has superior performance when one user has participated in many groups.

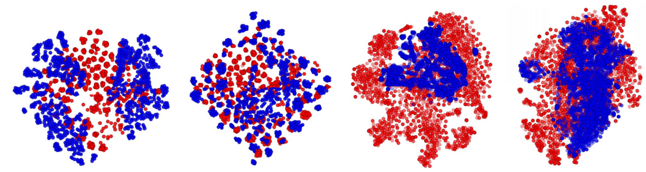


Fig. 3. Visualization on the positive and negative user latent feature distributions before and after adversarial learning (best viewed in color). The figures show the difference in (a) and (b) positive user and (c) and (d) negative user latent features. Source domain data are shown in blue and target domain is in red.

3) Effectiveness of Knowledge Transfer: NATR is an individual cross-domain recommendation method with item overlapping. It transfers knowledge from item-side information to assist group recommendation. It improves around 30% on group recommendation than the other nontransfer group recommendation methods involving BPR-AVG, NCF-AVG, and Light-AVG, which proves the effectiveness of knowledge transfer across group– and group member–item interactions. Moreover, our proposed method HAN-CDGR achieves around 5% improvement than HAN-GR and NATR, which means that the knowledge transfer from another user domain is also effective and improves the recommendation in the target domain, especially for the group recommendation task.

To ensure that the trained latent user features from the source and target domains are similar and properly aligned by adversarial learning, we use stochastic neighbor embedding (t-SNE) [54] to visualize both the positive and negative user latent features. Fig. 3(a) shows the positive user latent features of the source domain (blue) and target domain (red) that are trained separately, i.e., the source domain is trained by the NCF method and the target domain is trained by the HAN-GR method. Fig. 3(b) shows the positive user latent features after training 200 epochs in the proposed HAN-CDGR. Similarly, Fig. 3(c) and (d) shows the negative user feature distributions before and after alignment. Fig. 3 suggests that the user latent features are aligned after adversarial learning, and the divergence of the distributions is decreased after alignment. Therefore, we can come to the conclusion that our proposed HAN-CDGR is able to effectively match the latent features of users between the source and target domains.

D. Ablation Analysis

In this section, we present an ablation study to validate the functions of each module in HAN-CDGR.

We first examine the hierarchical attention network and adversarial module and construct four variants of HAN-CDGR: 1) **H-AVG-GR**; 2) **H-LM-GR**; 3) **H-MS-GR**; and 4) **H-EXP-GR** are the four simplified versions of our HAN-CDGR method. They do not transfer data from the source domain to the target domain and change the second-level attention (vanilla attention) to a fixed aggregation strategy, including average, LM, MS, and expertise strategies, respectively.

As depicted in Table VIII, we conducted studies on three datasets and only report group recommendation performances

TABLE IV

OVERALL COMPARISON RESULTS ON THE MAFENGWO DATASET

Method	User						Group					
	HR@5	NDCG@5	T-test	HR@10	NDCG@10	T-test	HR@5	NDCG@5	T-test	HR@10	NDCG@10	T-test
BPR	0.8203	0.5410	0.0061	0.9199	0.5741	0.0198	0.4647	0.3064	1.82e-5	0.6219	0.4574	1.62e-5
NCF	0.8851	0.6424	0.0003	0.9501	0.6639	0.0352	0.6567	0.5210	0.0122	0.7525	0.5522	0.0314
LightGCN	0.8159	0.6350	4.1e-7	0.8864	0.6580	0.0014	0.6247	0.4515	4.45e-6	0.7577	0.4952	7.3e-6
BPR-AVG	-	-	-	-	-	-	0.4460	0.3174	1.22e-8	0.5612	0.3544	5.59e-7
NCF-AVG	-	-	-	-	-	-	0.6090	0.4479	0.0114	0.7248	0.4858	0.0180
Light-AVG	-	-	-	-	-	-	0.6965	0.5039	6.42e-6	0.8096	0.5406	0.0001
NATR	-	-	-	-	-	-	0.7688	0.6670	0.048	0.7954	0.6758	0.2190
AGREE	0.8981	0.6416	0.0037	0.9427	0.6563	0.0093	0.7445	0.6287	0.0004	0.7941	0.6450	0.0077
GRADI	0.8974	0.6418	0.0260	0.9464	0.6581	0.0680	0.7261	0.5880	8.84e-5	0.7789	0.6053	0.0030
GAME	0.8855	0.5975	0.0003	0.9443	0.6169	4.31e-5	0.7317	0.6064	0.0009	0.8070	0.6310	1.65e-5
HAN-GR	0.9115	0.6884	-	0.9558	0.7133	-	0.8034	0.6295	-	**0.8690**	0.6510	-
HAN-CDGR	**0.9207**	**0.6987**	**-**	**0.9596**	**0.7173**	**-**	**0.8137**	**0.6836**	-	0.8659	**0.7007**	**-**

TABLE V

OVERALL COMPARISON RESULTS ON THE CAMRA2011 DATASET

Method	User						Group					
	HR@5	NDCG@5	T-test	HR@10	NDCG@10	T-test	HR@5	NDCG@5	T-test	HR@10	NDCG@10	T-test
BPR	0.6258	0.4235	9.75e-5	0.7912	0.4773	0.0007	0.5859	0.3993	0.0003	0.7772	0.4616	0.0045
NCF	0.6252	0.4222	5.62e-5	0.7922	0.4765	6.65e-5	0.5840	0.3975	0.0003	0.7783	0.4607	0.0017
LightGCN	0.6242	0.4217	2.53e-5	0.7885	0.4756	0.0002	0.5862	0.4016	0.0005	0.7759	0.4637	0.0082
BPR-AVG	-	-	-	-	-	-	0.5853	0.3981	0.0005	0.7789	0.4610	0.0026
NCF-AVG	-	-	-	-	-		0.5839	0.3972	0.0004	0.7769	0.4601	0.0215
Light-AVG	-	-	-	-	-		0.5882	0.4037	0.0005	0.7771	0.4650	0.0270
NATR							0.6175	0.4078	0.0491	0.8017	0.4676	0.4776
AGREE	0.6262	0.4216	3.01e-5	0.7993	0.4777	0.0006	0.5878	0.4013	0.0009	0.7818	0.4641	0.0105
GRADI	0.6270	0.4214	3.76e-5	0.8011	0.4779	0.0002	0.5865	0.3992	0.0006	0.7829	0.4628	0.0040
GAME	0.6272	0.4110	3.63e-5	0.7934	0.4753	6.85e-5	0.5855	0.3987	0.0004	0.7724	0.4596	0.0021
HAN-GR	**0.6916**	0.4326	-	0.8534	0.4985	-	0.6236	0.4026	-	**0.8121**	**0.4755**	**-**
HAN-CDGR	0.6829	**0.4471**	**-**	**0.8567**	**0.5069**	**-**	**0.6468**	**0.4113**	-	0.8068	0.4729	-

TABLE VI

OVERALL COMPARISON RESULTS ON THE MOVIELENS-SIMI DATASET

Method	User						Group					
	HR@5	NDCG@5	T-test	HR@10	NDCG@10	T-test	HR@5	NDCG@5	T-test	HR@10	NDCG@10	T-test
BPR	0.5403	0.3850	0.0055	0.7036	0.4378	0.0040	0.8447	0.5739	0.0023	0.9641	0.6130	0.0007
NCF	0.6236	0.4121	0.0201	0.7823	0.4638	0.0111	0.9395	0.7380	0.0018	0.9831	0.7525	0.0098
LightGCN	0.5216	0.3779	2.42e-5	0.6803	0.4280	1.70e-5	0.8658	0.6837	3.81e-8	0.9549	0.7115	1.28e-6
BPR-AVG	-	-	-	-	-	-	0.8965	0.6763	0.0221	0.9697	0.7005	0.0120
NCF-AVG	-	-	-	-	-	-	0.9570	0.7682	7.29e-5	0.9859	0.7778	0.0561
Light-AVG	-	-	-	-	-	-	0.8690	0.6837	2.97e-7	0.9602	0.7137	2.13e-6
NATR							0.9186	0.7423	5.77e-6	0.9750	0.7602	0.0003
AGREE	0.6168	0.4179	0.0141	0.7655	0.4659	0.0239	0.9723	0.7465	0.0022	**0.9977**	0.7551	0.0004
GRADI	0.6196	0.4192	0.0269	0.7787	0.4705	0.0235	0.9672	0.7715	0.0241	0.9933	0.7802	0.0783
GAME	0.5485	0.3933	5.32e-6	0.6728	0.4335	1.51e-5	0.9599	0.8069	7.59e-6	0.9876	0.8161	6.59e-6
HAN-GR	0.6456	0.4489	-	0.7875	0.4973	-	0.9648	0.7717	-	0.9951	0.7819	-
HAN-CDGR	**0.6544**	**0.4746**	**-**	**0.8013**	**0.5124**	**-**	**0.9817**	**0.8173**	-	0.9974	**0.8234**	**-**

TABLE VII

TRAINING TIME ON THE MAFENGWO DATASET

Method	Time		Method	Time	
	user	group		user	group
BPR	2.8431	0.1890	NATR	-	1.3716
NCF	2.9357	0.1991	AGREE	2.3536	0.8216
LightGCN	3.2673	0.2167	GRADI	1.8526	0.4452
BPR-AVG	-	0.5068	GAME	39.09	1.3737
NCF-AVG	-	0.5196	HAN-GR	78.45	1.2657
Light-AVG	-	0.6218	HAN-CDGR	86.20	1.889

on both HR and NDCG in Table VIII. Our proposed method HAN-CDGR shows a better performance in group recommendation than most fixed-aggregation-strategy-based variants. This validates the effectiveness of the hierarchical attention neural network and adversarial learning module in modeling group preferences. Since each user in the CAMRa2011 dataset has taken part in only one group, the user preference modeling reduced to a fixed aggregation-based strategy which may degrade the performance on group recommendation. Besides, there is no one fixed strategy among the four types that is absolutely dominant in all the datasets. This also confirms that a dynamic weight learning strategy is more suitable than predefined, static aggregation strategies. To illustrate hierarchical attention, we randomly select ten groups and visualize the self-attention weights within one group and vanilla attention weights for ten groups in Fig. 4(a) and (b), respectively.

As described in M1 and M2 of HAN-CDGR, both the group preference and user preference are modeled by combining two components, where the group preference modeling is composed of M1-a) group member embedding aggregation and M1-b) group inherent embedding, and user preference modeling is composed of M2-c) group-specific user embedding aggregation and M2-d) user inherent embedding. To further investigate the effectiveness of each module in HAN-CDGR, we compare HAN-CDGR with its five other variants as follows.

1) **HAN-CDGR** (M1-a and M2-c) only uses M1-a for group preference modeling and M2-c for user preference

TABLE VIII
HAN-CDGR Ablation Study on the Second-Level Attention

Method	Mafengwo				CAMRa2011				MovieLens-simi			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
H-AVG-GR	0.7787	0.6059	0.8360	0.6366	0.6326	0.4060	0.8190	0.4702	0.9528	0.6921	0.9873	0.7237
H-LM-GR	0.7779	0.6700	0.8239	0.6828	0.5970	0.3844	0.7752	0.4565	0.9503	0.7160	0.9845	0.7386
H-MS-GR	0.7863	0.6829	0.8323	0.6976	0.6309	0.4038	0.8130	0.4768	0.9520	0.6715	0.9859	0.7094
H-EXP-GR	0.7790	0.6838	0.8239	0.6739	0.6269	0.3931	0.8199	0.4699	0.9490	0.6868	0.9855	0.7223
HAN-CDGR	**0.8137**	0.6836	**0.8659**	**0.7007**	**0.6468**	**0.4113**	0.8068	0.4729	**0.9817**	**0.8173**	**0.9974**	**0.8234**

TABLE IX
HAN-CDGR Ablation Study on Each Component of User/Group Preference Modeling

Method	User		Group	
	HR@5	NDCG@5	HR@5	NDCG@5
HAN-CDGR(M1-a & M2-c)	0.9142	0.6882	0.8068	0.6984
HAN-CDGR(M1-a & M2-d)	0.9139	0.6796	0.7210	0.5578
HAN-CDGR(M1-b & M2-c)	0.9127	0.6779	0.8020	0.6391
HAN-CDGR(M1-b & M2-d)	0.9115	0.6731	0.6858	0.5052
HAN-CDGR(Without M2)	-	-	0.7819	0.6717
HAN-CDGR	**0.9207**	**0.6987**	**0.8137**	0.6836

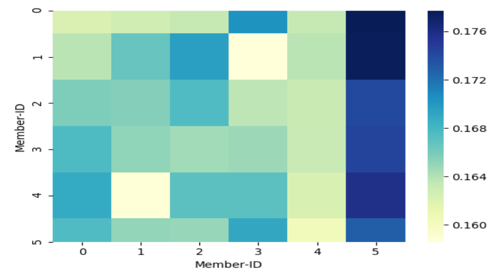
modeling, which are both learned using the hierarchical attention network in HAN-CDGR.

2) **HAN-CDGR** (M1-a and M2-d) uses M1-a for group preference modeling and only considers the user inherent embedding (M2-d) as the users' latent feature.
3) **HAN-CDGR** (M1-b and M2-c) regards the group's inherent preference as its overall preference and uses M2-c for user preference modeling.
4) **HAN-CDGR** (M1-b and M2-d) does not use the hierarchical neural network to model either group preference or user preference. It only considers the inherent preferences of users or groups as their final latent feature. It can be regarded as an NCF-based cross-domain group recommendation framework.
5) **HAN-CDGR** (without M2) deletes the user preference modeling module in HAN-CDGR and only generates recommendations for groups using a hierarchical neutral network.

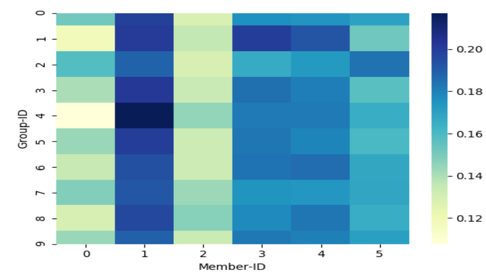
We conducted ablation studies on component analysis of user/group preference modeling to report HR@5 and NDCG@5 on both the users and groups on the MaFengWo dataset in Table IX. We can observe that HAN-CDGR (M1-a and M2-c) performs better than other four variants, indicating M1-a and M2-c play essential roles in group/user preference modeling. HAN-CDGR (without M2) is stronger than HAN-CDGR (M1-a and M2-d) and HAN-CDGR (M1-b and M2-d), but inferior to the other variants highlighting the benefits of user preference modeling. In addition, the ignorance of training data from both user–item and group-item interactions in HAN-CDGR (without M2) may also impair model accuracy. Overall, HAN-CDGR outperforms most of its variants, validating the benefits of our hierarchical neural network and M3 in modeling both user and group preferences and facilitating recommendations for both individual users and groups in the target domain. Moreover, modeling user–item and group–item interactions simultaneously can reinforce recommendation tasks for both the users and groups.

E. Parameter Analysis

There are two important parameters in our proposed method: the dropout ratio p and embedding size d. We analyze



(a)



(b)

Fig. 4. Visualization of (a) self-attention weights within one group and (b) vanilla attention weights for ten sampled groups. Both the x-axis and y-axis in (a) represent the member-ID in a group, while the x-axis represents the member-ID and y-axis represents the group-ID in (b). Note that all the original user-IDs and group-IDs have been reidentified, and a darker color indicates a larger weight.

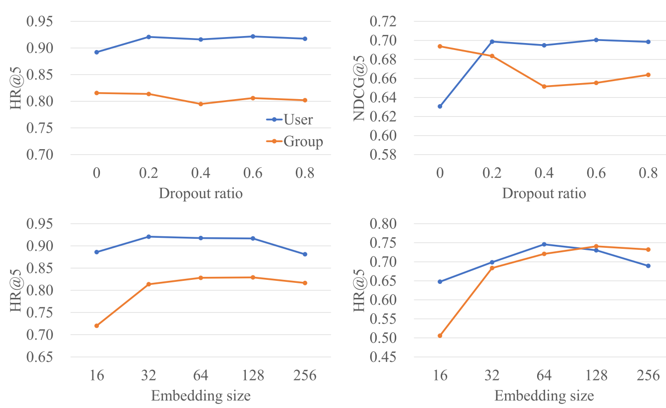


Fig. 5. Parameter analysis on drop ratio and embedding size on the Mafengwo dataset.

the parameters p and d in the range of [0.0, 0.2, 0.4, 0.6, 0.8] and [16, 32, 64, 128, 256], respectively. Fig. 5 shows the parameter analysis results on HR@5 and NDCG@5 on the Mafengwo dataset. The results show that using a dropout ratio $p \approx 0.2$ achieves an optimal recommendation accuracy on HR@5 for both individual users and groups, while group recommendation performance on NDCG@5 achieves best on $p \approx 0$. For the performance analysis on d, the influence of d on the accuracy of HAN-CDGR shows an upward and then downward trend.

V. Conclusion and Future Study

In this article, we propose a novel hierarchical attention neural-network-based cross-domain group recommendation method, HAN-CDGR. Extensive experiments on real-world datasets show that HAN-CDGR and its variant HAN-GR outperform the state-of-the-art baselines. This shows that HAN-CDGR is able to model user and group adaptive preferences from data and significantly improve both user and group recommendation performance in the target domain through transferring knowledge from a source domain. The experiment results also suggest that the inner relationships of group members and containment relationships between users and groups are important in both user and group preference modeling.

The advantages of this work are twofold. First, our proposed method constructs a hierarchical attention neural network to model both user and group's latent preferences. On the user side, we leverage the hierarchical structure of users, groups, and items to capture the containment relationship between users and groups, revealing additional insights into user preferences. On the group side, we aggregate group members' preferences as a part of group representation with consideration of both the dynamic weights of different group members and the inner relationships among them. Second, our proposed method is able to effectively transfer knowledge from a similar source domain to facilitate recommendation for both individual users and groups in the target domain through adversarial learning. However, the main drawback of this work is its computational complexity. Our method may be potentially sped up by distributed learning and a big data framework such as the consensus learning algorithm.

For future work, we aim to improve our proposed method in two aspects. First, we aim to propose novel hierarchical graph attention networks to improve user/group preference and item property modeling by incorporating multimodule features such as images and texts. In addition, the relationships between users and groups are sometimes uncertain, so we will also investigate how to deal with the uncertainties in the group recommendation [55]. Second, modeling static (long-term) and dynamic (short-term) behaviors and trends of users/groups and items to enhance group recommendation performance provides a very significant future research direction for us, and hence, we can explore how to improve our HAN-CDGR method to enable it to learn temporal latent features.

References

[1] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015.

[2] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decis. Support Syst.*, vol. 87, pp. 80–93, Jul. 2016.

[3] Y. Du, X. Meng, Y. Zhang, and P. Lv, "GERF: A group event recommendation framework based on learning-to-rank," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 674–687, Apr. 2020.

[4] X. Hu, X. Meng, and L. Wang, "SVD-based group recommendation approaches: An experimental study of moviepilot," in *Proc. 2nd Challenge Context-Aware Movie Recommendation (CAMRa)*, 2011, pp. 23–28.

[5] Z. He, C.-Y. Chow, J.-D. Zhang, and N. Li, "GRADI: Towards group recommendation using attentive dual top-down and bottom-up influences," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 631–636.

[6] Z. He, C.-Y. Chow, and J.-D. Zhang, "GAME: Learning graphical and attentive multi-view embeddings for occasional group recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 649–658.

[7] D. Cao, X. He, L. Miao, G. Xiao, H. Chen, and J. Xu, "Social-enhanced attentive group recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1195–1209, Mar. 2019.

[8] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 566–577.

[9] Z. Huang *et al.*, "Social group recommendation with TrAdaBoost," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 5, pp. 1278–1287, Oct. 2020.

[10] H. Yin, Q. Wang, K. Zheng, Z. Li, and X. Zhou, "Overcoming data sparsity in group recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 1–12, Jul. 2020.

[11] Q. Zhang, J. Lu, D. Wu, and G. Zhang, "A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1998–2012, Jul. 2018.

[12] N. A. Christakis and J. H. Fowler, "Social contagion theory: Examining dynamic social networks and human behavior," *Statist. Med.*, vol. 32, no. 4, pp. 556–577, Feb. 2013.

[13] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo, "Modelling hierarchical relationships in group recommender systems," in *Proc. Int. Conf. Case-Based Reasoning*. Cham, Switzerland: Springer, 2015, pp. 320–335.

[14] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and W. Cao, "Deep modeling of group preferences for group-based recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 28, no. 1, 2014, pp. 1861–1867.

[15] J. Lu, Q. Zhang, and G. Zhang, *Recommender Systems: Advanced Developments*. Singapore: World Scientific, 2020.

[16] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 119–126.

[17] S. Berkovsky and J. Freyne, "Group-based recipe recommendations: Analysis of data aggregation strategies," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 111–118.

[18] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu, "Group recommendation: Semantics and efficiency," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 754–765, 2009.

[19] E. Quintarelli, E. Rabosio, and L. Tanca, "Recommending new items to ephemeral groups using contextual user influence," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 285–292.

[20] S. Seko, T. Yagi, M. Motegi, and S. Muto, "Group recommendation using feature space representing behavioral tendency and power balance among members," in *Proc. 5th ACM Conf. Recommender Syst. (RecSys)*, 2011, pp. 101–108.

[21] X. Liu, Y. Tian, M. Ye, and W.-C. Lee, "Exploring personal impact for group recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2012, pp. 674–683.

[22] Y. Du, X. Meng, and Y. Zhang, "CVTM: A content-venue-aware topic model for group event recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 7, pp. 1290–1303, Jul. 2020.

[23] J. Han *et al.*, "Adaptive deep modeling of users and items using side information for recommendation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 737–748, Jun. 2019.

[24] U. S. Shanthamallu, J. J. Thiagarajan, H. Song, and A. Spanias, "GrAMME: Semisupervised learning using multilayered graph attention models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 3977–3988, Oct. 2019.

[25] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 645–654.

[26] Q. Yuan, G. Cong, and C.-Y. Lin, "COM: A generative model for group recommendation," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. discovery data mining*, Aug. 2014, pp. 163–172.

[27] L. Vinh Tran, T.-A. N. Pham, Y. Tay, Y. Liu, G. Cong, and X. Li, "Interact and decide: Medley of sub-attention networks for effective group recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 255–264.

[28] L. Guo, H. Yin, Q. Wang, B. Cui, Z. Huang, and L. Cui, "Group recommendation with latent voting mechanism," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Apr. 2020, pp. 121–132.

[29] A. Sankar, Y. Wu, Y. Wu, W. Zhang, H. Yang, and H. Sundaram, "GroupIM: A mutual information maximization framework for neural group recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1279–1288.

[30] L. Guo, H. Yin, T. Chen, X. Zhang, and K. Zheng, "Hierarchical hyperedge embedding-based representation learning for group recommendation," *ACM Trans. Inf. Syst.*, vol. 40, no. 1, pp. 1–27, Jan. 2021.

[31] X. Wang, Y. Liu, J. Lu, F. Xiong, and G. Zhang, "TruGRC: Trust-aware group recommendation with virtual coordinators," *Future Gener. Comput. Syst.*, vol. 94, pp. 224–236, May 2019.

[32] P. Yildirim and D. Birant, "K-linkage: A new agglomerative approach for hierarchical clustering," *Adv. Electr. Comput. Eng.*, vol. 17, no. 4, pp. 77–88, 2017.

[33] J.-Y. Park and J.-H. Kim, "Incremental class learning for hierarchical classification," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 178–189, Jan. 2018.

[34] L. Quijano-Sanchez, J. A. Recio-Garcia, and B. Diaz-Agudo, "HappyMovie: A Facebook application for recommending movies to groups," in *Proc. IEEE 23rd Int. Conf. Tools Artif. Intell.*, Nov. 2011, pp. 239–244.

[35] M. Zhong *et al.*, "HIGnet: Hierarchical and interactive gate networks for item recommendation," *IEEE Intell. Syst.*, vol. 35, no. 5, pp. 50–61, Sep. 2020.

[36] D. Qin, X. Zhou, L. Chen, G. Huang, and Y. Zhang, "Dynamic connection-based social group recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 453–467, Mar. 2020.

[37] E. Yalcin, F. Ismailoglu, and A. Bilge, "An entropy empowered hybridized aggregation technique for group recommender systems," *Exp. Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114111.

[38] C. Wang, M. Niepert, and H. Li, "RecSys-DAN: Discriminative adversarial networks for cross-domain recommender systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2731–2740, Aug. 2019.

[39] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng, "DTCDR: A framework for dual-target cross-domain recommendation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1533–1542.

[40] P. Li and A. Tuzhilin, "DDTCDR: Deep dual transfer cross domain recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 331–339.

[41] G. Hu, Y. Zhang, and Q. Yang, "CoNet: Collaborative cross networks for cross-domain recommendation," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 667–676.

[42] C. Gao *et al.*, "Cross-domain recommendation without sharing user-relevant data," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 491–502.

[43] Q. Zhang, W. Liao, G. Zhang, B. Yuan, and J. Lu, "A deep dual adversarial network for cross-domain recommendation," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 7, 2021, doi: 10.1109/TKDE.2021.3132953.

[44] F. Yuan, L. Yao, and B. Benatallah, "DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4227–4233.

[45] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Feb. 2019.

[46] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[47] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[48] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[49] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 173–182.

[50] C. Li, X. Yu, T. Huang, and X. He, "Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2407–2418, Jun. 2018.

[51] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 639–648.

[52] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 278–288.

[53] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. discovery data mining (KDD)*, 2008, pp. 426–434.

[54] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.

[55] J. Lu, H. Zuo, and G. Zhang, "Fuzzy multiple-source transfer learning," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 12, pp. 3418–3431, Dec. 2020.

Ruxia Liang received the Ph.D. degree in management science and engineering from the Business School, Central South University, Changsha, China, in 2021.

From 2019 to 2020, she was a Visiting Student with the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Faculty of Engineering and Information Technology, Australian Artificial Intelligence Institute (AAII), University of Technology Sydney, Ultimo, NSW, Australia. She is currently a Research Assistant Professor with the National Engineering Research Center for Educational Big Data and the Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, China. Her current research interests include group recommender systems and transfer learning techniques, especially in cross-domain group recommender systems.

Qian Zhang (Member, IEEE) received the Ph.D. degree from the University of Technology Sydney, Ultimo, NSW, Australia, in 2018.

She is currently a Post-Doctoral Research Fellow and a member of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Faculty of Engineering and Information Technology, Australian Artificial Intelligence Institute (AAII), University of Technology Sydney. Her research interests include recommender systems and personalized techniques. She specializes in cross-domain recommender systems.

Jianqiang Wang received the Ph.D. degree in management science and engineering from the Business School, Central South University, Changsha, China, in 2005.

He is currently a Professor with the Department of Management Science and Information Management, Business School, Central South University, Changsha, China. He is also a Ph.D. Supervisor in management science and engineering. Over the past couple decades, his research interests include decision-making theory. His current research interests include decision-making theory and application, risk management and control, and information management.

Jie Lu (Fellow, IEEE) received the Ph.D. degree from the Curtin University of Technology, Perth, WA, Australia, in 2000.

She is currently a Distinguished Professor and the Director of the Australian Artificial Intelligence Institute (AAII), University of Technology Sydney, Ultimo, NSW, Australia. She has authored six research books and over 500 papers in refereed journals and conference proceedings. She has received over 20 ARC Laureate, ARC Discovery and Linkage Projects, and government and industry projects. Her main research interests include fuzzy transfer learning, concept drift, decision support systems, and recommender systems.

Dr. Lu is an IFSA Fellow and an Australian Laureate Fellow. She has delivered over 35 keynote speeches at international conferences and chaired 20 international conferences. She has received various awards nationally and internationally. She serves as the Editor-in-Chief for *Knowledge-Based Systems* (Elsevier) and the *International Journal of Computational Intelligence Systems*.