

# Group-Aware Interest Disentangled Dual-Training for Personalized Recommendation

Xiaolong Liu<sup>1</sup>, Liangwei Yang<sup>1</sup>, Zhiwei Liu<sup>2</sup>, Xiaohan Li<sup>3</sup>, Mingdai Yang<sup>1</sup>, Chen Wang<sup>1</sup>, Philip S. Yu<sup>1</sup>

<sup>1</sup>University of Illinois Chicago, Chicago, IL, USA

{xliu262, lyang84, myang72, cwang266, psyu}@uic.edu

<sup>2</sup>Salesforce AI Research, Palo Alto, USA

zhiwei.liu@salesforce.com

<sup>3</sup>Walmart Global Tech, Sunnyvale, CA, USA

xiaohan.li@walmart.com

**Abstract**—Personalized recommender systems aim to predict users’ preferences for items. It has become an indispensable part of online services. Online social platforms enable users to form groups based on their common interests. The users’ group participation on social platforms reveals their interests and can be utilized as side information to mitigate the data sparsity and cold-start problem in recommender systems. Users join different groups out of different interests. In this paper, we generate group representation from the user’s interests and propose **IGRec** (Interest-based Group enhanced Recommendation) to utilize the group information accurately. It consists of four modules, (1) Interest disentangler via self-gating that disentangles users’ interests from their initial embedding representation. (2) Interest aggregator that generates the interest-based group representation by Gumbel-Softmax aggregation on the group members’ interests. (3) Interest-based group aggregation that fuses user’s representation with the participated group representation. (4) A dual-trained rating prediction module to utilize both user-item and group-item interactions. We conduct extensive experiments on three publicly available datasets. Results show **IGRec** can effectively alleviate the data sparsity problem and enhance the recommender system with interest-based group representation. Experiments on the group recommendation task further show the informativeness of interest-based group representation.

**Index Terms**—Recommender System, Personalization, Group Recommendation, Graph Neural Network

## I. INTRODUCTION

Recommender systems (RS) [1]–[3] are becoming indispensable to web applications owing to their prominent ability in user retention [4] and commercial conversion [5]. Data sparsity and cold-start problems [6]–[8] are still obstacles that most RS suffer from. Data sparsity indicates users can only consume an extremely small proportion of all items on the platform. The cold-start problem with long-tail and incoming users is more severe for the bare information we can acquire. Presently, several social platforms allow users to join groups out of their shared interests in items. Users’ group participation is a direct signal to reflect their interests in items, which is informative side information to alleviate the above two obstacles [9]–[11] in RS.

Personalized RS [12]–[16] aims to recommend unique items for each user based on his/her behaviors and profile. The group participation of users on social platforms is personalized behavior that can reflect his/her interests in items. An

illustrative example from the Steam platform <sup>1</sup> is shown in Figure 1: Alice, Aaron, and Tiffany are keen on the action game *Hollow Knight*, and therefore they form a group to discuss the game. It explicitly reflects members’ interest in action games. A user can join multiple groups for different interests in items. For example, Tiffany not only plays the action game *Hollow Knight* but also engages in two other offline games *Portal* and *Don’t Starve*. On those platforms, groups can also host activities for group members based on common interests and have direct interaction with items. In Figure 1, competitive groups can host online meetups for the “Dota2” game competition. The availability of users’ group participation and group activities motivates industrial and academic researchers to utilize it as side information to enrich user embedding [17]–[20]. These models assume users who participate in the same group share the same preferences. Hence, the user representations are not only learned from interacted items but also from other group members.

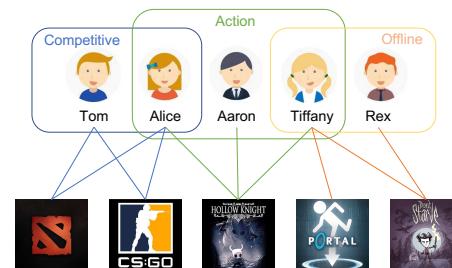


Fig. 1: A toy example of the groups of users on Steam Platform. In this example, 5 users play 5 kinds of games. Users form “Competitive”, “Action” and “Offline” groups based on their gaming preferences.

Though utilizing group information in RS has attracted academic attention, several key challenges are less explored in effectively exploiting group information: **(1) Multiple user interests.** Users join multiple groups out of different interests, which are latent and entangled in initial user embedding. For example, in Figure 1, Alice joins the ‘competitive’ group

<sup>1</sup><https://store.steampowered.com/>

because she likes competing with others online in the game CS:GO, while she also joins the ‘action’ group since she enjoys controlling action games. Disentangling and characterizing users’ interests in joining different groups enables a more accurate interest-based group representation. **(2) Heterogeneous group members.** Previous works [19], [20] directly aggregate the embedding of its members into group representation, which intrinsically assumes that members in the same group share homogeneous interests. However, it is far from real-world scenarios. Users in the same group only indicate their interests are partially overlapped. Ignoring the heterogeneity of group members within the group would misconstrue the information of this group, thus greatly impairing the quality of a group representation. **(3) Effective group representation for personalized RS.** The availability of group activities motivates previous research [19]–[23] to focus on recommending items to a whole group, *i.e.*, recommending the same item to a whole group of users rather than personalized RS. How to effectively utilize group activities for personalized RS is still under-explored.

In this paper, we propose a novel framework **IGRec** (**I**nterest-based **G**roup enhanced **R**ecommendation) to tackle the aforementioned research problems. **IGRec** is designed to enhance personalized RS from users’ group participation behavior and group activities. As Graph Neural Networks (GNNs) have achieved wide success in RS [24]–[30], the **IGRec** is built upon a GNN to learn node embeddings over a multi-interest heterogeneous graph that contains user, item, interest, and group nodes. **IGRec** consists of four modules. **(1) Interest Disentangler** ingests user embeddings effectively for characterizing the crucial interests of users. It identifies users’ interests in joining different groups to address the first challenge of multiple user interests. Specifically, we employ a self-gating unit for each type of latent interest, which is trained to extract users’ interests from initial embedding. **(2) Interest Aggregator** infers the embeddings of those interests aggregated in groups. This interest aggregator passes those disentangled interests of users and learns interest-based group representation from all group members via a novel Gumbel-Softmax aggregation to address the second challenge of heterogeneous group members on group representation. Then, since **IGRec** is built upon the GNN backbone, we stack multiple layers to update the user embeddings by aggregating those interest-based group embeddings from their engaged groups. In analogy, the item embeddings are also updated by aggregating all connected users’ embeddings. **(3) Interest-based group aggregation** addresses the third challenge on utilizing group representation to enhance user representation. **(4) Dual-trained rating prediction** module unifies the training of both user recommendation and group recommendation tasks. The two tasks train the shared interest disentangler and interest aggregator modules. The dual-trained **IGRec** can achieve state-of-the-art performance on both personalized and group recommendation tasks. **IGRec** is open-sourced at <https://github.com/Xiaolong-Liu-bdsc/IGRec>. In summary, the key contributions of this paper are as follows:

- This is the first work that characterizes users’ interests in items via group participation, identifies the challenges and proposes the solution of interest-based group representation.
- We propose the interest disentangler via self-gating modules to retrieve multiple interests from user embeddings and a novel interest aggregator to infer interest embeddings.
- We devise **IGRec** based on interest disentanglement. It achieves state-of-the-art performance on both personalized and group recommendation tasks.
- To verify the effectiveness, we conduct extensive experiments on two publicly available benchmark datasets MaFengWo and Douban, and extract and publish a ready-to-use group recommendation dataset containing real-world group activity created from the Steam platform <sup>2</sup>.

In the following sections, we first present preliminaries in Section II, including task formulation, multi-interest heterogeneous graph, and graph neural network. **IGRec** is illustrated in Section III. Experiment results, including model performance, ablation study, and parameter sensitivity, are shown in Section IV. Related works in the group-enhanced recommendation and multi-interest network embedding are introduced in Section V. Finally, the conclusion and future research problems are given in Section VI.

## II. PRELIMINARY

In this section, we first formulate the group-enhanced personalized recommendation task, and then we illustrate the multi-interest heterogeneous graph proposed in this paper.

### A. Task Formulation

We have a set of users  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ , a set of items  $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$  and a set of groups  $\mathcal{G} = \{g_1, g_2, \dots, g_{|\mathcal{G}|}\}$ . The interactions of users, items, and groups form three graphs: user-item graph  $G_{UV}$ , group-item graph  $G_{GV}$ , and group-user graph  $G_{GU}$ . We let  $\mathbf{R}_u \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{V}|}$  and  $\mathbf{R}_g \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{V}|}$  denote user-item interaction matrix and group-item interaction matrix respectively, where  $r_{uv}$  (or  $r_{gv}$ ) = 1 if user  $u$  (or group  $g$ ) purchased item  $v$ , otherwise  $r_{uv}$  (or  $r_{gv}$ ) = 0. Similarly,  $\mathbf{S} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{U}|}$  represents group-user interaction matrix, where  $s_{gu} = 1$  if group  $g$  contains user  $u$ , otherwise  $s_{gu} = 0$ . We define  $U(g)$  to denote the set of users in group  $g$ . The primary objective is to predict how likely a user  $u$  would adopt item  $v$  based on historical user-item interactions and the group information.

### B. Multi-interest Heterogeneous Graph

To integrate user-item graph  $G_{UV}$ , group-item graph  $G_{GV}$ , and group-user graph  $G_{GU}$  into a unified graph representation, we propose the multi-interest heterogeneous graph. **IGRec** introduces a set of interests  $\mathcal{I} = \{\mathcal{I}^1, \mathcal{I}^2, \dots, \mathcal{I}^M\}$  where  $M$  is the number of interests.  $\mathcal{I}^n = \{i_{u_1}^n, i_{u_2}^n, \dots, i_{u_{|\mathcal{U}|}}^n\}$  is the set of all users’  $n$ -th interest,  $n \leq M$ .

**Definition II.1. Multi-interest Heterogeneous Graph.** The multi-interest heterogeneous graph is defined as  $G = (V, E)$

<sup>2</sup>To be released upon acceptance

where  $V = (\mathcal{U}, \mathcal{V}, \mathcal{G}, \mathcal{I})$  denotes the node set and  $E = (E_{uv}, E_{ui}, E_{gi}, E_{gv})$  denotes the edge set.  $\mathcal{U}$ ,  $\mathcal{V}$ ,  $\mathcal{G}$ , and  $\mathcal{I}$  represent the set of users, items, groups, and interests.  $E_{uv}$ ,  $E_{ui}$ ,  $E_{gi}$  and  $E_{gv}$  denote the user-item edges, user-interest edges, group-interest edges, and group-item edges, respectively.

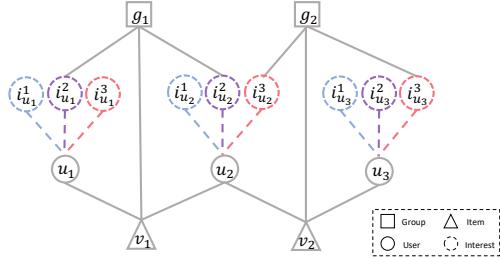


Fig. 2: Multi-Interest Heterogeneous Graph.

The defined multi-interest heterogeneous graph is shown in Fig. 2. It can be split into four parts based on different types of edges. The first part is a user-item ( $u-v$ ) bipartite graph representing user-item interaction. An edge between  $u$  and  $v$  indicates the user  $u$  has an interaction with the item  $v$ . The second part is a tree-structured user-interest ( $u-i$ ) graph. Users' interests latently existed in the data, and IGRec disentangles the interests as the dashed interest nodes ( $i$ ) in Fig. 2. Each user node is connected with a fixed number of interest nodes representing the user's interests. The third part is a group-interest ( $g-i$ ) graph. It is also a tree-structured graph. Each group node has edges with one or more specific interest nodes of its group members. The last part is a group-item ( $g-i$ ) graph that encodes group-item interactions.

### III. THE PROPOSED MODEL: IGRec

The proposed **IGRec** is shown in Figure 3, which consists of Interest Disentangler, Interest Aggregator, Interest-based Group Aggregation, and Dual-trained Rating Predictor.

#### A. Interest Disentangler

The core idea is that we assume users participate in different groups out of different interests in items, which is to solve the **multiple user interests** challenge. For example, in Figure 1, Alice joins the "Competitive" group since she enjoys the competitive game CS:GO and joins the "Action" group because of her interest in the action game Hollow Night. Moreover, these interests should follow the user's overall representation. IGRec utilizes *self-gating unit* [31] to disentangle user's interests from the user embedding. It computes a threshold on each dimension as a filter to extract the user's interests.

Given a user node  $u$  and its embedding  $\mathbf{e}_u$ , interest disentangler aims to disentangle  $\mathbf{e}_u$  into  $M$  interests  $\mathbf{i}_u^1, \mathbf{i}_u^2, \dots$  and  $\mathbf{i}_u^M$ . In previous works [32]–[34], interests are generated randomly or learned from different interaction graphs. In this paper, we use multiple *self-gating units* to disentangle user embedding into different interests. By passing  $\mathbf{e}_u$  through these units and

then each unit generates one interest representation for this user  $u$ , which is formulated as:

$$\mathbf{i}_u^n = f_{\text{gate}}^n(\mathbf{e}_u) = \mathbf{e}_u \odot \sigma(\mathbf{e}_u \mathbf{W}^n + \mathbf{b}^n), \quad (1)$$

where  $\mathbf{i}_u^n$  denotes  $u$ 's  $n$ -th interest vector.  $f_{\text{gate}}^n(\cdot)$  is the self-gating function that extracts  $u$ 's  $n$ -th interest.  $\mathbf{W}^n \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b}^n \in \mathbb{R}^d$  are trainable parameters.  $d$  is the embedding size.  $\odot$  denotes the element-wise product and  $\sigma$  is the sigmoid function. The left part of Fig. 3 illustrates the interest disentangler.

#### B. Interest Aggregator

After obtaining the interests of each user, we next introduce the interest aggregator of each group  $g$  to learn from **heterogeneous group members**, as shown in the middle part of Fig. 3. We argue that a group is formed based on the user's specific interests, and IGRec utilizes an interest aggregator for each group to select and aggregate the relevant interests. Different from previous works [11], [35], [36] that obtain the group representation by directly aggregating group members' overall embeddings, IGRec learns interest-based group representation by aggregating its members' selected interests. The probability of interest selection given memberships is calculated by the softmax formula as follows:

$$p(\delta(g) = n | U(g), g) = \frac{\exp(\mathbf{e}_g \cdot \text{Readout}(\{\mathbf{i}_u^n\}_{u \in U(g)}))}{\sum_{j=1}^M \exp(\mathbf{e}_g \cdot \text{Readout}(\{\mathbf{i}_u^j\}_{u \in U(g)}))}. \quad (2)$$

To simplify the notation, we drop the subscript on group, where  $\mathbf{e}_g$  represents group embedding for group  $g$ .  $\cdot$  denotes the dot product.  $\delta(g)$  is the currently selected interest for group node  $g$ . Readout function:  $\mathbb{R}^{|\{\mathbf{i}_u\}_{u \in U(g)}| \times d} \rightarrow \mathbb{R}^d$  is applied to aggregate group member's common interest information of all members within the group.

1) *Readout Function*: The readout function plays an important role in the message aggregating phase in GNNs. It summarizes an arbitrary number of embedding vectors into a fixed vector representation. Mean, Max, and Sum are the most common readout functions. We utilize Attention-based pooling as it is explainable that the interest of each user node contributes unequally to this group. Moreover, it also achieves the best performance in our experiments.

$$\gamma_{\mathbf{i}_u^n} = \frac{\exp(\mathbf{W}_{\text{Att}} \mathbf{i}_u^n)}{\sum_{j \in U(g)} \exp(\mathbf{W}_{\text{Att}} \mathbf{i}_j^n)}, \quad (3)$$

$$\mathbf{i}^n = \text{Readout}(\{\mathbf{i}_u^n\}_{u \in U(g)}) = \sum_{j \in U(g)} \gamma_{\mathbf{i}_j^n} \mathbf{i}_j^n, \quad (4)$$

where  $\mathbf{W}_{\text{Att}} \in \mathbb{R}^{1 \times d}$  is the trainable attention parameters.  $\gamma_{\mathbf{i}_u^n}$  is calculated by the softmax formula and represents the learned weight of  $\mathbf{i}_u$  in group  $g$  with respect to the  $n$ -th interest. The readout function outputs the weighted sum of members'  $n$ -th interest vectors.  $\mathbf{i} = (\mathbf{i}^1, \mathbf{i}^2, \dots, \mathbf{i}^M) \in \mathbb{R}^{d \times M}$  denotes the aggregated interest embedding.

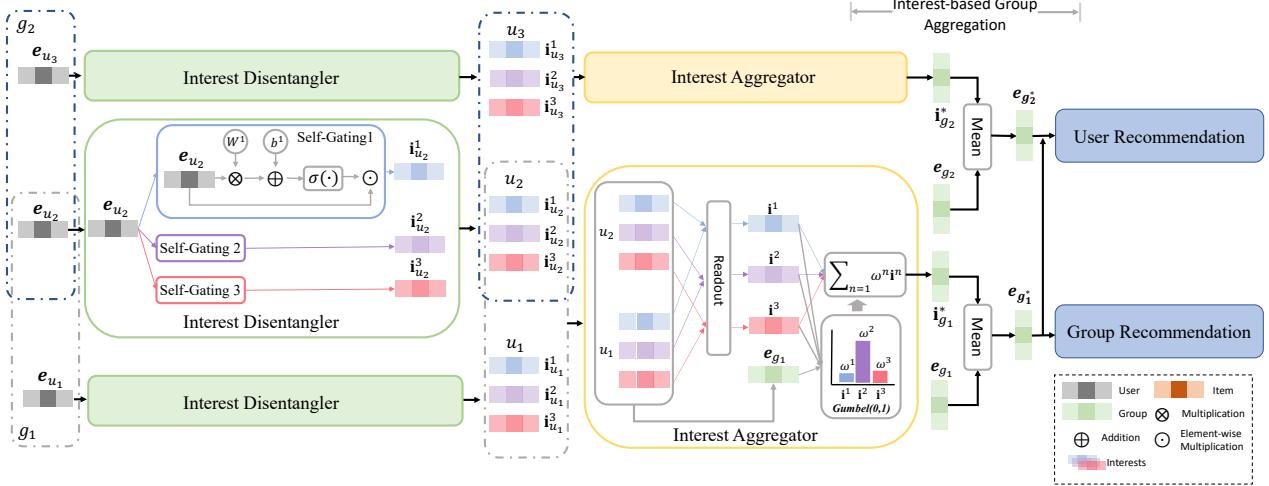


Fig. 3: The framework of IGRec. In this illustrated example,  $u_2$  forms group  $g_1$  with  $u_1$  and group  $g_2$  with  $u_3$ . Based  $g_1$  and  $g_2$ , IGRec predicts  $u_2$ 's rating score towards item  $v_1$ . IGRec consists of 4 modules: 1) Interest disentangler that disentangles each user vector into interest vectors via self-gating. 2) Interest aggregator that selects and aggregates the relevant interests of the members within each group via gumbel sampling to obtain interest-based group representation. 3) Interest-based group aggregation that fuses  $u_2$ 's representation with the participated group representation. 4) Dual-trained rating predictor on both personalized recommendation and group recommendation.

2) *Interest Selection via Gumbel Sampling*: The gumbel sampling method is used to determine the interest of the group formation. Gumbel-Softmax works as a differentiable method for categorical sampling via its reparameterization trick [37]. We can obtain the Gumbel noise  $\mathbf{G}_n \sim \text{Gumbel}(0, 1)$  by:

$$G_n = -\log(-\log(\varepsilon_n)), \quad \varepsilon_n \sim \text{Uniform}(0, 1). \quad (5)$$

Then, we sample from the following equation and draw the  $M$ -dimensional vector  $\omega = (\omega^1, \omega^2, \dots, \omega^M) \in \mathbb{R}^M$ .

$$\omega^n = \text{Softmax}((\log(\psi_n) + G_n)/\tau), \quad \psi_n = \mathbf{e}_g \cdot \mathbf{i}^n, \quad (6)$$

where  $\psi_n$  denotes the relevance score of the  $n$ -th interest under the distribution and  $\tau$  is the temperature that approximate  $\omega$  to a one-hot vector as  $\tau$  goes to 0. Then, the Gumbel-Softmax  $\omega^n$  is assigned as the probability of interest selection as  $p(\delta(g) = n | \{u\}_{u \in U(g)}, g)$  shown in Eq. 2. Finally, for group  $g$ , the learned interest-based vector  $\mathbf{i}_g^* \in \mathbb{R}^d$  is calculated by multiplying  $\omega$  and the aggregated interest vectors  $\mathbf{i}$ :

$$\mathbf{i}_g^* = \sum_{n=1}^M \omega^n \mathbf{i}^n. \quad (7)$$

Gumbel-Softmax has two versions: hard and soft. 1) Hard version: The output  $M$ -dimensional vector  $\omega$  becomes a one-hot vector, which means only the most important interest will be kept. 2) Soft version: It is similar to the general softmax, but it assigns a larger weight to the most essential interest. We propose that users participate in a group for one specific interest or the combination of some interests, which corresponds to the hard Gumbel-Softmax and soft Gumbel-Softmax separately. Our experiments show that soft Gumbel-Softmax achieves better performance than the hard version.

### C. Interest-based Group Aggregation

In this module, we achieve the **effective group representation for personalized RS** by aggregating interest vectors. Given the aggregated interest vector  $\mathbf{i}_g^*$  for group  $g$ , we collect the users' preferences from multiple interests with different weights. Then, IGRec updates group vector  $\mathbf{g}^*$  as:

$$\mathbf{e}_g^* = (\mathbf{e}_g + \mathbf{i}_g^*)/2, \quad (8)$$

The formation of a group is determined by its members, while the group also has implicit influences on each user. Therefore, we update the user's representation  $\mathbf{e}_u$  by aggregating the embeddings of groups in which this user participates:

$$\mathbf{e}_{\hat{u}} = (\mathbf{e}_u + \text{Pooling}(\{\mathbf{e}_{g_i}^*\}_{g_i \in G(u)}))/2, \quad (9)$$

where  $\mathbf{e}_{\hat{u}}$  is the updated user representation,  $G(u)$  represents user  $u$  participated group and  $\text{Pooling}(\cdot)$  denotes the pooling operation for integrating the group information. We chose average pooling for its simplicity and outstanding performance in our experiments.

### D. Dual-trained Rating Predictor

1) *Personalized Recommendation Loss*: GNN-based models [38], [39] have shown that graph neural network achieves better performance on recommendation tasks by aggregating the high-order user-item interactions. Here we utilize LGCN [38] on the user-item bipartite graph to model the relationship between the user and item as:

$$\begin{aligned} \mathbf{e}_{\hat{u}}^{(k+1)} &= \sum_{v \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|} \sqrt{|\mathcal{N}_v|}} \mathbf{e}_v^{(k)}, \\ \mathbf{e}_v^{(k+1)} &= \sum_{u \in \mathcal{N}_v} \frac{1}{\sqrt{|\mathcal{N}_v|} \sqrt{|\mathcal{N}_u|}} \mathbf{e}_{\hat{u}}^{(k)}, \end{aligned} \quad (10)$$

where  $\mathbf{e}_u^{(k)}, \mathbf{e}_v^{(k)}$  denote the latent vectors of user  $u$  and item  $v$  after  $k$  layers propagation. In the user-item bipartite graph,  $\mathcal{N}_u$  denotes the set of items that interact with user  $u$ , and  $\mathcal{N}_v$  denotes the set of users which interacts with item  $v$ . After  $k$  layers propagation, we fuse these embeddings obtained at each layer into the final embedding for each user and item:

$$\mathbf{e}_u = \sum_{k=0}^K \mathbf{e}_u^{(k)}, \mathbf{e}_v = \sum_{k=0}^K \mathbf{e}_v^{(k)}. \quad (11)$$

IGRec predicts the preference score from  $u$  to  $v$  by the inner product of their final representations as:

$$r_{u,v} = \mathbf{e}_u \cdot \mathbf{e}_v. \quad (12)$$

We use the Bayesian Personalized Ranking (BPR) loss [40] to optimize the model for top-N recommendation as:

$$\mathcal{L}_{bpr} = - \sum_{(u,v,j) \in \mathcal{S}_{user}} \log \sigma(r_{u,v} - r_{u,j}), \quad (13)$$

where the  $\mathcal{S}_{user}$  is a set of triplets  $(u, v, j)$  for users,  $v$  is the interacted item for  $u$  and  $j$  is a negative sampled item.

2) *Group Recommendation Loss*: We further conduct the group recommendation task to illustrate the effectiveness of interest-based group representation. Similar to the personalized (user) recommendation task, BPR loss is adopted for this task:

$$\mathcal{L}_{group} = - \sum_{(g,v',j') \in \mathcal{S}_{group}} \log \sigma(\mathbf{e}_g^* \cdot \mathbf{e}_{v'} - \mathbf{e}_g^* \cdot \mathbf{e}_{j'}), \quad (14)$$

where  $\mathcal{S}_{group}$  is a set of triplets  $(g, v', j')$ .  $v'$  is the item that group  $g$  interacted with and  $j'$  is a sampled negative item.

3) *Interest Regularization*: Moreover, one interest embedding captures one certain interest. For example, in Figure 1, the Action group is formed by Alice, Aaron, and Tiffany because of their specific interest in action games. They also participate in other groups of other interests. We further add a regularization term to distinguish different interests:

$$\text{Reg}_{asp} = \sum_{u=1}^{|\mathcal{U}|} \sum_{p=1}^{M-1} \sum_{q=p+1}^M m_{p,q}^u \cdot \text{Sim}(\mathbf{i}_u^p, \mathbf{i}_u^q), \quad (15)$$

where  $\text{Sim}(\cdot)$  is the cosine distance that denotes the interest similarity between  $p$ -th and  $q$ -th interest of user  $u$ . Therefore, the value range of  $\text{Sim}(\cdot)$  is  $[-1, 1]$ , which is calculated as:

$$\text{Sim}(\mathbf{i}_u^p, \mathbf{i}_u^q) = \frac{\mathbf{i}_u^p \cdot \mathbf{i}_u^q}{\|\mathbf{i}_u^p\| \|\mathbf{i}_u^q\|}. \quad (16)$$

Regularizing all interest pairs is inefficient and calculation intensive, and we only regularize on interest pairs with high similarity by a binary mask  $m_{p,q}^u$  controlled by a threshold  $t$ :

$$m_{p,q}^u = \begin{cases} 1, & |\text{Sim}(\mathbf{i}_u^p, \mathbf{i}_u^q)| \geq t \\ 0, & \text{otherwise} \end{cases}, \quad (17)$$

where larger  $t$  allows more information can be shared between different interests, and smaller  $t$  forces interests to be diverse. Then the final loss function is defined as:

$$\mathcal{L} = \eta_1 \mathcal{L}_{bpr} + (1 - \eta_1) \mathcal{L}_{group} + \eta_2 \text{Reg}_{asp} + \lambda \|\Theta\|_2^2. \quad (18)$$

TABLE I: Statistics of the Datasets

| Dataset           | MaFengWo | Douban  | Steam      |
|-------------------|----------|---------|------------|
| #Users            | 5,275    | 12,925  | 375,257    |
| #Items            | 1,513    | 10,783  | 2,580      |
| #U-I interactions | 39,767   | 553,766 | 12,181,681 |
| Density           | 0.498%   | 0.397%  | 1.258%     |
| #Groups           | 995      | 2,753   | 849        |
| #G-I interactions | 3,595    | 82,590  | 25,470     |
| Avg. group size   | 9.63     | 203.46  | 547.10     |

It consists of BPR loss, group recommendation loss, interest regularization, and parameter regularization.  $\eta_1$ ,  $\eta_2$  and  $\lambda$  are the weights for different losses.

#### IV. EXPERIMENTS

In this section, we conduct extensive experiments on IGRec to answer the following Research Questions (RQs):

- **RQ1:** Does IGRec outperform existing methods in group enhanced recommendation?
- **RQ2:** Can IGRec achieve better performance on group recommendation tasks with interest-based group representation?
- **RQ3:** Are the components in IGRec necessary?
- **RQ4:** What is the impact of different experiment settings?

##### A. Experiment Settings

1) *Datasets*: We conduct experiments on three publicly available datasets: MaFengWo [41], Douban [42], and Steam [43]. The details of the datasets can be seen in Table I. MaFengWo is a tourism website that provides a platform for customers to create or join a group trip. Douban, a well-known social media platform in China, allows users to rate and share their experiences with movies and books they watched. Steam is an online gaming platform on which game players can form interest groups to discuss games. Since Steam and Douban datasets do not contain the interactions between group and item, we follow [19] to generate group item interactions based on the top 30 most frequently purchased among all members to run methods that need group-item interactions.

2) *Baselines*: To demonstrate the effectiveness of IGRec, we compare it with three groups of representative baselines. 1) Non-personalized RS (Popularity [44]) that does not differentiate user's personalized preference. 2) Methods without using group information (NeuMF [45], MF [46], LightGCN [38]) that recommend items purely based on user-item interactions. 3) Group-enhanced personalized RS (AGREE [11], GREE [11], GGRM [19]) that predicts user preference by modeling both user-item interactions and group information.

Previous multi-interest works are applied on sequential recommendation tasks [32]–[34], [47]–[52]. We also test the multi-interest frameworks by transforming user-item interactions into sequences and comparing their proposed multi-interest extractor layer with our interest disentangler module. However, our experiments show those methods are specifically designed for the sequential recommendation and perform much worse without items' sequential patterns as in our experimented dataset. Thus, we do not include them as baselines.

TABLE II: Overall comparison, the best and second-best results are in bold and underlined, respectively

| Dataset  | Metric | Popularity          | NeuMF               | MF                  | LightGCN            | GREE                | AGREE               | GGRM                                  | IGRec                                 | Improv. |
|----------|--------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------------------------|---------------------------------------|---------|
| MaFengWo | R@5    | 0.1832 $\pm$ 0.0000 | 0.2312 $\pm$ 0.0378 | 0.2603 $\pm$ 0.0161 | 0.3259 $\pm$ 0.0234 | 0.2915 $\pm$ 0.0098 | 0.2914 $\pm$ 0.0081 | 0.3387 $\pm$ 0.0372                   | <b>0.3694 <math>\pm</math> 0.0016</b> | 9.06%   |
|          | R@10   | 0.3334 $\pm$ 0.0000 | 0.3925 $\pm$ 0.0591 | 0.4285 $\pm$ 0.0188 | 0.4996 $\pm$ 0.0364 | 0.4499 $\pm$ 0.0032 | 0.4477 $\pm$ 0.0043 | 0.5211 $\pm$ 0.0340                   | <b>0.5463 <math>\pm</math> 0.0039</b> | 4.84%   |
|          | N@5    | 0.1297 $\pm$ 0.0000 | 0.1780 $\pm$ 0.0332 | 0.2020 $\pm$ 0.0156 | 0.2601 $\pm$ 0.0217 | 0.2237 $\pm$ 0.0054 | 0.2230 $\pm$ 0.0040 | <u>0.2684 <math>\pm</math> 0.0295</u> | <b>0.3036 <math>\pm</math> 0.0058</b> | 13.11%  |
|          | N@10   | 0.1863 $\pm$ 0.0000 | 0.2386 $\pm$ 0.0410 | 0.2648 $\pm$ 0.0164 | 0.3242 $\pm$ 0.0165 | 0.2821 $\pm$ 0.0029 | 0.2809 $\pm$ 0.0022 | 0.3362 $\pm$ 0.0284                   | <b>0.3684 <math>\pm</math> 0.0056</b> | 9.58%   |
| Douban   | R@5    | 0.0444 $\pm$ 0.0000 | 0.0413 $\pm$ 0.0034 | 0.0713 $\pm$ 0.0026 | 0.0726 $\pm$ 0.0027 | 0.0571 $\pm$ 0.0008 | 0.0571 $\pm$ 0.0005 | 0.0720 $\pm$ 0.0020                   | <b>0.0814 <math>\pm</math> 0.0015</b> | 13.06%  |
|          | R@10   | 0.0762 $\pm$ 0.0000 | 0.0721 $\pm$ 0.0043 | 0.1117 $\pm$ 0.0026 | 0.1164 $\pm$ 0.0025 | 0.0916 $\pm$ 0.0011 | 0.0915 $\pm$ 0.0013 | 0.1143 $\pm$ 0.0027                   | <b>0.1249 <math>\pm</math> 0.0015</b> | 9.27%   |
|          | N@5    | 0.0479 $\pm$ 0.0000 | 0.0479 $\pm$ 0.0035 | 0.0748 $\pm$ 0.0021 | 0.0846 $\pm$ 0.0024 | 0.0759 $\pm$ 0.0003 | 0.0763 $\pm$ 0.0003 | 0.0812 $\pm$ 0.0049                   | <u>0.0992 <math>\pm</math> 0.0009</u> | 22.17%  |
|          | N@10   | 0.0574 $\pm$ 0.0000 | 0.0567 $\pm$ 0.0031 | 0.0867 $\pm$ 0.0020 | 0.0952 $\pm$ 0.0021 | 0.0814 $\pm$ 0.0004 | 0.0819 $\pm$ 0.0005 | 0.0923 $\pm$ 0.0040                   | <b>0.1085 <math>\pm</math> 0.0010</b> | 17.55%  |
| Steam    | R@5    | 0.1154 $\pm$ 0.0000 | 0.0780 $\pm$ 0.0208 | 0.3825 $\pm$ 0.0089 | 0.4268 $\pm$ 0.0050 | 0.1795 $\pm$ 0.0109 | 0.1941 $\pm$ 0.0078 | 0.4255 $\pm$ 0.0027                   | <b>0.4511 <math>\pm</math> 0.0034</b> | 6.02%   |
|          | R@10   | 0.2444 $\pm$ 0.0000 | 0.1415 $\pm$ 0.0322 | 0.4975 $\pm$ 0.0092 | 0.5526 $\pm$ 0.0041 | 0.3157 $\pm$ 0.0248 | 0.3429 $\pm$ 0.0041 | <u>0.5513 <math>\pm</math> 0.0016</u> | <b>0.5727 <math>\pm</math> 0.0024</b> | 3.88%   |
|          | N@5    | 0.1100 $\pm$ 0.0000 | 0.0680 $\pm$ 0.0148 | 0.3944 $\pm$ 0.0090 | 0.4490 $\pm$ 0.0050 | 0.1839 $\pm$ 0.0054 | 0.1927 $\pm$ 0.0044 | 0.4466 $\pm$ 0.0026                   | <b>0.4762 <math>\pm</math> 0.0037</b> | 6.63%   |
|          | N@10   | 0.1594 $\pm$ 0.0000 | 0.0927 $\pm$ 0.0194 | 0.4334 $\pm$ 0.0092 | 0.4895 $\pm$ 0.0047 | 0.2297 $\pm$ 0.0111 | 0.2437 $\pm$ 0.0038 | 0.4868 $\pm$ 0.0021                   | <b>0.5136 <math>\pm</math> 0.0032</b> | 5.51%   |

3) *Evaluation Method*: Same as AGREE [11], for each user(group), we randomly split her(its) interactions into training (80%), validation (10%), and test (10%) sets. We tune hyper-parameters on the validation set and test the model's performance on the test set. Rank-based evaluation metrics include NDCG@K and Recall@K with K ranges in {5,10}. We test each model 5 times with different random seeds to ensure the stability of the result.

4) *Hyper-parameter Settings*: We use Adam [53] as the optimizer in our model. Hyper-parameters are tuned based on grid search. For the common hyper-parameters learning rate, weight decay, and embedding size, we tune them within the ranges of {0.1, 0.05, 0.01, 0.005, 0.001}, {1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6} and {8, 16, 32, 64, 128, 256, 512}, respectively. We tune the number of LGCN for LightGCN, GGRM, and our model from 1 to 6. The number of interests  $M$ , temperature  $\tau$ , and interest regularization  $\eta$  are the particular hyper-parameters in our model. We tune the number of interests  $M$  with the range of {2, 3, 4, 5, 6, 7, 8}, and tune temperature  $\tau$  and interest regularization  $\eta$  from 0.1 to 1 with interval 0.1.

### B. Performance Comparison (RQ1)

Experiment results on performance comparison are shown in Table II. We can have the following observations.

- IGRec consistently outperforms other methods on the three datasets. On NDCG@5, IGRec has over 5% improvement over the second-best method on the three datasets. The variance of IGRec is also similar to the best baseline. It shows the effectiveness of IGRec by enhancing RS with group information. It reveals that disentanglement in IGRec is effective in using group information.
- For other group recommendation models, GREE and AGREE perform worse than LightGCN. It shows joint learning on user-item and group-item bipartite graphs do not generalize to personalized recommendations well. IGRec and GGRM perform better than baselines without using group information (MF and LightGCN). It shows group information can be utilized to enhance the RS.
- GNN-based models such as LightGCN, GGRM, and IGRec exhibit notable superiority over other techniques. This underscores the GNN's proficiency in acquiring user and item representations by leveraging high-order connectivity.

### C. Group Recommendation Task (RQ2)

Group recommendation focuses on providing personalized recommendations to a group of individuals rather than individual recommendations. In this section, we conduct experiments on three datasets to demonstrate that IGRec can achieve good performance on group recommendation tasks.

Except for AGREE and GREE models, two state-of-the-art group recommendation models: GroupIM [35] and SG-GCF [54], are chosen as our baselines. Experiment results are shown in Table III. GroupIM's performance on Douban and Steam are excluded because it exceeds the memory capacity of our machine. ND and ED of SG-GCF represent the node-dropout and edge-dropout augmentation methods, respectively. From the group recommendation experiments, we can have the following observations:

- On Recall and NDCG, IGRec consistently has a huge improvement over the second-best method on MaFengWo and Steam datasets. On NDCG@10, IGRec even outperforms more than 7% over the second-best method. It shows the effectiveness of interest-based group representation.
- However, IGRec performs slightly worse than SG-GCF on the Douban dataset. Compared with MaFengWo and Steam, where groups are formed spontaneously, Douban's group information is manually built from previous research [55]. Thus, the improvement on MaFengWo and Steam datasets aligns better with real-life situations.

### D. Ablation Study (RQ3)

In this section, we conduct an ablation study to investigate the advantages of each component in IGRec. We consider 4 variants demonstrated in Fig. 4. (1) Variant A removes the multi-interest module; (2) Variant B is built by replacing gumbel softmax by mean pooling on aggregating interests to the group. (3) Variant C removes the regularization of interests. (4) Variant D utilizes hard Gumbel-Softmax.

We can have the following observations. (1) The regularization of interests has a critical impact on IGRec's performance. From the performance of variant C in each histogram, we find that removing the regularization of interests significantly influences the performance of IGRec. The model suffers from performance drop in all metrics, e.g., NDCG@10, 9.14% in MaFengWo, 28.78% in Douban, and 4.06% in Steam. We aim to disentangle interests from users' embedding and let

TABLE III: Comparison of Top-K performance on three datasets with baselines in group recommendation problem. The best and second-best results are in bold and underlined, respectively

| Dataset  | Metric | GREE                | AGREE               | GroupIM                               | SGGCF (ND)                            | SGGCF (ED)                            | IGRec                                 | Improv. |
|----------|--------|---------------------|---------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------|
| MaFengWo | R@5    | 0.0028 $\pm$ 0.0000 | 0.0028 $\pm$ 0.0000 | 0.3114 $\pm$ 0.0243                   | 0.3178 $\pm$ 0.0122                   | 0.3240 $\pm$ 0.0025                   | <b>0.5477 <math>\pm</math> 0.0340</b> | 69.04%  |
|          | R@10   | 0.0055 $\pm$ 0.0051 | 0.0033 $\pm$ 0.0005 | <u>0.4173 <math>\pm</math> 0.0180</u> | 0.3607 $\pm$ 0.0044                   | 0.3658 $\pm$ 0.0075                   | <b>0.6345 <math>\pm</math> 0.0247</b> | 73.46%  |
|          | N@5    | 0.0022 $\pm$ 0.0004 | 0.0022 $\pm$ 0.0003 | <u>0.1954 <math>\pm</math> 0.0115</u> | 0.2645 $\pm$ 0.0096                   | 0.2702 $\pm$ 0.0046                   | <b>0.4456 <math>\pm</math> 0.0256</b> | 64.91%  |
|          | N@10   | 0.0031 $\pm$ 0.0017 | 0.0024 $\pm$ 0.0004 | 0.2324 $\pm$ 0.0085                   | 0.2795 $\pm$ 0.0076                   | 0.2849 $\pm$ 0.0055                   | <b>0.4761 <math>\pm</math> 0.0229</b> | 67.11%  |
| Douban   | R@5    | 0.5353 $\pm$ 0.0395 | 0.5349 $\pm$ 0.0323 | -                                     | 0.5268 $\pm$ 0.0152                   | <b>0.5350 <math>\pm</math> 0.0131</b> | 0.5143 $\pm$ 0.0055                   | -3.87%  |
|          | R@10   | 0.6778 $\pm$ 0.0403 | 0.6772 $\pm$ 0.0355 | -                                     | 0.6916 $\pm$ 0.0062                   | <u>0.6953 <math>\pm</math> 0.0039</u> | <b>0.6987 <math>\pm</math> 0.0035</b> | 0.49%   |
|          | N@5    | 0.4941 $\pm$ 0.0278 | 0.4945 $\pm$ 0.0225 | -                                     | <u>0.4731 <math>\pm</math> 0.0120</u> | <b>0.4819 <math>\pm</math> 0.0139</b> | 0.4540 $\pm$ 0.0030                   | -5.79%  |
|          | N@10   | 0.5593 $\pm$ 0.0286 | 0.5597 $\pm$ 0.0245 | -                                     | 0.5489 $\pm$ 0.0081                   | <u>0.5557 <math>\pm</math> 0.0098</u> | 0.5389 $\pm$ 0.0014                   | -3.02%  |
| Steam    | R@5    | 0.4174 $\pm$ 0.0485 | 0.4663 $\pm$ 0.1249 | -                                     | 0.4935 $\pm$ 0.0127                   | 0.4930 $\pm$ 0.0123                   | <b>0.5366 <math>\pm</math> 0.0096</b> | 8.73%   |
|          | R@10   | 0.5725 $\pm$ 0.0493 | 0.6383 $\pm$ 0.0318 | -                                     | 0.7066 $\pm$ 0.0045                   | 0.7060 $\pm$ 0.0050                   | <b>0.7611 <math>\pm</math> 0.0029</b> | 7.71%   |
|          | N@5    | 0.3838 $\pm$ 0.0439 | 0.4273 $\pm$ 0.0074 | -                                     | <u>0.4355 <math>\pm</math> 0.0140</u> | 0.4352 $\pm$ 0.0138                   | <b>0.4692 <math>\pm</math> 0.0066</b> | 7.74%   |
|          | N@10   | 0.4543 $\pm$ 0.0444 | 0.5056 $\pm$ 0.0153 | -                                     | 0.5330 $\pm$ 0.0100                   | 0.5327 $\pm$ 0.0101                   | <b>0.5721 <math>\pm</math> 0.0044</b> | 7.34%   |

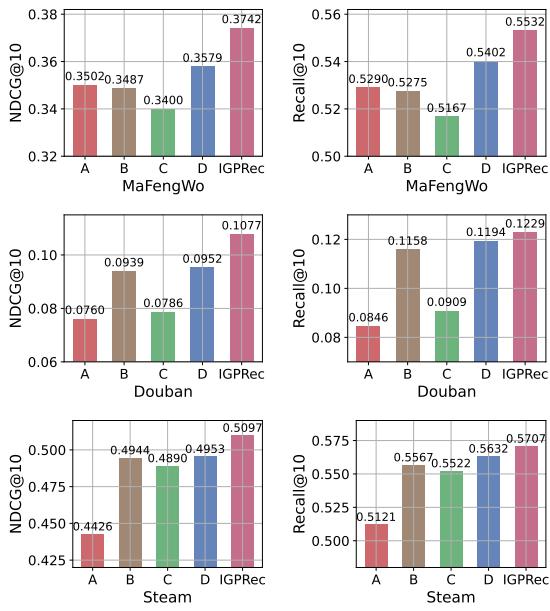


Fig. 4: Ablation study of IGRec.

these interests be distinguishable. After removing the regularization of interests, it is hard to differentiate information among interests such that group embedding could not be representative corresponding to some specific interests. (2) The influence of multi-interest is highly related to datasets. Variant A learns group embeddings using mean pooling to directly aggregate members' embedding. The decrements of all metrics compared with **IGRec** are different in all datasets, which are 6.41%, 29.43%, and 13.16% dropping ratio of NDCG@10 in MaFengWo, Douban, and Steam, respectively. This mainly depends on the property of datasets. Overall, it reveals that the multi-interests module is effective. (3) Soft Gumbel Softmax can further improve the performance. Variant B is the model that keeps all other modules but replaces Gumbel softmax with mean pooling to aggregate learned interests to the groups and variant D replaces soft Gumbel Softmax with the hard version. Among the four modules, soft Gumbel Softmax has the least improvement on **IGRec**.

TABLE IV: Benefit of Self-Gating

| Method             | R@5           | R@10          | N@5           | N@10          | # Parameters                       |
|--------------------|---------------|---------------|---------------|---------------|------------------------------------|
| 1 FC layer         | 0.2721        | 0.4266        | 0.2131        | 0.2699        | $M \times (d+1) \times d$          |
| 2 FC layers        | 0.2104        | 0.3418        | 0.1662        | 0.2144        | $2 \times M \times (d+1) \times d$ |
| Interest embedding | 0.3496        | 0.5251        | 0.2811        | 0.3467        | $M \times  \mathcal{U}  \times d$  |
| Self-Gating        | <b>0.3694</b> | <b>0.5463</b> | <b>0.3036</b> | <b>0.3684</b> | $M \times (d+1) \times d$          |
| Improv.            | 5.66%         | 4.04%         | 8.00%         | 6.26%         |                                    |

#### E. Hyper-parameter Analysis (RQ4)

1) *Hyper-parameter Analysis on  $M$ ,  $\eta$ ,  $t$ , and  $\tau$ :* The experiment results are shown in Fig. 5. The shadow space represents the variance. Due to the limited space, we only present experiment results on MaFengWo and Steam datasets. A similar trend is also observed in Douban. The number of interests  $M$ , interest regularization  $\eta$ , and threshold  $t$  are three representative parameters in **IGRec**. For the number of interests from Fig. 5a and Fig. 5e, we observe that the performance of NDCG@5 drops after achieving the peak point for each dataset. The range of  $M$  for Steam dataset is from 1 to 5 due to the limited memory space. Similarly, increasing the value of interest regularization  $\eta$  will lead to better performance at the first few steps but fall rapidly when the value is too large, which is shown in Fig. 5b and Fig. 5f. Fig. 5c and Fig. 5g show the benefit of the threshold. Best performance is achieved when the threshold among interest embeddings is small, which leads to more diverse interests. By contrast, the temperature parameter  $\tau$  is the least important hyper-parameter to our model, illustrated in Fig. 5d and Fig. 5h.

2) *Interest Generation Experiments:* We compare self-gating with three methods, one and two Fully Connected (FC) layers and interest embedding. FC methods generate interests by linear layers, and interest embedding directly represents  $M$  interests by  $M$  embedding tables. Table IV shows that the self-gating outperforms the second best by about 5.35% on Recall and 7.13% on NDCG. Besides, self-gating and FC layers need much fewer parameters than interest embedding since  $|\mathcal{U}| \gg (d+1)$ . The self-gating mechanism achieves the best performance with fewer parameters, which effectively and efficiently generates interest from user embedding.

3) *The Influence of  $t$  on interest similarity:* The heatmaps in Fig. 6 show the average pairwise cosine similarity between interest embeddings. According to Fig. 6a, each interest is distinguished compared with other interests when the threshold

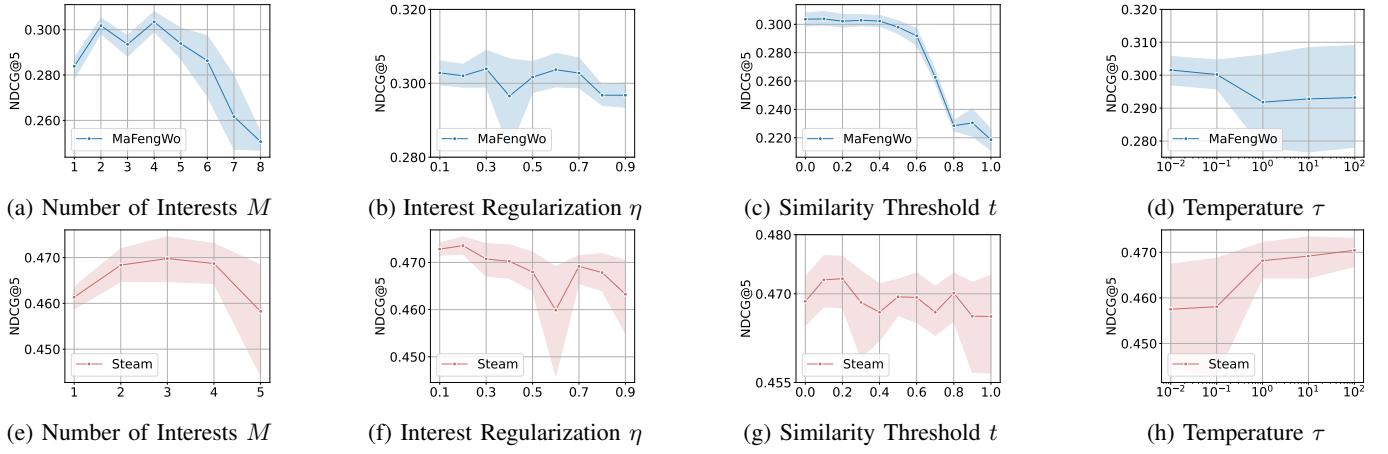


Fig. 5: The influence of model hyper-parameters  $M$ ,  $\eta$ ,  $t$ , and  $\tau$ .

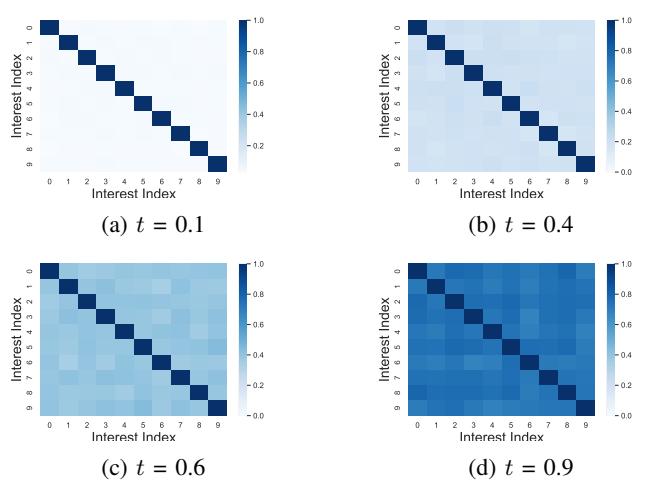


Fig. 6: The influence of threshold  $t$  on Douban dataset

$t$  is 0.1. On the other hand, a larger  $t$  allows interests to be similar (dark color) and share more common information with each other as shown in Fig. 6b, Fig. 6c, and Fig. 6d. The best performance is achieved with a small threshold from Fig. 5c and Fig 5g. It shows smaller  $t$  helps achieve better performance by distinguishing the representation of the interests.

4) *The effect of  $\eta_1$  on personalized recommendation and group recommendation:* The line maps in Fig. 7 show the changing trend of user recommendation and group recommendation performances along with the increment of  $\eta_1$ . According to Fig. 7a, the performance on user recommendation increases steadily and achieves best when  $\eta_1 = 0.9$ . For Fig. 7b, the best performance on group recommendation is achieved as  $\eta_1 = 0.2$ , which indicates the user recommendation task could promote the group recommendation. Then, the results decrease when increasing  $\eta_1$  because the model focuses primarily on user recommendation tasks.

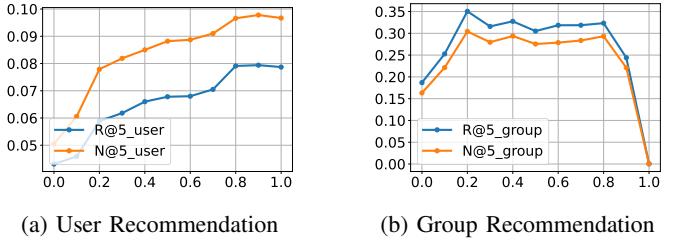


Fig. 7: The effect of  $\mathcal{L}_{bpr}$  weight  $\eta_1$  on Douban dataset

## V. RELATED WORKS

### A. Group Enhanced Recommendation

RS that utilize group information can be classified into two categories: group-enhanced personalized RS and group-oriented RS. Group-enhanced personalized RS aims to recommend items to users by utilizing users' group participation as side information. To differentiate group members' contributions by attention mechanism, AGREE [11] learns the dynamic aggregation from group members to infer the group's decision on the item. Yuan et al. [17] consider users' community memberships and friendship connections as social connections and get improvements on the overall performance of recommendations via the fusion of social connections and collaborative filtering. Lee et al. [18] explore the feasibility and value of community-based sociality for a personalized recommendation. In order to capture preference dependency relationships between the target user and other group members, GGRM [19] applies a graph neural network on three bipartite graphs (users-items, users-groups, and groups-items) to learn and concatenate user and item embedding for predicting.

Group-oriented RS aims to learn group representation and recommend items to a group of users. Previous works learn group representation by aggregating the preferences of its members with various predefined aggregation approaches, such as average [56], least misery [57] and maximum satisfaction [58]. The predefined approaches cannot capture the complicated dynamic process of group RS. Recent works

learn to differentiate group members' contributions by attention mechanism. MoSAN [36] models the user's preference with respect to all other members in the same group by sub-attention module. GroupIM [35] maximizes the mutual information between the user representation and their group representations which are aggregated from its members' preferences via the attention mechanism. Further, SGGCF [54] constructs a unified user-centered graph and implements group recommendation with self-supervised learning.

Previous works directly aggregate the group members' embedding into group representation, which assumes users who participate in the same group share totally the same interests. They fail to consider that users join different groups out of different interests in real life, and a single group can only partially reflect the user's interests. Following this idea, **IGRec** disentangles the user's embedding into different interests and learns interest-based group representation.

### B. Multi-interest Recommendation

The majority of RS models represent a single user by one vector with a fixed length. However, such a user representation might be less effective in capturing users' diverse interests. Recently, learning multiple representations/interests of each user has been attracting much research attention [32]–[34], [47]–[52]. MIND [32] designs a multi-interest extractor layer with Behavior-to-Interest (B2I) dynamic routing for clustering users' behavior into interest representations and then chooses the interest associated with the specific item through a label-aware attention layer. ComiRec [34] provides two methods to construct a multi-interest extraction module: dynamic routing method and self-attentive method. Additionally, it retrieves top-k items by a greed inference algorithm and introduces a controllable factor to trade off accuracy and diversity of recommendations. Zhang et al. [33] devise a Re4 framework consisting of Re-contrast, Re-attend, and Re-construct. These three modules address the following problems: the distinction among interest representations, the consistency between the forward attention weights and the interest-item correlation, and the semantic reflection from interest on representative items.

Taking full advantage of group information can help infer users' interest in items. Motivated by this, we introduce **IGRec** that fills this research blank and propose to learn interest-based group representations.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we study how to enhance recommender systems with users' group information. Compared with previous works, we argue that users join different groups out of different reasons, and one group can only reveal one or more interests of group members. We make the first attempt to disentangle users' interest in items via users' group participation and propose **IGRec** that disentangles users' embedding into interests and learns interest-based group representation. Experiment results show we significantly outperform the SOTA group enhanced personalized RS and group recommendation RS.

## VII. ACKNOWLEDGE

This work is supported in part by NSF under grant III-2106758.

## REFERENCES

- [1] C. Gao, X. Wang, X. He, and Y. Li, "Graph neural networks for recommender system," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1623–1625.
- [2] J. Leskovec, "New directions in recommender systems," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 2015, pp. 3–4.
- [3] Y. Song, S. Sun, J. Lian, H. Huang, Y. Li, H. Jin, and X. Xie, "Show me the whole world: Towards entire item space exploration for interactive personalized recommendations," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 947–956.
- [4] G. Shani and A. Gunawardana, "Evaluating recommendation systems," *Recommender systems handbook*, pp. 257–297, 2011.
- [5] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.
- [6] G. Guo, "Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems," in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 451–454.
- [7] R. Togashi, M. Otani, and S. Satoh, "Alleviating cold-start problems in recommendation through pseudo-labelling over knowledge graph," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 931–939.
- [8] M. Volkovs, G. Yu, and T. Poutanen, "Dropoutnet: Addressing cold start in recommender systems," *Advances in neural information processing systems*, vol. 30, 2017.
- [9] Y. Wang, C. Feng, C. Guo, Y. Chu, and J.-N. Hwang, "Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 717–725.
- [10] H. Yin, Q. Wang, K. Zheng, Z. Li, and X. Zhou, "Overcoming data sparsity in group recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3447–3460, 2020.
- [11] 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.
- [12] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert systems with Applications*, vol. 23, no. 3, pp. 329–342, 2002.
- [13] M. S. Pera and Y.-K. Ng, "A personalized recommendation system on scholarly publications," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 2133–2136.
- [14] Y. Pang, L. Wu, Q. Shen, Y. Zhang, Z. Wei, F. Xu, E. Chang, B. Long, and J. Pei, "Heterogeneous global graph neural networks for personalized session-based recommendation," in *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022, pp. 775–783.
- [15] H. Chen, Y. Li, X. Sun, G. Xu, and H. Yin, "Temporal meta-path guided explainable recommendation," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 1056–1064.
- [16] Y. Li, H. Chen, X. Sun, Z. Sun, L. Li, L. Cui, P. S. Yu, and G. Xu, "Hyperbolic hypergraphs for sequential recommendation," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 988–997.
- [17] Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, and W. Zheng, "Augmenting collaborative recommender by fusing explicit social relationships," in *Workshop on Recommender Systems and the Social Web, Recsys 2009*, 2009.
- [18] D. H. Lee and P. Brusilovsky, "Improving personalized recommendations using community membership information," *Information Processing & Management*, vol. 53, no. 5, pp. 1201–1214, 2017.
- [19] Z. Tian, Y. Liu, J. Sun, Y. Jiang, and M. Zhu, "Exploiting group information for personalized recommendation with graph neural networks," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 2, pp. 1–23, 2021.

[20] R. Sato, "Enumerating fair packages for group recommendations," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 870–878.

[21] E. Ntoutsi, K. Stefanidis, K. Nørvåg, and H.-P. Kriegel, "Fast group recommendations by applying user clustering," in *Conceptual Modeling: 31st International Conference ER 2012, Florence, Italy, October 15–18, 2012. Proceedings 31*. Springer, 2012, pp. 126–140.

[22] 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.

[23] M. Stratigi, E. Pitoura, and K. Stefanidis, "Squirrel: A framework for sequential group recommendations through reinforcement learning," *Information Systems*, vol. 112, p. 102128, 2023.

[24] X. Li, M. Zhang, S. Wu, Z. Liu, L. Wang, and S. Y. Philip, "Dynamic graph collaborative filtering," in *2020 IEEE international conference on data mining (ICDM)*. IEEE, 2020, pp. 322–331.

[25] Z. Liu, X. Li, Z. Fan, S. Guo, K. Achan, and S. Y. Philip, "Basket recommendation with multi-intent translation graph neural network," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 728–737.

[26] X. Li, Z. Liu, S. Guo, Z. Liu, H. Peng, S. Y. Philip, and K. Achan, "Pre-training recommender systems via reinforced attentive multi-relational graph neural network," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 457–468.

[27] X. Li, Y. Liu, Z. Liu, and S. Y. Philip, "Time-aware hyperbolic graph attention network for session-based recommendation," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 626–635.

[28] D. Li, D. Bhargavi, and V. S. Ravipati, "Impression-informed multi-behavior recommender system: A hierarchical graph attention approach," *arXiv preprint arXiv:2309.03169*, 2023.

[29] L. Yang, S. Wang, Y. Tao, J. Sun, X. Liu, P. S. Yu, and T. Wang, "Dgrec: Graph neural network for recommendation with diversified embedding generation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 661–669.

[30] L. Yang, Z. Liu, C. Wang, M. Yang, X. Liu, J. Ma, and P. S. Yu, "Graph-based alignment and uniformity for recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 4395–4399.

[31] 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.

[32] C. Li, Z. Liu, M. Wu, Y. Xu, H. Zhao, P. Huang, G. Kang, Q. Chen, W. Li, and D. L. Lee, "Multi-interest network with dynamic routing for recommendation at tmall," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 2615–2623.

[33] S. Zhang, L. Yang, D. Yao, Y. Lu, F. Feng, Z. Zhao, T.-s. Chua, and F. Wu, "Re4: Learning to re-contrast, re-attend, re-construct for multi-interest recommendation," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2216–2226.

[34] Y. Cen, J. Zhang, X. Zou, C. Zhou, H. Yang, and J. Tang, "Controllable multi-interest framework for recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2942–2951.

[35] A. Sankar, Y. Wu, Y. Wu, W. Zhang, H. Yang, and H. Sundaram, "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.

[36] 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.

[37] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[38] 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.

[39] L. Yang, Z. Liu, Y. Dou, J. Ma, and P. S. Yu, "Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation," in *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*, 2021, pp. 2141–2145.

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

[41] 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*, vol. 33, no. 3, pp. 1195–1209, 2019.

[42] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proceedings of the 24th ACM international conference on information and knowledge management*, 2015, pp. 453–462.

[43] L. Yang, Z. Liu, Y. Wang, C. Wang, Z. Fan, and P. S. Yu, "Large-scale personalized video game recommendation via social-aware contextualized graph neural network," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3376–3386.

[44] 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.

[45] 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.

[46] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[47] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1059–1068.

[48] Z. Chai, Z. Chen, C. Li, R. Xiao, H. Li, J. Wu, J. Chen, and H. Tang, "User-aware multi-interest learning for candidate matching in recommenders," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1326–1335.

[49] Y. Tian, J. Chang, Y. Niu, Y. Song, and C. Li, "When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1632–1641.

[50] G. Chen, X. Zhang, Y. Zhao, C. Xue, and J. Xiang, "Exploring periodicity and interactivity in multi-interest framework for sequential recommendation," *arXiv preprint arXiv:2106.04415*, 2021.

[51] D. Liu, Y. Yang, M. Zhang, W. Wu, X. Xie, and G. Sun, "Knowledge enhanced multi-interest network for the generation of recommendation candidates," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3322–3331.

[52] Q. Tan, J. Zhang, J. Yao, N. Liu, J. Zhou, H. Yang, and X. Hu, "Sparse-interest network for sequential recommendation," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 598–606.

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

[54] K. Li, C.-D. Wang, J.-H. Lai, and H. Yuan, "Self-supervised group graph collaborative filtering for group recommendation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 69–77.

[55] 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.

[56] 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.

[57] 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.

[58] 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, 2011, pp. 1–20.