2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00036

Hypergraph Convolutional Network for Group Recommendation

Renqi Jia[†‡], Xiaofei Zhou[†‡*], Linhua Dong[†‡] and Shirui Pan[§]
[†] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[‡] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[§] Faculty of Information Technology, Monash University, Melbourne, Australia
[†‡]jiarenqi@iie.ac.cn, [†‡]zhouxiaofei@iie.ac.cn, [†‡]donglinhua@iie.ac.cn, [§]shirui.pan@monash.edu

***Abstract*—Group activities have become an essential part of people's daily life, which stimulates the requirement for intensive research on the group recommendation task, i.e., recommending items to a group of users. Most existing works focus on aggregating users' interests within the group to learn group preference. These methods are faced with two problems. First, these methods only model the user preference inside a single group while ignoring the collaborative relations among users and items across different groups. Second, they assume that group preference is an aggregation of user interests, and factually a group may pursue some targets not derived from users' interests. Thus they are insufficient to model the general group preferences which are independent of existing user interests. To address the above issues, we propose a novel dual channel Hypergraph Convolutional network for group Recommendation (HCR), which consists of member-level preference network and group-level preference network. In the member-level preference network, in order to capture cross-group collaborative connections among users and items, we devise a member-level hypergraph convolutional network to learn group members' personal preferences. In the group-level preference network, the group's general preference is captured by a group-level graph convolutional network based on group similarity. We evaluate our model on two real-world datasets and the experimental results show that the proposed model significantly and consistently outperforms state-of-the-art group recommendation techniques.**

***Index Terms*—Group Recommendation, Hypergraph Convolution, Graph Convolution, Representation Learning**

I. Introduction

With the rapid growth of Internet services and mobile device usages, users are inundated with tremendous choices and options. Recommender systems have played a significant role in alleviating information overload problems with the development of artificial intelligence [1]–[3]. Recently, many works focus on a more complicated scenario where items are recommended to a group of users, denoted as group recommendation [4]–[6]. Different from the decision of an individual user, the decision-making process of a group is more complex, as different members in the group have diverse interests and contribute unequally to the final decision.

The problem of preference assessment and preference aggregation of group members has attracted researchers from academics and industry. The earlier predefined methods emphasize the process of preference aggregation. In these methods, the group interest is modeled by fixed strategies such as average [7], least misery [8], maximum satisfaction [9], and so on. However, these predifined strategies are data-independent, short of the ability to model the preferences of group members and adjust their weights dynamically. This ability is significant for a group to make decisions on different items. Therefore, these predefined strategies are insufficient to capture the diverse member intention and the complex decision-making process, which leads to the suboptimal performance of the group recommendation.

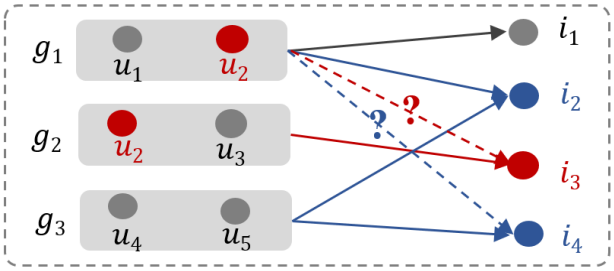


Fig. 1. The example of the group-item interaction data. The red and blue parts show the group's collaborative preferences from the perspective of user and item, respectively. The question marks indicate the cross-group collaborative signals.

The model-based approaches focus on the group preference aggregation process by capturing the interactions among members. The probabilistic model aggregates both the group member's individual preferences and their impact on the group to make recommendation. The PIT [10] distincts different user's contributions towards the final decision based on the assumption that the most significant user have a high impact on the group's decisions. The COM [11] captures the generative process of group activities supposing that a user's influence depends on the decided topic. However, these probabilistic models suffer from an incomplete assumption that a user has the same likelihood to follow the group's decisions across different groups.

Recently, deep neural networks are applied to group recommender systems thanks to their high-quality recommendation, which provides an in-depth understanding of user interests and the dynamic aggregation process of group preference. Cao et al. [5] incorporated the attention mechanism to aggregate the users' interests as the group representation in

*Xiaofei Zhou is the corresponding author.

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00036

a learnable way. To further refine the aggregation process, Vinh Tran et al. [6] applied the sub-attention networks to model every single user and the fine-grained interactions between group members. However, they rely on the user information within the group when aggregating user interests, while ignoring the relationships of users between groups. To address this limitation, some methods introduce cross-group social relationships to learn user interests. Guo et al. [12] proposed a stacked social self-attention network to model the social influence and user-user interactions. And the preference aggregation process is designed as a multiple voting step. Cao et al. [4] utilized social followee information as user-user relations to obtain the user's preference by aggregating the user's social friends representation. In summary, these methods usually utilize neural attention mechanisms to aggregate users' interests within the group or introduce social information to promote the representation of group member preferences.

Although the above methods have achieved good results, there are still several unresolved challenges: 1) These methods tend to isolate the groups when developing preference aggregation strategies, which is inadequate to capture cross-group collaborative signals. In Fig. 1, the group g_1's collaborative interests can be considered from user perspective and item perspective. From the collaborative user aspect, g_1 and g_2 have the overlap user u_2, indicating g_1 may be interested in g_2's history clicked item i_3. In view of the collaborative item, both g_1 and g_3 have clicked on i_2, showing that g_1 and g_3 have similar interests, then g_1 may be interested in g_3's historical interacted item i_4. Therefore, it is of significance to capture cross-group collaborative relation for better group preference modeling. 2) As a group may pursue some targets distinct from each user's interests, the above aggregation methods are not sufficient to characterize the group's general preference. For example, in a family of three, the child prefers cartoon movie and the parents are interested in comedy movie; but the final chosen movie could be an educational movie when they go to a cinema together. Thus modeling the group's general preference is a valuable research issue.

To address these challenges, we propose a dual channel Hypergraph Convolutional network for group Recommendation (HCR). First, we employ a member-level hypergraph convolutional network to integrate users' preferences across groups, capturing high-order collaborative information. We connect all group members and items as a hypergraph of overlapping sets, where each hyperedge consists of all group members and interacted items in the corresponding group. In order to learn the diverse group member preference, we apply multiple convolutional layers to propagate information along hyperedge. Second, we propose a group-level graph convolutional network to learn the group's general preference which is independent of the user's interest. We connect all groups based on group similarity and learn the representation of general group preference by the multi-layer convolution operation. With the dual channel hypergraph convolutional network, we devise a joint training strategy to learn both group-item and user-item interactions to further capture the member and group preference. The main contributions of this paper are summarized as follows:

- We propose a novel dual channel hypergraph convolutional network for group recommendation, which extracts collaborative information and group similarity to enhance group's member preference and general preference, respectively.
- We separately formulate group data as member-level hypergraph and group-level overlap graph tailored for graph convolutional network, which provides a new perspective to learn fine-grained group representation.
- We conduct experiments on two real-world datasets and show that the proposed framework significantly outperforms the state-of-the-art methods for the group recommendation.

II. Related work

A. Group Recommendation

In recent years, group recommendation [13] has been widely researched and applied in various domains. Existing methods can be characteristically dichotomized into memory-based and model-based approaches.

Memory-based methods can be further divided into two categories, namely preference aggregation and score aggregation. Preferences aggregation strategy obtains the group preference profile by combining the profiles of group members and then applies recommendation techniques tailored for individuals to make group recommendation. The score aggregation strategy computes the score of the candidate item for each user and then aggregates group member's scores as the group's preference through hand-crafted heuristics, such as average [7], least misery [8], and so on. Specifically, the average strategy takes the average score across members in the group as the final recommendation score to maximize overall group satisfaction. The least misery technique chooses the lowest score among all member's scores as the final scores. However, both approaches have weaknesses in measure group preference. The average approach may select items satisfied some members but not beneficial to others, while the least misery method may recommend ordinary items that no one either likes or dislikes. Baltrunas et al. [7] point out these methods are not strong enough, as their performance is dependent on group size and inner-group similarity. Amer-Yahia et al. [8] argue that group members' preference disagreements of items are inevitable and the experiments show that taking into account disagreement significantly improves the recommendation quality of the average and least misery strategies.

Different from memory-based methods, model-based approaches concentrate on the group preference aggregation process by capturing the interactions among members. The model-based approaches are further divided into the probabilistic model and neural model. The probabilistic model aggregates both the group member's individual preferences and their impact on the group to make recommendation. The PIT [10] constructs the group's preference profile assuming that the

most influential user is capable to represent the group and have a high impact on the group's decisions, thus differentiate different member's contributions towards the final decision. The COM [11] captures the generative process of group activities supposing that a user's influence depends on the decided topic and the group decision-making process is affected by both the group's preference topic and each member's interest. However, these probabilistic models suffer from an incomplete assumption that a user has the same likelihood to follow the group's decisions across different groups.

The neural model explores attention mechanisms [14]–[16] to provide an in-depth understanding of user interests and the dynamic aggregation process of group preference. Cao et al. [5] incorporated the attention mechanism to aggregate the members as the group representation in a learnable way. To further refine the aggregation process, Vinh Tran et al. [6] applied the sub-attention networks to model every single member and the fine-grained interactions between group members. However, they rely on the user information within the group when aggregating user interests, while ignoring the relationships of users between groups. To address this limitation, an alternative approach is to exploit external side information, e.g., social network of users, personality traits, and interpersonal relationships. For example, Guo et al. [12] proposed a stacked social self-attention network to model the social influence and user-user interactions. And the preference aggregation process is designed as a multiple voting step. Cao et al. [4] utilized social followee information as user-user relations to obtain the member preference by aggregating the user's social friends representation. In contrast, our setting is conservative and does not include extra side information: we know only user and item ids, and item implicit feedbacks. We capture the cross-group collaborative information through a novel dual channel hypergraph convolutional network.

B. Neural Graph-based Recommendation

With the recent advance in neural graph embedding algorithms, there is increasing attention on exploiting graph structures for various recommendation scenarios. These works utilize (hyper) graph neural networks (GNN) [17]–[19] to generate enriched latent representations for users and items. Through multi-layer propagation, the GNN-based methods capture the high-order connections in the graph. For example, LightGCN [20] proposes to solve the collaborative filtering task, which learns user and item embedding via graph convolutional propagation on the user-item bipartite graph. As an application on industrial-scale data, PinSage [21] proposes to generate item embeddings on a graph constructed with item-item connections, which can be applied for the downstream recommendation. In social recommendation [22]–[24], social connections between users can be investigated with GNN to model the propagation of user preference influenced by social friends. The graph-based sequential recommender systems [25]–[27] convert the clicked sequence to the item-item graph for GNN learning, aiming at exploring the complex sequential pattern and temporal preference. The recommender systems incorporating knowledge graph (KG) [28]–[30] take full advantage of the rich information in the KG to capture connectivity between items and further model the user's preferences for items. However, these methods are not designed for capturing the group preference.

Guo et al. [31] proposes a GNN-based model for group recommendation, which leverages friends' preference from the social network and connects group members as hyperedge to aggregate group preference. This method concentrates on utilize side information and aggregate member interest while ignores explicitly modeling the group's general preference. Compared with this approach, our proposed method captures both the group's member aggregation preference and general preference.

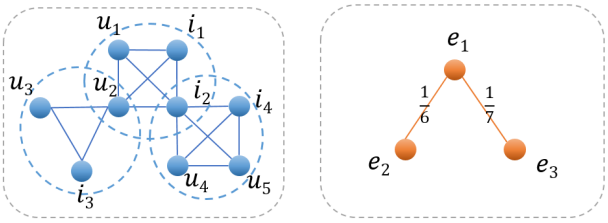
III. Notations and Problem Formulation

We use bold capital letters (e.g., $\mathbf{X}$) and bold lowercase letters (e.g., $\mathbf{x}$) to represent matrices and vectors, respectively. We employ nonbold letters (e.g., x) to denote scalars, and squiggle letters (e.g., $\mathcal{X}$) to denote sets. If not clarified, all vectors are in column forms.

Let $\mathcal{U} = \{u_1, u_2, ..., u_m\}$, $\mathcal{I} = \{i_1, i_2, ..., i_n\}$, and $\mathcal{G} = \{g_1, g_2, ..., g_k\}$ be the sets of users, items, and groups. m, n, and k denote the numbers of users, items, and groups in the three sets respectively. Each user and group interacts with different items, which indicate their preferences. There are two kinds of observed interaction data among $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{G}$, namely, group-item interactions and user-item interactions. We use $\mathbf{Y} = [y_{th}]_{k\times n}$ to denote the group-item interactions and $\mathbf{R} = [r_{jh}]_{m\times n}$ to denote the user-item interactions. The t-th group $g_t \in \mathcal{G}$ consists of a set of user members $\mathcal{G}_t = \{u_1, u_2, ..., u_j..., u_{|\mathcal{G}_t|}\}$, where $u_j \in \mathcal{U}$, $|\mathcal{G}_t|$ is the size of $\mathcal{G}_t$, and $\mathcal{G}_t$ is the user set of g_t. We denote the set of items which are interacted by g_t as $\mathcal{Y}_t = \{i_1, i_2, ..., i_h..., i_{|\mathcal{Y}_t|}\}$, where $i_h \in \mathcal{I}$, $|\mathcal{Y}_t|$ is the size of $\mathcal{Y}_t$, and $\mathcal{Y}_t$ is the item set of g_t. Then, given a target group g_t (or target user u_j), our task is defined as recommending a list of items that group g_t (or target user u_j) may be interested in. Formally, we aim to learn a function that maps an item to a real-valued score which indicates its probability of being interacted by the target group (or target user u_j): $f_t : \mathcal{I} \to \mathbb{R}$.

Definition 1. Hypergraph. Let $\mathcal{G}^h = (\mathcal{V}^h, \mathcal{E}^h)$ denote a hypergraph, where $\mathcal{V}^h$ is a set containing N unique vertices and $\mathcal{E}^h$ is a set containing M hyperedges. Each hyperedge $\epsilon \in$ contains two or more vertices and is assigned a positive weight $W_{\epsilon\epsilon}$. The hyperedge weights formulate a diagonal matrix $\mathbf{W} \in \mathbb{R}^{M\times M}$. The incidence matrix $\mathbf{H} \in \mathbb{R}^{N\times M}$ is utilized to represent the hypergraph. $H_{i\epsilon} = 1$ if the hyperedge $\epsilon \in \mathcal{E}^h$ contains a vertex $v_i \in \mathcal{V}^h$, otherwise 0. The vertex degree D_{ii} defined as $D_{ii} = \sum_{\epsilon=1}^{M} W_{\epsilon\epsilon} H_{i\epsilon}$, and $\mathbf{D}$ is the diagonal degree matrices of vertex. The hyperedge degree $B_{\epsilon\epsilon}$ is defined as $B_{\epsilon\epsilon} = \sum_{i=1}^{N} H_{i\epsilon}$, and $\mathbf{B}$ is the diagonal degree matrices of hyperedge.

Definition 2. Overlap Graph of Hypergraph. Given the hypergraph $\mathcal{G}^o = (\mathcal{V}^o, \mathcal{E}^o)$, the overlap graph of the hypergraph $\mathcal{G}^o$ is a graph where each node of $\mathcal{G}^o$ is a hyperedge in



(a) Hypergraph (b) Overlap graph

Fig. 2. The example of constructed hypergraph and overlap graph.

$\mathcal{G}_h$. Two nodes of $\mathcal{G}^o$ are connected if their corresponding hyperedges in $\mathcal{G}_h$ share at least one common node [32]. Formally, $\mathcal{G}^o = (\mathcal{V}^o, \mathcal{E}^o)$ where $\mathcal{V}^o = \{e : e \in \mathcal{E}^h\}$, $\mathcal{E}^o = \{(e_p, e_q) : e_p, e_q \in \mathcal{E}^h, |e_p \cap e_q| \geq 1\}$. We assign each edge (e_p, e_q) a weight $W_{p,q}$, where $W_{p,q} = |e_p \cap e_q| \,/\, |e_p \cup e_q|$.

IV. The Proposed Model

In this section, we firstly show the way to model group-based data as a hypergraph. Based on the hypergraph, we present our dual channel hypergraph convolutional network from member-level and group-level for group recommendation, respectively. Finally, we show the joint training strategy to learn both group-item and user-item interactions to further capture the member and group preference. The overall architecture of HCR is shown in Fig. 3.

A. Hypergraph Construction

To capture the beyond collaborative relations among group members and interacted items, we adopt a hypergraph $\mathcal{G}^h = (\mathcal{V}^h, \mathcal{E}^h)$ to represent each group as a hyperedge. Each hyperedge consists of the group members and the interacted items. Formally, we denote each hyperedge as $\mathcal{E}^h_t = \mathcal{G}_t \cap \mathcal{Y}_t$, i.e., $\mathcal{E}^h_t = \{u_1, u_2, ..., u_{|\mathcal{G}_t|}, i_1, i_2, ..., i_{|\mathcal{Y}_t|}\}$, where each $u_* \in \mathcal{U}$, each $i_* \in \mathcal{I}$, and $\mathcal{E}^h_t \in \mathcal{E}^h$. In Fig. 2(a), we show the example of hypergraph construction, which is transformed from the group interaction data shown in Fig. 1. As illustrated, the original group interaction is organized as a bipartite graph where the group and items are connected if the group interacted with the items. After transforming the interaction data into a hypergraph, the group members and interacted items in the hyperedge are explicitly connected. By doing so, the many-to-many high-order relations are captured from the graph. In Fig. 2(b), we further induce the overlap graph of the hypergraph according to Definition 2. Each group is modeled as a node and different groups are connected via shared members and items. Compared with the hypergraph which extracts the member-level high-order relations, the overlap graph indicates the group-level relations.

B. Member-level Preference Network

In this subsection, we propose to model the member-level preference from member's personal interest. We first learn the high-order collaborative representation of users and items from the hypergraph convolutional network. Then, we aggregate the member interest as the group's member-level preference with attention network.

Member-level Hypergraph Convolutional Network. We have constructed the hypergraph as the basis for member preference modeling. In the hypergraph, the collective users and items shared by two hyperedges indicate the collaborative interest among groups. To learn the user and item representation from the hypergraph, we propose to apply the hypergraph convolution operation to encode the high order relations among users and items. Specifically, we feed the concatenation of user embeddings $\mathbf{U} \in \mathbb{R}^{m \times d}$ and item embeddings $\mathbf{I} \in \mathbb{R}^{n \times d}$ to the hypergraph convolutional network, denoted as $\mathbf{X}^{(0)} = [\mathbf{U}; \mathbf{I}]$. Referring to the spectral hypergraph convolution [33], we define our hypergraph convolution as:

$$\mathbf{x}_i^{(l+1)} = \sum_{j=1}^{N} \sum_{\epsilon=1}^{M} H_{i\epsilon} H_{j\epsilon} W_{\epsilon\epsilon} \mathbf{x}_j^{(l)} \mathbf{P}^{(l)}, \tag{1}$$

where $\mathbf{x}_i^{(l)}$ is the embedding of the i-th vertex in the (l)-th layer. $\mathbf{P} \in \mathbb{R}^{d \times d}$ is the learnable weight matrix between two convolutional layers. To keep the model simplified, nonlinear activation function is not used. Each hyperedge is assigned the same weight 1, indicating every group is treated equally important. The matrix form of Eq. (1) with row normalization is:

$$\mathbf{X}^{(l+1)} = \mathbf{D}^{-1}\mathbf{H}\mathbf{W}\mathbf{B}^{-1}\mathbf{H}^T\mathbf{X}^{(l)}\mathbf{P}^{(l)}. \tag{2}$$

The hypergraph convolution can be viewed as a two-stage refinement performing 'node-hyperedge-node' feature transformation upon hypergraph structure. The multiplication operation $\mathbf{H}^T\mathbf{X}^{(l)}$ defines the information aggregation from nodes to hyperedges. Intuitively, the group aggregates the user members and interacted items as its representation. The premultiplication of $\mathbf{H}$ is viewed to aggregate information from hyperedges to nodes. This process extracts collaborative information across groups to enhance the user and item representation.

After passing $\mathbf{X}^{(0)}$ through L hypergraph convolutional layers, we average the embeddings obtained at each layer to get the collaborative representation $\mathbf{X}^* = \frac{1}{L+1}\sum_{l=0}^{L}\mathbf{X}^{(l)}$. The average operation can preserve collaborative relations of different orders. We devide the collaborative representation $\mathbf{X}^*$ to user representation $\mathbf{U}^*$ and item representation $\mathbf{I}^*$, denoted as $[\mathbf{U}^*; \mathbf{I}^*] = \mathbf{X}^*$.

Member Preference Aggregation Network. Intuitively, if a user has more expertise on an item, he should have a larger influence on the group's choice on the item. For example, the group considers which movie to watch. If a user has watched comedies many times, then he will be more influential when the group considers watching a comedy film. We get the group's member-level representation $\mathbf{g}_t^M$ by performing a weighted sum on the representation of group g_t's member users:

$$\mathbf{g}_t^M = \sum_{\mathbf{u}_j \in G_t} \alpha(j, h)[\mathbf{u}_j; \mathbf{u}_j^*], \tag{3}$$

where the coefficient $\alpha(j, h)$ is a learnable parameter denoting the influence of member user $\mathbf{u}_j$ in deciding the group's

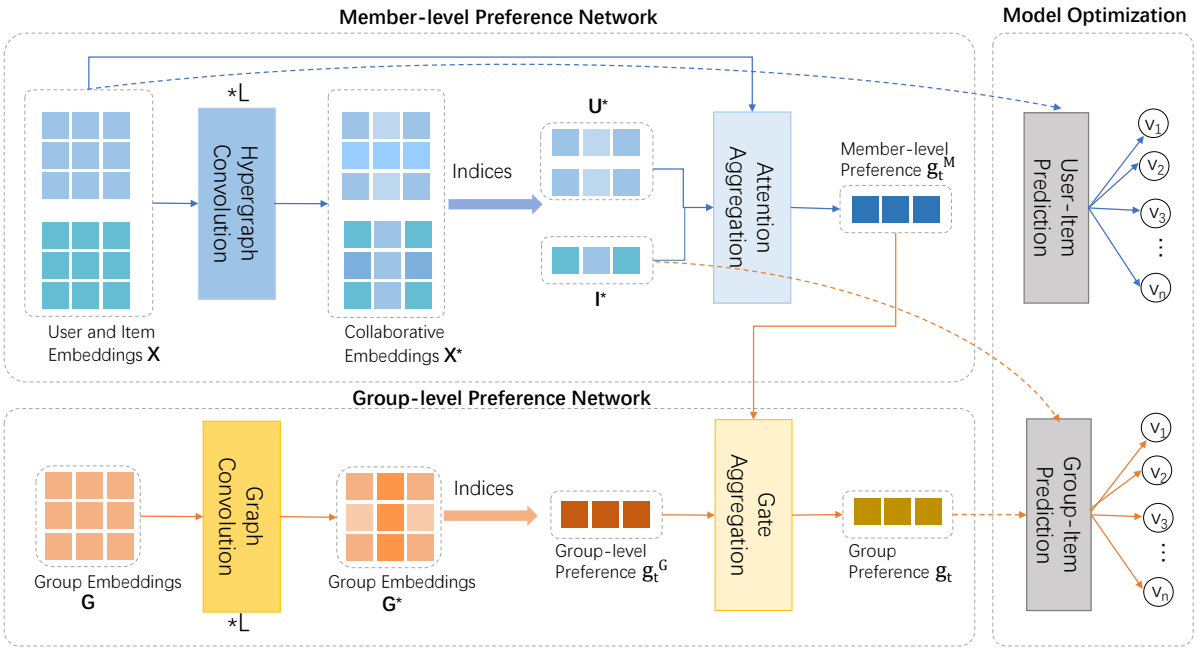


Fig. 3. Overall architecture of the proposed model HCR. The model consists of member-level preference network and group-level preference network. We jointly optimize the model with user-item and group-item interactions.

choice on item i_h. Specifically, the representation $\mathbf{u}_j$ encodes the member user's historical preference and the representation $\mathbf{u}_j^*$ contains collaborative information. Similarly, the representation $\mathbf{i}_h$ encodes the target item's property and the representation $\mathbf{i}_h^*$ captures the collaborative characteristics. We define $\alpha(j,h)$ as a attention network with $\mathbf{u}_j$, $\mathbf{u}_j^*$, $\mathbf{i}_h$, and $\mathbf{i}_h^*$ as input:

$$\begin{aligned}\alpha(j,h) &= \mathbf{h}^T \mathrm{ReLU}(\mathbf{W}_u[\mathbf{u}_j;\mathbf{u}_j^*] + \mathbf{W}_i[\mathbf{i}_h;\mathbf{i}_h^*] + \mathbf{b}),\\ \alpha(j,h) &= \frac{\exp(\alpha(j,h))}{\sum_{u_{h'}\in G_t}\alpha(j,h')},\end{aligned} \tag{4}$$

where $\mathbf{W}_u$ and $\mathbf{W}_i$ are weight matrices of the attention network that convert item representation and user representation to hidden layer, respectively. We use ReLU as the activation function of the hidden layer, and then project it to a score $\alpha(j,h)$ with a weight vector $\mathbf{h}$. A softmax function normalizes the scores, which is a common practice in neural attention network to adapt groups of different sizes. The soft attention mechanism can compute the contribution of a user in a group's decision, where the contribution of a user is dependent on his historical/collaborative preference and the target item's origin/collaborative property.

C. Group-level Preference Network

Simply aggregating group member's individual interests as the group's preference would miss the intrinsic group-level preferences which may be different and independent from all individuals' preferences within the group. We first capture the group-level preferences from the graph convolution network on the overlap graph of the hypergraph. Then, we aggregate the member-level preference and group-level preference to get the final representation of the group.

Group-level Graph Convolutional Network. The constructed overlap graph of the hypergraph is the basis for group-level preference modeling. The overlap graph can be seen as a simple graph that contains the cross-group information and depicts the connectivity of hyperedges. To learn group representation from the hypergraph, we propose to apply the graph convolution operation to encode the high order relations among groups based on the group similarity. Note that here we are not concerned with the process of group member aggregation, but only learn the representation of independent group preference. Specifically, we feed the group embeddings $\mathbf{G} \in \mathbb{R}^{k\times d}$ to the graph convolutional network, denoted as $\mathbf{G}^{(0)} = \mathbf{G}$. Referring to the spectral graph convolution, we define our graph convolution as:

$$\mathbf{G}^{(l+1)} = \hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}\mathbf{G}^{(l)}\mathbf{Q}^{(l)}, \tag{5}$$

where $\mathbf{Q}^{(l)} \in \mathbb{R}^{d\times d}$ is the learnable weight matrix. According to Definition 2, the incidence matrix for $L(G)$ is defined as $\mathbf{A} \in \mathbb{R}^{M\times M}$ where M is the number of nodes in the overlap graph and $A_{p,q} = W_{p,q}$. We set the matrix $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ where $\mathbf{I}$ is an identity matrix. We denote the diagonal matrix as $\hat{\mathbf{D}} \in \mathbb{R}^{M\times M}$ where $\hat{\mathbf{D}}_{p,q} = \sum_{q=1}^{m}\hat{\mathbf{A}}_{p,q}$. Intuitively, two groups with group members tend to have similar general preferences, and the more common members two groups share, the higher their similarity will be. In each convolution, the premultiplication $\hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}$ considers the group similarity when

TABLE I
THE STATISTICS OF THE DATASETS.

Dataset	#Users	#Groups	#Items	#User interactions	#Group interactions	#Avg. interactions per user	#Avg. interactions per group	#Avg. users per group
Mafengwo	5275	995	1513	39761	3595	7.54	3.61	7.19
CAMRa2011	602	290	7710	116344	145068	193.26	500.23	2.08

the group gather information from their neighbors. By doing so, the learned $\mathbf{G}$ can capture the cross-group information.

Likewise, we pass $\mathbf{G}^{(0)}$ through L graph convolutional layer, and then average the group embeddings obtained at each layer to get the final goup-level embeddings $\mathbf{G}^* = \frac{1}{L+1}\sum_{l=0}^{L}\mathbf{G}^{(l)}$. The group-level preference for group g_t is denoted as $\mathbf{g}_t^G = \mathbf{G}^*_{t,:}$.

Group Preference Aggregation Network. We have obtained the member-level group embedding $\mathbf{g}_t^M$ and group-level group embedding $\mathbf{g}_t^G$ from the dual channel hypergraph convolutional network. The next aim is to combine these two kinds of hidden representations to facilitate the group preference prediction on unrated items. We propose a neural gating layer to adaptively merge them. This is inspired by the gates in long short-term memory (LSTM) [34]. The gate $\mathbf{r}$ and the final group representation $\mathbf{g}_t$ are computed by:

$$\begin{aligned}\mathbf{r} &= \text{sigmoid}(\mathbf{W}_{g_1}\mathbf{g}_t^M + \mathbf{W}_{g_2}\mathbf{g}_t^G + \mathbf{b}_g),\\ \mathbf{g}_t &= \mathbf{r}\odot\mathbf{g}_t^M + (\mathbf{1}-\mathbf{r})\odot\mathbf{g}_t^G,\end{aligned} \tag{6}$$

where $\mathbf{W}_{g_1} \in \mathbb{R}^{d\times d}$, $\mathbf{W}_{g_2} \in \mathbb{R}^{d\times d}$, and $\mathbf{b}_g \in \mathbb{R}^d$ are the learnable parameters in the gating aggregation network. By using the gating aggregation network, the salient parts from these two hidden representations can be extracted and smoothly combined.

D. Model Optimization

For the group-item pair (g_t, i_h), we feed the concatenation of the the two embeddings and the vector of their element-wise product to compute the prediction score $\hat{y}_{th}$ of group g_t item i_h:

$$\hat{y}_{th} = \mathbf{w}^T\text{ReLU}(\mathbf{W}_f[\mathbf{g}_t\odot\mathbf{i}_h;\mathbf{g}_t;\mathbf{i}_h] + \mathbf{b}_f), \tag{7}$$

where $\mathbf{i}_h$ is the item embedding of the target item i_h. $\mathbf{w}$ denote the weights of the prediction layer. $\mathbf{W}_f$ and $\mathbf{b}_f$ denotes the weight matrix and bias vector of a feed-forward network.

Due to the implicit nature of the group-item interaction data, we utilize pairwise learning method for optimizing model parameters. Specifically, we employ the regression-based pairwise loss motivated by [4]:

$$\mathcal{L}_{group} = \sum_{(t,h,h')\in\mathcal{D}_G}(\hat{y_{th}} - \hat{y_{th'}} - 1)^2, \tag{8}$$

where $\mathcal{D}_G$ denotes the group-item training set, in which each instance is a triplet (t, h, h') meaning that group g_t has interacted with item i_h, but has not interacted with item $i_{h'}$. We set the margin of the prediction on the observed interaction (g_t, i_h) and the unobserved interaction $(g_t, i_{h'})$ as 1.

Because of the data sparsity of group interactions, the learned group representation is not sufficiently accurate. To further accelerate and enhance the group preference learning, we propose to incorporate the user-item interaction data to optimize the group-item and user-item recommendation tasks simultaneously. Similarly, for the user-item pair (u_j, i_h), we compute the prediction score $\hat{r}_{jh}$ of user u_j item i_h:

$$\hat{r}_{jh} = \mathbf{w}^T\text{ReLU}(\mathbf{W}_f[\mathbf{u}_j\odot\mathbf{i}_h;\mathbf{u}_j;\mathbf{i}_h] + \mathbf{b}_f), \tag{9}$$

where the parameter are shared with the group-item feed-forward network. The same pairwise loss function is utilized:

$$\mathcal{L}_{user} = \sum_{(j,h,h')\in\mathcal{D}_U}(\hat{r_{jh}} - \hat{r_{jh'}} - 1)^2, \tag{10}$$

where $\mathcal{D}_U$ denotes the user-item training set; the triplet (j, h, h') represents user u_j prefers observed item i_h over unobserved item $i_{h'}$.

V. EXPERIMENTS

In this section, we present our experimental setup and empirical results. Our experiments are designed to answer the following research questions:

RQ1: How does HCR perform in terms of group recommendation compared to other state-of-the-art methods?

RQ2: How do the dual channel hypergraph convolutional network (i,e., member-level hypergraph convolutional network and group-level graph convolutional network) contribute to the performance of our solutions?

RQ3: How do different predefined settings (e.g., the number of negative samples and convolutional layers) affect our framework?

A. Experimental Setup

Datasets. We conduct experiments on two real-world datasets, Mafengwo[1], and CAMRa2011[2]. As preprocessing, we filtered out the groups which have at least 2 members and have interacted with at least 3 items. As both datasets only contain positive instances (i.e., observed interactions), we randomly sampled from missing data as negative instances to pair with each positive instance. Table I reports the detailed statistics of these datasets.

Baseline Methods. To justify the effectiveness of our methods, we compared them with the following methods.

- **NCF** [35]: This method treats a group as a virtual user and ignores the member information of the group. Then

[1]http://www.mafengwo.cn
[2]http://2011.camrachallenge.com/2011

TABLE II
THE PERFORMANCE COMPARISON OF ALL METHODS ON GROUP RECOMMENDATION TASK IN TERMS OF HR@K AND NDCG@K. THE BEST-PERFORMING METHOD IS BOLDFACED. THE IMPROVEMENT OVER BASELINE METHODS IS SHOWN IN THE LAST COLUMN.

Dataset	Metric	NCF	Popularity	COM	UL_ALL	AGR	SoAGREE	HCR	Improv.
Mafengwo	HR@5	0.4701	0.3115	0.4432	0.4687	0.4729	0.4898	**0.7759**	58.4%
	NDCG@5	0.3657	0.2169	0.3325	0.3643	0.3694	0.3807	**0.6611**	73.6%
	HR@10	0.6269	0.4251	0.5528	0.6252	0.6321	0.6481	**0.8503**	31.2%
	NDCG@10	0.4141	0.2537	0.3812	0.4127	0.4203	0.4301	**0.6852**	59.3%
CAMRa2011	HR@5	0.5803	0.4324	0.5798	0.5559	0.5879	0.5883	**0.6772**	15.1%
	NDCG@5	0.3896	0.2825	0.3785	0.3765	0.3933	0.3955	**0.6115**	54.1%
	HR@10	0.7693	0.5793	0.7695	0.7624	0.7789	0.7807	**0.8193**	4.9%
	NDCG@10	0.4448	0.3302	0.4385	0.4400	0.4530	0.4575	**0.6576**	43.7%

TABLE III
THE PERFORMANCE COMPARISON OF ALL METHODS ON USER RECOMMENDATION TASK IN TERMS OF HR@K AND NDCG@K. THE BEST-PERFORMING METHOD IS BOLDFACED. THE IMPROVEMENT OVER BASELINE METHODS IS SHOWN IN THE LAST COLUMN.

Dataset	Metric	NCF	Popularity	AGR	SoAGREE	HCR	Improv.
Mafengwo	HR@5	0.6363	0.4047	0.6357	0.6510	**0.7571**	16.3%
	NDCG@5	0.5432	0.2876	0.5481	0.5612	**0.6703**	19.4%
	HR@10	0.7417	0.4971	0.7403	0.7610	**0.8290**	11.7%
	NDCG@10	0.5733	0.3172	0.5738	0.5775	**0.6937**	8.9%
CAMRa2011	HR@5	0.6119	0.4624	0.6196	0.6223	**0.6731**	8.1%
	NDCG@5	0.4018	0.3104	0.4098	0.4118	**0.4608**	11.9%
	HR@10	0.7894	0.6026	0.7897	0.7967	**0.8595**	7.9%
	NDCG@10	0.4535	0.3560	0.4627	0.4687	**0.5219**	11.4%

users and virtual users are embedded to predict with the same network and hyper-parameter setting of HCR.

- **Popularity** [36]: This method recommends items to users and groups based on the popularity of items. The popularity of an item is measured by its number of interactions in the training set.
- **COM** [11]: This is a group-oriented recommender system, which is based on the probability theory to model the generative process of group activities.
- **UL_ALL** [37]: This is a group recommendation algorithm, which involves proposing an upward leveling aggregation method to consider deviations for group recommendations.
- **AGR** [5]: This is an attention-based group recommendation solution. It learns the attention weight of a user by considering the impact of other group members, which ignores the influence of items.
- **SoAGREE** [4]: This is a social information enhanced method for group recommendation. It designs hierarchical attention network learning to represent groups and users in a hierarchical structure.

Evaluation Metrics. We adopted the leave-one-out evaluation protocol, which has been widely utilized to evaluate the performance of the Top-N recommendation [38]. Specifically, for each user (group), we randomly removed one of its interactions for testing. This results in disjoint training set and testing set Stest. Since it is too time-consuming to rank all items for each user and group, we followed the common scheme that randomly selected 100 items that were not interacted by the user or the group and ranked the testing item among the 100 items. To evaluate the performance of the Top-N recommendation, we employed the widely used metric, Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). The size N of the ranked list was chosen to be [5, 10] for HR@N and NDCG@N, respectively. HR@N examines whether or not the test item is present in the top N list, and NDCG@N places more weights on higher-ranked items than others in the top N list.

Experiment Settings. We implemented our method based on PyTorch. For hyper-parameter tuning, we randomly sampled one interaction for each user and group as the validation set. As have mentioned before, the negative sampling ratio was set to 4. For the initialization of the embedding layer, we applied the Glorot initialization strategy [39], which was found to have a good performance. For hidden layers, we randomly initialized their parameters with a Gaussian distribution of a mean of 0 and a standard deviation of 0.1. We used the Adam optimizer [40] for all gradient-based methods, where the mini-batch size and learning rate were searched in [128, 256, 512, 1024] and [0.01, 0.005, 0.01, 0.05, 0.1], respectively. We set the group (user, item) embedding dimension as 32. In the dual hypergraph convolutional network, we empirically set the size of the convolutional layer as 3. We repeated each set for 5 times and reported the average results. Our experiments are conducted with PyTorch running on GPU machines of Nvidia GeForce GTX 1080 Ti5[3].

B. Experimental Results

Overall Performance Comparison We compare the performance of HCR with the baselines. Note that since COM and UL_ALL are specifically designed for group recommendation, they can not provide recommendation for individual users.

[3]The code is available on http://github.com/GroupRec/GroupRec

Table 2 and Table 3 show the experimental performance of group recommendation task and user recommendation task, respectively. The result tables depict the performance of all methods under two metrics, i.e. Recall@K, and NDCG@K in two real-world datasets. From the tables, we can observe that:

- For the group recommendation task, HCR consistently outperforms all baselines under all evaluation metrics in two benchmark datasets. Specifically, for Mafengwo dataset, HCR improves 58.4%, 73.6%, 31.2%, and 59.3% in terms of HR@5, NDCG@5, HR@10, and NDCG@100, compared with the second best method (i.e., SoAGREE) in the three metrics respectively. For CAMRa2011 dataset, HCR improves 15.1%, 54.1%, 4.9%, and 43.7% in terms of HR@5, NDCG@5, HR@10, and NDCG@100, compared with the second best method (i.e., SoAGREE) in the three metrics respectively. The above experimental results illustrate the effectiveness of HCR in group preference learning.
- For the user recommendation task, HCR consistently outperforms all baselines under all evaluation metrics in two benchmark datasets. Specifically, for Mafengwo dataset, HCR improves 19.2%, 73.617.8%, 11.7%, and 18.6% in terms of HR@5, NDCG@5, HR@10, and NDCG@100, compared with the second best method (i.e., SoAGREE) in the three metrics respectively. For CAMRa2011 dataset, HCR improves 8.8%, 48.8%, 2.8%, and 40.3% in terms of HR@5, NDCG@5, HR@10, and NDCG@100, compared with the second best method (i.e., SoAGREE) in the three metrics respectively. The above experimental results illustrate the effectiveness of HCR in user preference learning.
- The performance of neural network-based methods (i.e. NCF, AGR, SoAGREE, HCR) is superior to that of non-personalized approach (Popularity), probabilistic graphical model (COM), and aggregation method (UL_ALL). This demonstrates the superiority of neural networks, especially among users, groups, and items.
- The other neural network-based methods (i.e., NCF, AGR, and SoAGREE) achieves good performance in the two datasets. Specifically, NCF and AGR are roughly the same in terms of the metrics. This indicates that the feed-forward network utilized in NCF and the neural attention mechanism used in AGR are both strong in learning the representation of preference for users and groups. While SoAGREE outperforms NCF and AGR under all evaluation metrics in the two datasets. This is because that SoAGREE aggregates the social followee to represent the user preference. It is more effective than directly learning the preference embedding for each user.

Importance of Components. We evaluate the effect of the member-level hypergraph convolutional network (HCR_M), group-level graph convolutional network (HCR_G), and the complete model (HCR). The experimental results of group recommendation task and user recommendation task are shown in Table 4 and Table 5, respectively.

TABLE IV
THE PERFORMANCE COMPARISON OF ALL ABLATION METHODS (I.E. HCR_M, HCR_G, AND HCR) ON GROUP RECOMMENDATION TASK IN TERMS OF HR@K AND NDCG@K.

Dataset	Metric	HCR_M	HCR_G	HCR
Mafengwo	HR@5	0.6653	0.7669	0.7759
	NDCG@5	0.4548	0.6529	0.6611
	HR@10	0.7829	0.8332	0.8503
	NDCG@10	0.4935	0.6738	0.6852
CAMRa2011	HR@5	0.6490	0.6331	0.6772
	NDCG@5	0.4755	0.4166	0.6115
	HR@10	0.8228	0.8090	0.8193
	NDCG@10	0.5321	0.4743	0.6576

TABLE V
THE PERFORMANCE COMPARISON OF ALL ABLATION METHODS (I.E. HCR_M, HCR_G, AND HCR) ON USER RECOMMENDATION TASK IN TERMS OF HR@K AND NDCG@K.

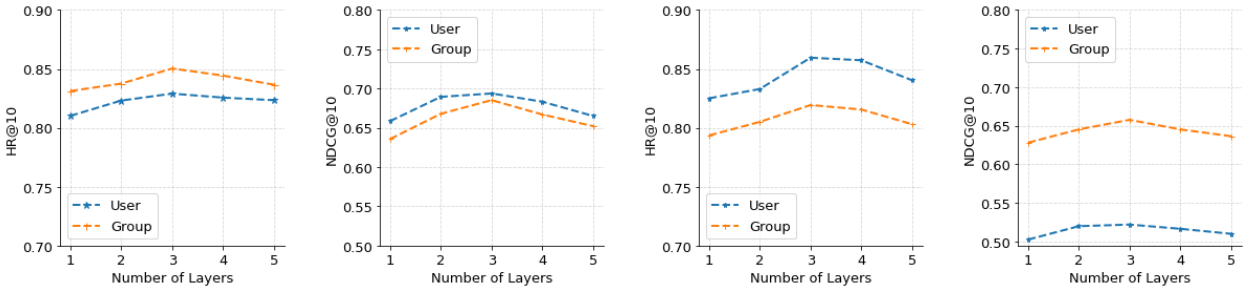
Dataset	Metric	HCR_M	HCR_G	HCR
Mafengwo	HR@5	0.7765	0.7457	0.7571
	NDCG@5	0.7083	0.6506	0.6703
	HR@10	0.8358	0.8158	0.8290
	NDCG@10	0.7277	0.6736	0.6937
CAMRa2011	HR@5	0.6591	0.6711	0.6731
	NDCG@5	0.4337	0.5152	0.4608
	HR@10	0.8495	0.8425	0.8595
	NDCG@10	0.4963	0.5709	0.5219

From Table 4, we can observe:

- For the group recommendation task, HCR consistently outperforms the other model at each step, indicating the effectiveness of the member-level hypergraph convolutional network and group-level graph convolutional network. The result implies that the member-level preference and group-level preference are both significant for group recommendation.
- HCR_M performs better than HCR_G on the Mafengwo dataset. We believe that the results are determined by the characteristics of the two datasets. For the Mafengwo dataset, since the number of average group members and total users are both larger, there are more combinations of the way to aggregate user interests. Meanwhile, there are fewer group interactions in the Mafengwo dataset, resulting in underfitting due to data sparsity. So it is more suitable to use the method of aggregating member-level member interests than directly modeling the group-level preference.
- HCR_G performs better than HCR_M on the CAMRa2011 dataset. For the CAMRa2011 dataset, the number of average group members and total users is both fewer, resulting in fewer combinations of member interests. So it is difficult to use the method of aggregating member-level member interests. Due to the abundance of group interaction data, it is possible to utilize these data to model group-level interests well.

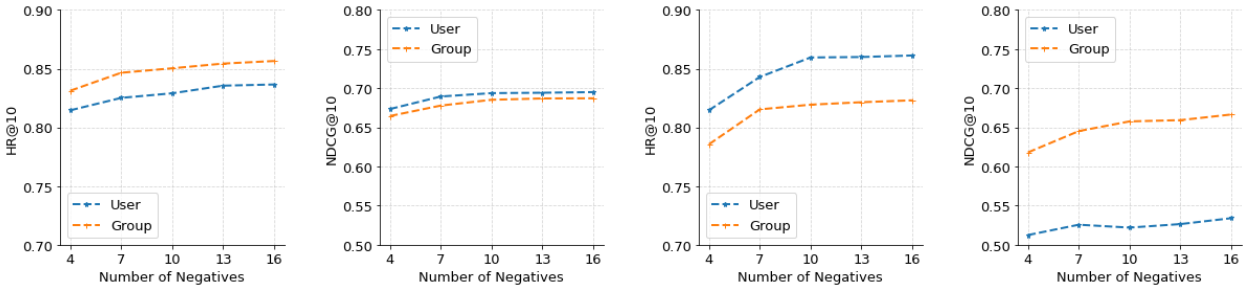
From Table 5, we can observe:

- For the user recommendation task, HCR_M achieves the best performance on Mafengwo dataset. In the Mafengwo



(a) Mafengwo-HR@10 (b) Mafengwo-NDCG@10 (c) CAMRa2011-HR@10 (d) CAMRa2011-NDCG@10

Fig. 4. Performance of HCR w.r.t. the number of convolutional layers



(a) Mafengwo-HR@10 (b) Mafengwo-NDCG@10 (c) CAMRa2011-HR@10 (d) CAMRa2011-NDCG@10

Fig. 5. Performance of HCR w.r.t. the number of negative samples for each positive instances

dataset, due to a large number of users, group-level group preference modeling may interfere with user interest learning.

- For the user recommendation task, HCR achieves the best performance on CAMRa2011 dataset. In the CAMRa2011 dataset, the number of users is smaller, so the process of group-level interests modeling assists in capturing user interests.

Effect of model Hyperparameters. The performance of the graph convolutional network is affected by the number of graph convolution layers. As the number of layers increases, the convolutional neural networks are faced with the problem of over-smoothing, so the node representation of the high layer is not discriminative enough to distinguish different groups' preferences. To illustrate the impact of convolutional operation for HCR, we show the performance of AGREE w.r.t. the different number of convolutional layers on both Mafengwo and CAMRa2011 datasets in Fig. 4. We observe that when the layer is 3, better results are shown in two datasets. The larger layer has equal results with the smaller ones. One possible reason is that the node of the high layer may learn the over-smoothing representation [41], which is detrimental to the recommendation effect.

The strategy of negative sampling has been proven rational and effective in [42]. It randomly samples various numbers of missing data as negative samples to pair with each positive instance. With more negative samples selected, the performance of negative sampling becomes stable and approximates the result of all missing data considered. To illustrate the impact of negative sampling for HCR, we show the performance of AGREE w.r.t. different negative sample ratios on both Mafengwo and CAMRa2011 datasets in Fig. 5. We can observe that too small negative samples are not enough to utilize the model performance. It is obviously seen that sampling more negative samples is beneficial. By increasing the number of negative samples, the model has more capacity to optimize. With the increase of negative samples, the model performance largely improves and becomes steady.

VI. Conclusion

In this paper, we propose a dual channel Hypergraph Convolutional network for group Recommendation (HCR) for group recommendation. We separately formulate group data as member-level hypergraph and group-level overlap graph tailored for graph convolutional network, which provides a new perspective to learn fine-grained group representation. The member-level hypergraph convolutional network integrates member preferences across groups, capturing high-order collaborative information and the group-level graph convolutional network learns the group's general preference independent of the user's interest. We conduct experiments on two real-world datasets and show that the proposed framework significantly outperforms the state-of-the-art methods for the group recommendation.

REFERENCES

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[2] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[3] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[4] D. Cao, X. He, L. Miao, G. Xiao, H. Chen, and J. Xu, "Social-enhanced attentive group recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[5] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 645–654.

[6] L. Vinh Tran, T.-A. Nguyen Pham, Y. Tay, Y. Liu, G. Cong, and X. Li, "Interact and decide: Medley of sub-attention networks for effective group recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 255–264.

[7] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 119–126.

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

[9] L. Boratto and S. Carta, "State-of-the-art in group recommendation and new approaches for automatic identification of groups," in *Information retrieval and mining in distributed environments*. Springer, 2010, pp. 1–20.

[10] X. Liu, Y. Tian, M. Ye, and W.-C. Lee, "Exploring personal impact for group recommendation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 674–683.

[11] Q. Yuan, G. Cong, and C.-Y. Lin, "Com: a generative model for group recommendation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 163–172.

[12] L. Guo, H. Yin, Q. Wang, B. Cui, Z. Huang, and L. Cui, "Group recommendation with latent voting mechanism," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 121–132.

[13] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 566–577.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[16] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao, and C. Zheng, "Synthesizer: Rethinking self-attention in transformer models," *arXiv preprint arXiv:2005.00743*, 2020.

[17] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognition*, vol. 110, p. 107637, 2021.

[18] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[19] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *arXiv preprint arXiv:1706.02216*, 2017.

[20] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.

[21] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 974–983.

[22] L. Wu, P. Sun, R. Hong, Y. Fu, X. Wang, and M. Wang, "Socialgcn: an efficient graph convolutional network based model for social recommendation," *arXiv preprint arXiv:1811.02815*, 2018.

[23] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proceedings of the Web Conference 2021*, 2021, pp. 413–424.

[24] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, "Diffnet++: A neural influence and interest diffusion network for social recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[25] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1172–1180.

[26] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 579–588.

[27] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation." in *IJCAI*, vol. 19, 2019, pp. 3940–3946.

[28] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The world wide web conference*, 2019, pp. 3307–3313.

[29] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 950–958.

[30] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[31] L. Guo, H. Yin, T. Chen, X. Zhang, and K. Zheng, "Hierarchical hyperedge embedding-based representation learning for group recommendation," *arXiv preprint arXiv:2103.13506*, 2021.

[32] H. Whitney, "Congruent graphs and the connectivity of graphs," in *Hassler Whitney Collected Papers*. Springer, 1992, pp. 61–79.

[33] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3558–3565.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[36] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 39–46.

[37] Y.-D. Seo, Y.-G. Kim, E. Lee, K.-S. Seol, and D.-K. Baik, "An enhanced aggregation method considering deviations for a group recommendation," *Expert Systems with Applications*, vol. 93, pp. 299–312, 2018.

[38] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[42] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 825–833.