Key Users Identification-based Heterogeneous Hypergraph for Group Recommendation

Lijin Mu[1], Nan Wang[1]✉, and Jinbao Li[2]⋆

[1] College of Computing and Big Data, Heilongjiang University, Harbin 150080, China
2232015@s.hlju.edu.cn, wangnan@hlju.edu.cn
[2] Shandong Artificial Intelligence Institute, School of Mathematics and Statistics, Qilu University of Technology, Jinan 250353, China
lijinb@sdas.org

Abstract. One of the major challenges facing the group recommendation task is how to effectively aggregate member preferences to achieve optimal group consensus. Most of the traditional group recommendation methods use heuristic or attention-based strategies to integrate group preferences, however, these methods still suffer from the following two problems: 1) current neural network-based models ignore the fine-grained higher-order interactions among groups, users, and items; 2) the importance of identifying key users is ignored. To this end, this paper proposes a Key Users Identification-based Heterogeneous Hypergraph for Group Recommendation (UIGRec). We innovatively design two kinds of fine-grained graphs, namely disentangled nested hypergraph and key users heterogeneous graph. We construct a coarse-grained group-level preference graph, which aims to mine the multi-dimensional features of groups comprehensively and deeply. Then, we design a novel hypergraph convolutional network that more precisely captures the group's higher-order interests. Finally, a method to identify key users across inter-groups is proposed, which accurately recognizes key users. The method effectively integrates the important influence of key users on group preferences. Experimental results show that the model proposed in this paper outperforms existing methods in terms of performance on multiple benchmark datasets.

Keywords: Group recommendation · Key users · Graph neural network.

1 Introduction

With the rapid expansion of social networks and the increasing popularity of group activities, more and more people tend to participate in online and offline social group activities. However, traditional recommendation systems are mainly designed for individual users, which are difficult to effectively deal with group decision-making scenarios [1]. Therefore, group recommendation has emerged,

⋆ This work was supported by the National Natural Science Foundation of China (No. 62172243).

which prompts group members to reach consensus by simulating the group decision-making process. For example, a group of travelers can join a travel program on Mafengwo, or a family might decide together which movie to watch.

Earlier research in the area of group recommendation [2, 3, 4] mainly relied on predefined rules and could not dynamically adjust the weights among group members. However, in real scenarios, when a group faces different types of items, the contribution of each member may be different, which makes reaching consensus more complicated. Therefore, it is crucial to design an adaptive strategy to dynamically assign weights to members. With the development of deep learning, graph neural networks show better prospects, and some models [5, 6] incorporate dynamically aggregated neural attention mechanisms to learn better embedding representations. However, most approaches consider each group in isolation when studying group preferences and fail to capture higher-order collaborative relationships across groups. In recent years, hypergraph neural network-based approaches [1, 7] have become popular due to the structural properties of hypergraphs, which are more suitable for group recommendation with multiple members. Although these methods improve the performance of group recommendation, however, they ignore two important issues:

(1) Limitations of coarse-grained group preferences: most existing neural network methods learn group representations through a relational graph of groups, users, and items. However, these methods [1] simply aggregate members and items within groups, ignoring the uniqueness of different semantics and causing information entanglement. It has been shown in [8] that the entangled representation of information in message aggregation impairs performance and leads to redundancy in group preference modeling. At the same time, these methods do not adequately model and encapsulate the higher-order fine-grained interactions of users' preferences for groups.

(2) Identify the importance of key users: there is often a problem of overlapping users between different groups in group recommendation. Previous group recommendation has not paid much attention to the overlapping user problem, while ignoring the fact that overlapping users do not play the same role [9]. Not all overlapping users play key leadership roles in group recommendation scenarios, as illustrated in Fig. 1, Jenny and Nina are overlapping users, however, when Jenny recommends milk to Kelly and Nina's group, it is not popular because young people prefer juice drinks or beer at parties. On the contrary, Nina reaps positive feedback when she recommended the burger and dessert separately to another group. So it seems that Nina's bridging role is more significant and is a key user in social relationships. Her recommendations are more likely to be accepted because the preferences of groups in the same age group are more similar. The key user can give groups access to different information sources and more opportunities to spread novel ideas to different groups, thus influencing group preferences. Nevertheless, existing approaches ignore the importance and advantages of identifying key users.

In order to effectively address the first problem, we design two complex structures, hypergraph and heterogeneous graph, to reveal the fine-grained effects

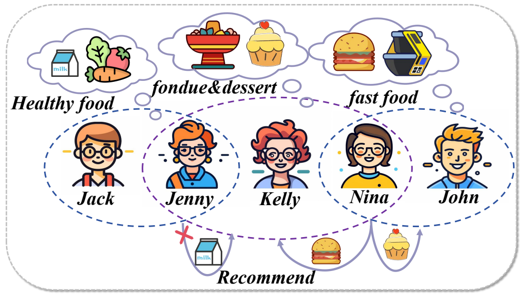

Fig. 1. Example of key users in group recommendations.

between multiple pieces of information. Since the current disentanglement representation learning has been proven to be more effective and resilient in coping with feature complexity [8]. The method aims to learn the explanatory factors behind complex data to improve generalization. Therefore, we design fine-grained encoders that encapsulate the different semantic relationships between groups, items, and users to enhance group representations.

To address the second problem, we introduce the idea of identifying key nodes in complex networks to better model the role of key users. In general, the structure of a complex network relies on a small number of nodes (which we call key nodes) that are able to dominate the evolution of the network itself [10]. Finding nodes with high influence and importance in complex networks is crucial for many realistic scenarios [11]. In group recommendation scenarios, some of the users connected to different groups are key users. By identifying these key users, we can capture more granular and precise potential group preferences. The main contributions of this paper are as follows:

• This paper proposes key users identification-based heterogeneous hypergraph for group recommendation, and designs a fine-grained disentangled nested hypergraph and key users heterogeneous graph to capture the higher-order interactions between groups and users. Meanwhile, the coarse-grained group-level preference graph captures the overall group-level preference more intuitively.

• Innovatively, we propose a method to identify key users across groups. The identified key users are assisted to construct a key users heterogeneous graph, and the group representations are enhanced by fine-grained learning of key users' item preferences, and experiments validate the effectiveness of our approach.

• Extensive experiments on two public datasets demonstrate the superiority of the model UIGRec and the effectiveness of each key design.

2 Related Work

2.1 Identifying Key Nodes

In recent years, the problem of identifying key nodes in complex networks has received a lot of attention. The most important areas in network analysis are the study of social influence [12], link prediction [13] and recommendation systems [14]. Finding nodes with high influence and importance in complex networks is crucial for many realistic scenarios [11]. The identification of key nodes can be seen as a task of ranking nodes based on some centrality metrics. Some of the existing methods for calculating node centrality rely only on local information, while others utilize the topology of the entire network to calculate node centrality. For example, Betweenness Centrality (BC) [15], Closeness Centrality (CC) [16] and Degree Centrality (DC) [17]. Among them, Degree Centrality is easy to use and understand with minimal computational cost. However, since it ignores higher-order neighborhoods, it is usually inaccurate and does not effectively identify important bridging nodes. Structural Holes (SH) [18] is another effective method to identify the key nodes of a network using only local information. However, the SH method has similar drawbacks to the DC method. It may not be able to accurately quantify node differences and effectively identify important bridging nodes. BC global metrics can more accurately identify the strategic locations of key nodes.

2.2 Group Recommendation

Existing methods for group recommendation use different strategies to aggregate members' preferences to form group preferences. Score-based aggregation methods are usually used, including average [3], minimum misery [2] and maximum satisfaction [4]. However, these predefined strategies cannot flexibly reflect multiple intentions among different members. With the development of deep learning, attention-based and graph neural network-based approaches are emerging. For example, Cao et al. [5] adjusted the influence weights among group members through an attention mechanism. Sankar et al. [19] captured the preferences of different individuals in the same group and adaptively weighted them. However, these approaches could not capture the higher-order interactions between groups and users. Recent research on group recommendation techniques has introduced more sophisticated hypergraph and knowledge graph approaches. Jia et al. [1] introduced a hypergraph convolutional network at the member level to capture cross-group collaborative information among users. Zhang et al. [7] designed a hierarchical hypergraph convolutional network to enhance learning. To alleviate the problem of data sparsity, some approaches [20, 21] have integrated contrastive learning into group recommendation to enhance user and group representations. Recently, Wang et al. [21] achieved improved performance by utilizing a hypergraph convolutional network and a collaborative loss computation method based on group and cross-view contrastive learning.

3 Notations and Problem Definitions

We denote the set of users, items and groups as $\mathcal{U} = \{u_1, u_2, ..., u_s, ..., u_M\}$, $\mathcal{I} = \{i_1, i_2, ..., i_j ..., i_N\}$ and $\mathcal{G} = \{g_1, g_2, ..., g_t ..., g_Q\}$, respectively, where M, N, and Q are the sizes of three sets. The t-th group $g_t \in \mathcal{G}$ contains a set of user members

$\mathcal{G}_t$ ={u_1^g,u_2^g,. . . ,u_s^g,. . . ,$u_{|\mathcal{G}_t|}^g$}, where $u_s^g \in \mathcal{U}$ and $|\mathcal{G}_t|$ is the size of $\mathcal{G}_t$. Denote the set of interactions of g_t and u_s respectively as $\mathcal{Y}_t$ ={i_1^g,i_2^g,. . . ,i_j^g,. . . ,$i_{|\mathcal{Y}_t|}^g$} and $\mathcal{R}_s$ ={i_1^u,i_2^u,. . . ,i_j^u,. . . ,$i_{|\mathcal{R}_s|}^u$}, where $|\mathcal{Y}_t|$ and $|\mathcal{R}_s|$ denote the set sizes of $\mathcal{Y}_t$ and $\mathcal{R}_s$, respectively. $\mathrm{Y} \in \mathbb{R}^{Q\times N}$ denotes the group-item interaction matrix, where Y(t, j)=1 means group g_t has interacted with item i_j^g otherwise Y(t, j)=0. Similarly, the user-item interaction matrix is represented by $\mathrm{R} \in \mathbb{R}^{M\times N}$. The set of key users is $\mathcal{U}_h = \{u_1^h, u_2^h, ..., u_s^h, ..., u_{|\mathcal{U}_h|}^h\}$. The process of identifying key users will be detailed in Section 4.1: Key Users Heterogeneous Graph Construction. Given a target group g_t, a list of the most interesting items is recommended to by identifying key users.

4 Methodology

In this section, we focus on the multi-granularity graph encoder and the fusion and prediction module. As shown in Fig. 2, the multi-granularity graph encoder includes: 1) fine-grained graph encoder: disentangled nested hypergraph convolutional network and key users heterogeneous graph convolutional network; 2) coarse-grained graph encoder: preference graph convolutional network.

4.1 Fine-grained Graph Encoder

Disentangled Nested Hypergraph Construction. In this paper, we design a disentangled nested hypergraph to decode the influence of underlying factors of group interest. Formally, the disentangled nested hypergraph is defined as G^D=(V^D, $\mathcal{E}^D$, H^D), where $V^D = \mathcal{U} \cup \mathcal{I}$ denotes the set of vertices, $\mathcal{E}^D$=$\mathcal{G}$ denotes the set of hyperedges, and H^D denotes the association matrix, $H^D = 1$ if the hyperedge contains vertex v, and 0 otherwise. For the t-th group g_t, we will denote e_{g_t} as the t-th hyperedge that connects the corresponding member and item nodes within the group. For example, node $v \in \mathcal{G}_t \cap \mathcal{Y}_t$, where $\mathcal{G}_t$ and $\mathcal{Y}_t$ denote the user members set and interaction set of g_t, respectively, when $H^D(v, e_{g_t}) = 1$, otherwise 0.

Disentangled Nested Hypergraph Convolutional Network. We design novel hypergraph encoders to address the performance degradation caused by entangled information. Specifically, the process of hypergraph convolution is divided into two phases: intra-hyperedge aggregation (node representation aggregate to hyperedges) and inter-hyperedge propagation (hyperedge representations aggregate to nodes), which are defined as $Agg_{n\to e}(\cdot)$ and $Agg_{e\to n}(\cdot)$, respectively. First for user and item different semantic information, we want to keep them distinct and get a decoupled representation of the two kinds of information.

Intra-hyperedge aggregation: By aggregating the information of user u_s^g and item i_j^g within the group g_t respectively, we can intuitively obtain user hyperedge representation $\overline{e}_u$ and item hyperedge representation $\overline{e}_i$:

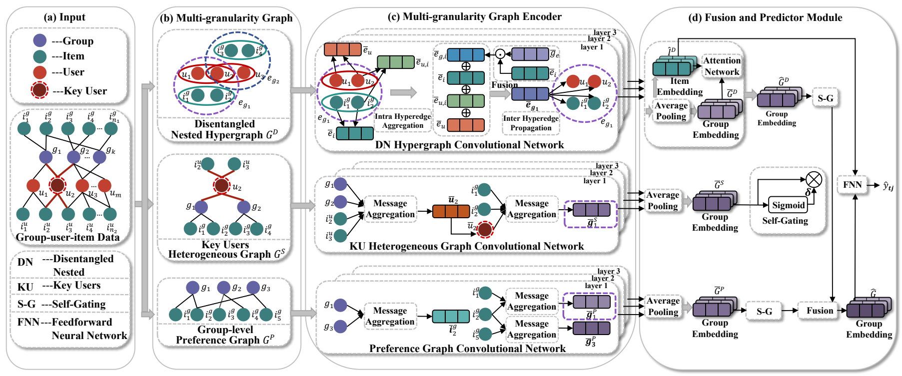

Fig. 2. Overall framework of the UIGRec model.

$$\overline{e}_u = Agg_{n\to e}(\{\overline{u}_s^g | u_s^g \in \mathcal{G}_t\}) \tag{1}$$

$$\overline{e}_i = Agg_{n\to e}(\{\overline{i}_j^g | i_j^g \in \mathcal{Y}_t\}) \tag{2}$$

where $\overline{u}_{\mathrm{s}}^g$ and $\overline{i}_j^g$ denote the embedding representations of user nodes and item nodes within a group, and $Agg_{n\to e}(\cdot)$ denotes the node aggregation function, which is implemented using a simple and efficient average pooling ($Agg_{\boldsymbol{e}\to n}(\cdot)$ as well). To capture the preferences of users u_{s}^g within a group, we aggregate the user preference information to obtain a hyperedge representation $\overline{e}_{u,i}$:

$$\overline{e}_{u,i} = Agg_{n\to e}(\{(\overline{u}_s^g, \overline{i}_j^g) | u_s^g \in \mathcal{G}_t, i_j^g \in \mathcal{Y}_t \cap \mathcal{R}_s\}) \tag{3}$$

where i_j^g represents that the item has been interacted with by both g_{t} and u_{s}^g. Then, we fuse these different representations through a linear transformation:

$$\overline{e}_{g_t} = Agg_g(\mathrm{CONCAT}(\overline{e}_u, \overline{e}_i, \overline{e}_{u,i}, \overline{e}_{g,i})) \tag{4}$$

$$\overline{e}_{g,i} = \overline{g}_e \odot \overline{e}_i \tag{5}$$

where $\overline{e}_{g_{\mathrm{t}}}$ denotes the aggregated hyperedge representation of the group information, to obtain an additional collaborative signaling representation $\overline{e}_{g,i}$ for group preferences, the group's initial embedding $\overline{g}_t$ is element-wise multiplied with the item hyperedge representation $\overline{e}_i$. The $Agg_g(\cdot)$ denotes the learnable weight matrix.

Inter-hyperedge propagation: Since each node u_{s}^g or i_j^g may belong to more than one hyperedge. In this stage, we aggregate all the group information hyperedge representations $\overline{e}_{g_t}$ belonging to the same node to obtain a higher-order collaboration signal between groups. The updated node representations $\widehat{u}_{\mathrm{s}}^{g,(l)}$ and $\widehat{i}_j^{g,(l)}$ from the encoder at layer l can be derived:

$$\widehat{u}_s^{g,(l)} = Agg_{e\to n}(\{\overline{e}_{g_t} | \mathcal{E}^d \in \mathcal{E}_s^d\}) \tag{6}$$

$$\widehat{i}_j^{g,(l)} = Agg_{e \to n}(\{\overline{e}_{g_t} | \mathcal{E}^d \in \mathcal{E}_j^d\}) \tag{7}$$

where $\mathcal{E}^d$ denotes the group information hyperedge, and $\mathcal{E}_s^d$ and $\mathcal{E}_j^d$ denote the set of group information hypergraphs connected to u_s^g nodes and i_j^g nodes respectively. We further stack the multi-layer propagation module for learning and obtain the updated node representation after average pooling operation:

$$\widehat{u}_s^{g,D} = \frac{1}{L+1}\sum_{l=0}^{l}\widehat{u}_s^{g,(l)}, \hat{i}_j^{g,D} = \frac{1}{L+1}\sum_{l=0}^{L}\hat{i}_j^{g,(l)}, \overline{g}_t^D = \frac{1}{L+1}\sum_{l=0}^{L}\overline{e}_{g_t}^{(l)} \tag{8}$$

where L represents the number of convolutional layers, $\overline{e}_{g_t}^{(l)}$ denotes the hyperedge representation of the group information after aggregation in the $l-$th layer, and $\overline{g}_t^D$ denotes the preference representation of the group g_t obtained after the hypergraph encoder. By stacking the representations of all item $\hat{i}_j^{g,D}(j = 1, 2, \cdots N)$ and all group $\overline{g}_t^D(t = 1, 2, \cdots Q)$, the rich item embedding representation $\hat{I}^D$ and the group embedding representation $\overline{G}^D$ are obtained, respectively.

Group Preference Attention Network: Each user in a group does not contribute equally to the group decision. For example, usually when a family decides which comedy movie to watch, one member will be more influential if he/she has watched more than one movie in the comedy genre. Therefore, the attention mechanism is used to give different weights to the users:

$$\widehat{g}_t^D = \overline{g}_t^D + \sum_{u_s^g \in \mathcal{G}_t} \frac{exp(O^T \mathrm{ReLU}(W^D[\hat{u}_s^{g,D} || \hat{i}_j^{g,D}] + b))}{\sum_{u_s^g \in \mathcal{G}_t} exp(O^T \mathrm{ReLU}(W^D[\hat{u}_s^{g,D} || \hat{i}_j^{g,D}] + b))} \widehat{u}_s^{g,D} \tag{9}$$

where $\widehat{u}_s^{g,D}$ and $\widehat{i}_j^{g,D}$ denote the representations after convolution operations of u_s^g and i_j^g, respectively, W^D denotes the weight matrix of the attention network converted to a hidden layer, and b denotes the bias vector. After passing through the ReLU nonlinear activation function, it is projected to the attention score by the weight vector O, and then the normalization operation is performed using the Softmax function. Getting a more accurate group representation $\widehat{g}_t^D$. Finally, the preference representations of all groups $\widehat{g}_t^D$ $(t = 1, 2, \cdots Q)$ are stacked to obtain the updated group representation $\widehat{G}^D$.

Key Users Heterogeneous Graph Construction. At this stage, in order to identify key user u_s^h, we propose a method to identify key users across groups. The specific process can be seen in Fig. 3, firstly, the User-Item Interaction Graph G^{UI} is used as an input to construct the Intra-group Social Graph G^{So} by utilizing the collaborative similarity between users. Since we believe that not all overlapping users are key users, we use the Betweenness Centrality [15] method to calculate the scores of overlapping users in social networks, and call some users with high scores as key users:

$$\mathrm{BC}(v) = \sum_{d \neq v \neq w} \frac{\sigma_{dw}(v)}{\sigma_{dw}} \tag{10}$$

where σ_{dw} denotes the total number of shortest paths from node d to w, and $\sigma_{dw}(v)$ denotes the total number of shortest paths from node d through node v to node w. We denote the set of key users $\mathcal{U}_h = \{u_1^h, u_2^h, ..., u_s^h, ..., u_{|\mathcal{U}_h|}^h\}$, and the set of groups to which the key users belong is denoted as $\mathcal{G}_u = \{g_1^u, g_2^u, ..., g_t^u, ..., g_{|\mathcal{G}_u|}^u\}$. Finally, we take the key user u_s^h as the center node to assist in constructing the key users heterogeneous graph $G^S = (V^S, \mathcal{E}^S, A^S)$. where $V^S = \mathcal{U} \cup \mathcal{I} \cup \mathcal{G}$ denotes the set of nodes; $\mathcal{E}^s = \{(g_t, i_j^g)|g_t \in \mathcal{G}, i_j^g \in \mathcal{I}, \mathrm{Y}(t,j){=}1\} \cup \{(u_s^h, i_j^u)|u_s^h \in \mathcal{U}_h, i_j^u \in \mathcal{I}, \mathrm{R}(s,j){=}1\} \cup \{(u_s^h, g_t^u)|u_s^h \in \mathcal{U}_h, g_t^u \in \mathcal{G}_u, \mathrm{T}(s,t) = 1\}$ denotes the set of edges, A^S is the adjacency matrix.

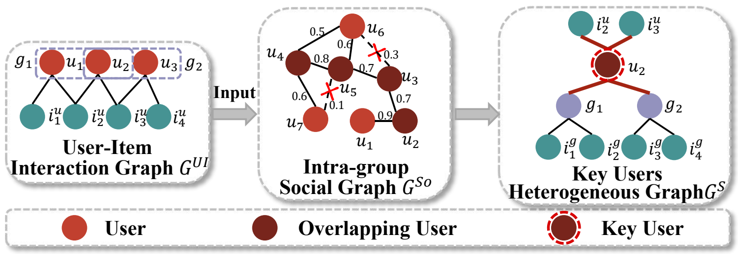

Fig. 3. The process of constructing key users heterogeneous graph.

Key Users Heterogeneous Graph Convolutional Network. We design key users heterogeneous graph G^S to model the importance of key users and capture the impact of key users' historical interactions on group preferences. We splice the group embedding $\overline{G} \in \mathbb{R}^{Q\times d}$, item embedding $\overline{I} \in \mathbb{R}^{N\times d}$, and user embedding $\overline{U} \in \mathbb{R}^{M\times d}$ to to obtain $B^{(0)} = [\overline{G}\ \overline{I}\ \overline{U}]^T$. In the message aggregation phase of each layer of the graph convolutional network, the node representation update process is as follows:

$$\overline{u}_s^h = Agg(\{(\overline{g}_t, \overline{i}_j^{u_s^h})|g_t \in \mathcal{G}_u, i_j^{u_s^h} \in \mathcal{R}_s\}) \tag{11}$$

$$\overline{g}_t^S = Agg(\{(\overline{u}_s^h, \overline{i}_j^{g_t})|u_h \in \mathcal{U}_h, i_j^{g_t} \in \mathcal{Y}_t\}) \tag{12}$$

where $\overline{g}_t$ and $\overline{i}_j^{u_s^h}$ denote the initial embedding of group g_t and item $i_j^{u_s^h}$, respectively. The $\overline{u}_{\mathrm{s}}^h$ and $\overline{i}_j^{g_t}$ denote the initial embeddings of user u_{s}^h and the item $i_j^{g_t}$ that has been interacted with by group g_t, respectively. During the aggregation process, as shown in Fig. 2, the updated representation $\overline{u}_2$ is aggregated into the first-order neighbor g_1 to obtain a richer and more accurate implicit preference representation $\overline{g}_1^S$. In the aggregation function $Agg(\cdot)$, We average the l-th layer feature representation of each node with the feature vectors of its local neighborhood. Referring to previous work on graph convolution networks [22], more succinctly, we denote it by the following equation:

$$B^{(l+1)} = D^{-\frac{1}{2}} A^S D^{-\frac{1}{2}} B^{(l)} \tag{13}$$

where A^S is the adjacency matrix of the key users heterogeneous graph G^S. $B^{(l)}$ denotes the node representation of the $l-$th layer, we then pass each layer of embedding through an average pooling operation to obtain the final embedding representation $\overline{B}$:

$$\overline{B} = \frac{1}{L+1} \sum_{l=0}^{L} B^{(l)} = [\overline{G}^S \quad \overline{I}^S \quad \overline{U}^S]^T \tag{14}$$

At this stage, we obtain the group embedding representation $\overline{G}^S$ after key users heterogeneous graph convolutional network, which implicitly aggregates the information of the key user interaction items and injects it into the group representation, and the resulting $\overline{G}^S$ representation can reflect underlying group preferences.

4.2 Coarse-grained Graph Encoder

Group-level Preference Graph Construction. Construct a group-level preference graph $G^P = (V^P, \mathcal{E}^P, A^P)$ to capture the overall preference of items at the group-level, $V^P = \mathcal{G} \cup \mathcal{I}$ to denote the set of nodes, $\mathcal{E}^P$ to denote the set of edges $\mathcal{G}^P = \{(g_t, i_j^g) | g_t \in \mathcal{G}, i_j^g \in \mathcal{I}, Y(t,j) = 1\}$, and $A^P \in \mathbb{R}^{(Q+N)\times(Q+N)}$ is the adjacency matrix.

Preference Graph Convolutional Network. We concatenate the group embedding $\overline{G} \in \mathbb{R}^{Q\times d}$ with the item embedding $\bar{I} \in \mathbb{R}^{N\times d}$ to obtain $C^{(0)} = \begin{bmatrix} \overline{G} \\ \bar{I} \end{bmatrix}$, which serves as the input to the first layer of the graph convolutional network. The propagation mechanism of each layer is similar to Eq. (13), and then the average pooling operation is expressed as follows:

$$\overline{C} = \frac{1}{L+1} \sum_{l=0}^{L} C^{(l)} = \begin{bmatrix} \overline{G}^P \\ \overline{I}^P \end{bmatrix} \tag{15}$$

after passing through L convolutional layers, we obtain the final group-level preference graph embedding representation $\overline{G}^P$.

4.3 Fusion and Predictor Module

Next we explicitly mine group intentions and preferences by fusing graphs from multiple different granularities. Common fusion approaches are element summing or concatenation operations, but they ignore the different importance of different granularity graphs. To address this limitation, we propose an adaptive fusion approach that fuses three view-specific group representations by designing three different gates:

$$\gamma = \sigma(\widehat{G}^D W^D), \delta = \sigma(\overline{G}^S W^S), \vartheta = \sigma(\overline{G}^P W^P) \tag{16}$$

$$\widehat{G} = \gamma \widehat{G}^D + \delta \overline{G}^S + \vartheta \overline{G}^P \tag{17}$$

where W^D, W^S and W^P denote the different trainable weight matrices, and γ, δ and ϑ are the learning weights. σ is the Sigmoid activation function. Our model automatically balances the contributions of disentangled nested hypergraph, key users heterogeneous graph and group-level preference graph. Then, based on the group embedding $\widehat{G}$ obtained by dynamic preference aggregation and the item $\hat{I}^D$ obtained by hypergraph convolution, we input the embedding representations of the corresponding group and item interaction pairs (g_t, i_j) into a Feedforward Neural Network (MLP) to compute the final predicted score $\hat{y}_{tj}$:

$$\hat{y}_{tj} = \mathrm{MLP}(\widehat{g}_t \odot \hat{i}_j) \tag{18}$$

for group and item interaction data, we optimize using Bayesian Personalized Ranking (BPR) loss as follows:

$$\mathcal{L}_{group} = -\sum_{g_t \in \mathcal{G}} \frac{1}{|\mathcal{D}_{g_t}|} \Sigma_{(j^+, j^-) \in \mathcal{D}_{g_t}} \ln \sigma(\hat{y}_{tj^+} - \hat{y}_{tj^-}) \tag{19}$$

where $\mathcal{D}_{g_t}$ denotes the group-item dataset for group g_t, where each instance is a pair of (j^+, j^-), denoting that group g_t interacts with item i_{j^+}, but does not interact with item i_j-. To further exploit the supervised signals, we optimize the group recommendation task by combining user-item interaction data. Similarly, the user's prediction score for item is computed by $\overline{z}_{sj}$ =MLP($\overline{u}_{\mathrm{s}} \odot \overline{i}_j$), where $\overline{u}_{\mathrm{s}}$ and $\overline{i}_j$ are the initial embeddings of the user and item, respectively. Finally, the loss function is optimized using into:

$$\mathcal{L}_{user} = -\sum_{u_s \in \mathcal{U}} \frac{1}{|\mathcal{D}_{u_s}|} \Sigma_{(j^+, j^-) \in \mathcal{D}_{u_s}} \mathrm{In}\, \sigma(\overline{z}_{sj^+} - \overline{z}_{sj^-}) \tag{20}$$

where $\mathcal{D}_{u_{\mathrm{s}}}$ is denoted as the user-item dataset for user u_{s}, the recommendation performance is improved by jointly training $\mathcal{L}_{group}$ and $\mathcal{L}_{user}$.

5 Experiment

In this section, we validate the effectiveness of the UIGRec model through extensive experiments, which are designed to answer the following research questions:

• RQ1: How does our group recommendation model UIGRec perform compared to state-of-the-art group recommendation models?

• RQ2: How do our key components work?

• RQ3: How identified key users affect recommendation performance?

5.1 Experimental Settings

Datasets. We conduct experiments on two publicly available datasets Mafengwo and MovieLens. Among them, Mafengwo[5] is a travel website where users can create or join tours of interest by themselves. MovieLens refers to previous work [23], and the statistical information of the datasets is shown in Table 1.

Table 1. Statistical information of the dataset.

Dataset	Mafengwo	MovieLens
Users	5275	943
Groups	995	5000
Items	1513	1682
User–item interactions	39,765	100,000
group-item interactions	3595	92,885
User–Item sparsity	0.9950	0.9370
Group-Item sparsity	0.9976	0.9890

Baselines. To evaluate our proposed framework UIGRec, comparisons were made with six other baselines:

(1) AGREE [5]: adopts an attentional mechanism adapted to group representations.

(2) GroupIM [19]: utilizes mutual information to dynamically learn user and group representations in order to capture the preferences of different users.

(3) HCR [1]: proposes a dual-channel hypergraph convolutional network to capture cross-group collaborative information from users.

(4) S^2-HHGR [7]: developed a hierarchical hypergraph convolutional network to alleviate the data sparsity problem.

(5) CubeRec [20]: utilized the geometric representation of hypercubes to build self-supervised signals.

(6) HGRec [21]: utilized hypergraph convolutional networks and a collaborative loss computation method based on learning from group and cross-view contrastive learning.

Evaluation Metrics. To evaluate the accuracy of recommendations, we apply two metrics widely used for evaluating recommender systems, namely Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG), denoted by HR@K and NDCG@K, respectively, where K $\in \{5,10\}$. HR measures whether a test item is ranked in the top K lists, and NDCG indicates the position of a hit in a list by assigning a higher score. Where the larger the HR and NDCG values, the better the model performance.

Implementation. We conducted experiments based on NVIDIA GeForce RTX 2080Ti and implemented our proposed method on PyTorch using Adam optimizer. We empirically set the initial embedding dimensions for groups, users, and items in Mafengwo and MovieLens to 32. The learning rates are set to 0.0001 and 0.0005, respectively.

5.2 Overall Performance Comparison (RQ1)

The performance of UIGRec is compared with all baselines. The experimental results are shown in Table 2 and we observe the following results.

• Our proposed method outperforms all baselines in all two datasets. Recommendation performance is significantly improved in MovieLens, probably due to the fact that groups and users in MovieLens have similar preferences and interactions for popular items. Our proposed UIGRec captures fine-grained higher-order interactions of users and groups, especially the important influence of key users on group preferences.

• In all baselines, both AGREE and GroupIM use an attention mechanism to dynamically capture the different weights among group members. However, these two approaches fail to model the complex relationships between groups. In contrast, HCR, S^2-HHGR, and HGRec consider higher-order interactions between groups through hypergraph modeling. In addition, S^2-HHGR, HGRec and CubeRec alleviate the problem of sparse data due to the introduction of Contrastive learning objectives. However, our approach captures finer group representations by designing a multi-granularity graph encoder and by identifying key users, which makes our proposed approach more superior.

Table 2. Performance comparison of methods for group recommendation task.

Dataset	Metric	AGREE	GroupIM	HCR	S^2-HHGR	CubeRec	HGRec	**UIGRec**
Mafengwo	HR@5	0.73869	0.73668	0.75778	0.75870	0.86231	0.84623	**0.87337**
	HR@10	0.77588	0.81106	0.82412	0.78630	0.90251	0.86633	**0.90352**
	NDCG@5	0.64151	0.61348	0.65771	0.73120	0.75749	0.74241	**0.77927**
	NDCG@10	0.65374	0.63611	0.67875	0.72540	0.77167	0.74989	**0.78902**
MovieLens	HR@5	0.74211	0.71029	0.70770	0.74230	0.73804	0.77464	**0.80096**
	HR@10	0.84163	0.82943	0.80840	0.83158	0.86244	0.88588	**0.89522**
	NDCG@5	0.56795	0.51193	0.51900	0.51745	0.55025	0.59778	**0.62921**
	NDCG@10	0.60036	0.55062	0.55140	0.43019	0.58674	0.63201	**0.66350**

5.3 Ablation Study (RQ2)

In order to assess the contribution of the three graphs, we created the following variants: (1) w/o. Hy: remove the disentangled nested hypergraph; (2) w/o. He: remove the key users heterogeneous graph; (3) w/o. Per: remove the group-level preference graph. In each of these three variants, one graph was removed at a time for comparison with UIGRec. From Table 3, we observe that removing graphs either way leads to a decrease in model performance, indicating that all three graphs play a key role in the model. In the two datasets, the performance degradation is most obvious after removing the disentangled nested hypergraph (w/o. Hy), proving that constructing the hypergraph and using the encoder designed in this paper can improve the model's expressive power more effectively. Secondly, the performance of removing the key users heterogeneous graph (w/o. He) is also degraded, indicating that the removal of this graph fails to capture the leadership role of key users in the group as well as the latent preferences of group preferences.

Table 3. Ablation experiment.

Dataset	Metric	w/o. Hy	w/o. He	w/o. Per	**UIGRec**
Mafengwo	HR@5	0.80704	0.87238	0.86633	**0.87337**
	HR@10	0.88342	0.89749	0.89749	**0.90352**
	NDCG@5	0.63995	0.77860	0.77003	**0.77927**
	NDCG@10	0.66622	0.78629	0.77975	**0.78902**
MovieLens	HR@5	0.78589	0.79569	0.77703	**0.80096**
	HR@10	0.88660	0.89258	0.88636	**0.89522**
	NDCG@5	0.62145	0.62355	0.59721	**0.62921**
	NDCG@10	0.65067	0.65633	0.63294	**0.66350**

5.4 Research on Key Users (RQ3)

This subsection provides an in-depth study on the importance of identifying key users by conducting experiments using {20%, 40%, 60%, 80%, 100%} of all overlapping user numbers as the number of key users r. From Fig. 4, it can be seen that in Mafengwo, probably its data is sparse and the number of users that play a key role is small, and also, combined with Fig. 5, it can be observed that only a very small number of nodes with BC scores greater than 0.1 play a leadership role across groups. So the best performance of the recommendation is achieved when 20% is chosen. Overlapping users in MovieLens account for a larger percentage compared to the number of all users, while the corresponding number of critical users may be larger. In Fig. 5, it can be seen that most nodes have less direct connections with other nodes in the network, and the BC scores are generally lower, probably because most people have more distinct preferences for watching movies. The observation from Fig. 4 but not the selection of all overlapping users (100%) leads to the best results, verifying the effectiveness of the key user identification strategy, and that accessing the preferences of key users enables better modeling of group preferences.

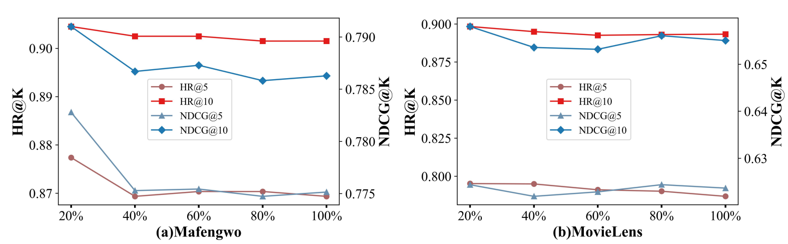

Fig. 4. Impact of selecting different numbers of key users on performance under the Mafengwo and MovieLens datasets.

To further validate the accuracy of the identified key users for modeling group preferences, we compared the heterogeneous graphs constructed by the key users

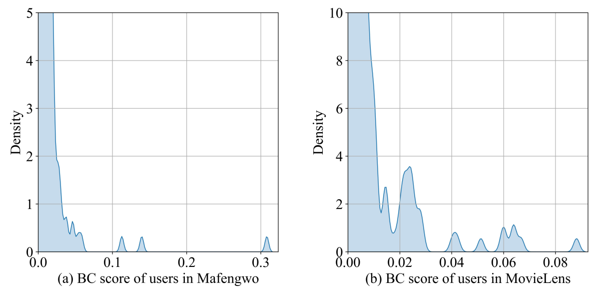

Fig. 5. The concentration of BC score of all overlapping users in different value ranges is shown by Kernel Density Estimate.

with the heterogeneous graphs constructed by all users per group. From Fig. 6, it is observed that the heterogeneous graph formed by key users performs better. This may be due to the fact that in selecting all users' item preferences to augment group preferences, it leads to information redundancy and corrupts the recommendation performance. It is worth noting that the results of the NDCG@10 metric in the Mafengwo dataset are slightly worse, and we infer that the dataset may be too sparse, and we need to aggregate group preferences with more user information, however, the overall experimental results show that the key users we identify can effectively improve the performance.

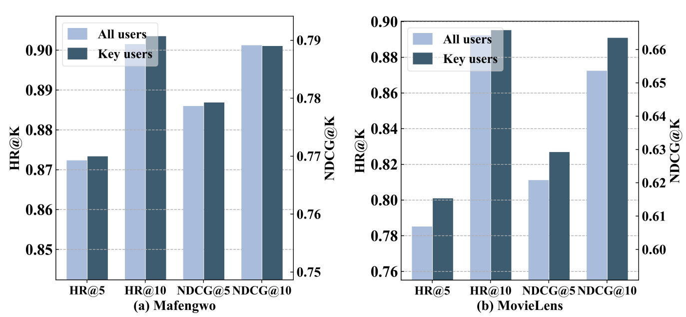

Fig. 6. Performance impact of selecting key users under the Mafengwo and MovieLens datasets.

6 Conclusion

In this paper, we propose key users identification-based heterogeneous hypergraph for group recommendation. where the fine-grained graph includes disentangled nested hypergraph and key users heterogeneous graph, which aims to reveal the higher-order interactions between groups and users. The preference extraction of the model is further enhanced by the coarse-grained graph group-level preference graph. The model effectively overcomes the limitations existing in the current group recommendation task and optimizes the aggregation process of different semantic information in groups. In addition, a method to recognize key users across groups is proposed for group recommendation. The model in this paper shows excellent performance on two datasets. In the future we will endeavor to explore dynamic hypergraphs, integrate interpretability into key user identification.

References

[1] Renqi Jia et al. "Hypergraph convolutional network for group recommendation". In: *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2021, pp. 260–269.

[2] Sihem Amer-Yahia et al. "Group recommendation: Semantics and efficiency". In: *Proceedings of the VLDB Endowment* 2.1 (2009), pp. 754–765.

[3] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. "Group recommendations with rank aggregation and collaborative filtering". In: *Proceedings of the fourth ACM conference on Recommender systems*. 2010, pp. 119–126.

[4] Ludovico Boratto and Salvatore 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, 2011, pp. 1–20.

[5] Da Cao et al. "Attentive group recommendation". In: *The 41st International ACM SIGIR conference on research & development in information retrieval*. 2018, pp. 645–654.

[6] Lei Guo et al. "Group recommendation with latent voting mechanism". In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE. 2020, pp. 121–132.

[7] Junwei Zhang et al. "Double-scale self-supervised hypergraph learning for group recommendation". In: *Proceedings of the 30th ACM international conference on information & knowledge management*. 2021, pp. 2557–2567.

[8] Jianxin Ma et al. "Disentangled graph convolutional networks". In: *International conference on machine learning*. PMLR. 2019, pp. 4212–4221.

[9] Tiancheng Lou and Jie Tang. "Mining structural hole spanners through information diffusion in social networks". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 825–836.

[10] Ning Wang et al. "Robustness evaluation of the air cargo network considering node importance and attack cost". In: *Reliability Engineering & System Safety* 217 (2022), p. 108026.

[11] Zahra Aghaee et al. "A survey on meta-heuristic algorithms for the influence maximization problem in the social networks". In: *Computing* 103 (2021), pp. 2437–2477.

[12] Hamid Ahmadi Beni et al. “A fast module identification and filtering approach for influence maximization problem in social networks”. In: *Information Sciences* 640 (2023), p. 119105.

[13] Huan Wang et al. “A multi-type transferable method for missing link prediction in heterogeneous social networks”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.11 (2023), pp. 10981–10991.

[14] Zahra Noshad, Asgarali Bouyer, and Mohammad Noshad. “Mutual information-based recommender system using autoencoder”. In: *Applied Soft Computing* 109 (2021), p. 107547.

[15] LC Freeman. “A set of measures of centrality based on betweenness”. In: *Sociometry* (1977).

[16] Gert Sabidussi. “The centrality index of a graph”. In: *Psychometrika* 31.4 (1966), pp. 581–603.

[17] Réka Albert and Albert-László Barabási. “Statistical mechanics of complex networks”. In: *Reviews of modern physics* 74.1 (2002), p. 47.

[18] Ronald S.. Burt. *Structural holes: The social structure of competition.* Harvard university press, 1992.

[19] Aravind Sankar et al. “Groupim: A mutual information maximization framework for neural group recommendation”. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval.* 2020, pp. 1279–1288.

[20] Tong Chen et al. “Thinking inside the box: learning hypercube representations for group recommendation”. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2022, pp. 1664–1673.

[21] Nan Wang et al. “HGRec: Group Recommendation With Hypergraph Convolutional Networks”. In: *IEEE Transactions on Computational Social Systems* (2024).

[22] Felix Wu et al. “Simplifying graph convolutional networks”. In: *International conference on machine learning.* PMLR. 2019, pp. 6861–6871.

[23] Yingqi Zhao et al. “DHMAE: A Disentangled Hypergraph Masked Autoencoder for Group Recommendation”. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2024, pp. 914–923.