AlignGroup: Learning and Aligning Group Consensus with Member Preferences for Group Recommendation

Jinfeng Xu
jinfeng@connect.hku.hk
The University of Hong Kong
HongKong SAR, China

Zheyu Chen
zheyu.chen@connect.polyu.hk
The Hong Kong Polytechnic University
HongKong SAR, China

Jinze Li
lijinze-hku@connect.hku.hk
The University of Hong Kong
HongKong SAR, China

Shuo Yang
shuoyang.ee@gmail.com
The University of Hong Kong
HongKong SAR, China

Hewei Wang
heweiw@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Edith C. H. Ngai*
chngai@eee.hku.hk
The University of Hong Kong
HongKong SAR, China

ABSTRACT

Group activities are important behaviors in human society, providing personalized recommendations for groups is referred to as the group recommendation task. Existing methods can usually be categorized into two strategies to infer group preferences: 1) determining group preferences by aggregating members' personalized preferences, and 2) inferring group consensus by capturing group members' coherent decisions after common compromises. However, the former would suffer from the lack of group-level considerations, and the latter overlooks the fine-grained preferences of individual users. To this end, we propose a novel group recommendation method **AlignGroup**, which focuses on both group consensus and individual preferences of group members to infer the group decision-making. Specifically, AlignGroup explores group consensus through a well-designed hypergraph neural network that efficiently learns intra- and inter-group relationships. Moreover, AlignGroup innovatively utilizes a self-supervised alignment task to capture fine-grained group decision-making by aligning the group consensus with members' common preferences. Extensive experiments on two real-world datasets validate that our AlignGroup outperforms the state-of-the-art on both the group recommendation task and the user recommendation task, as well as outperforms the efficiency of most baselines.

CCS CONCEPTS

• **Information systems → Recommender systems;**.

KEYWORDS

Group Recommendation, Self-supervised learning, Hypergraph Neural Network

ACM Reference Format:
Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Hewei Wang, and Edith C. H. Ngai. 2024. AlignGroup: Learning and Aligning Group Consensus with Member Preferences for Group Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679697

1 INTRODUCTION

As social media becomes ubiquitous [25, 28], the online platform has become the main position and key information source of group activities, such as Mafengwo facilitating travel planning and Meetup hosting social gatherings [30]. Traditional recommendation systems, designed for individuals, struggle to cope with the complexities of group decision-making. This gap has spurred the need for group recommendation systems, which aim to balance members' individual preferences and group consensus to deliver fine-grained recommendations. In the e-commerce domain, such systems are crucial, as seen from Facebook Events to Yelp, where they must respect individual tastes while capturing group desires to ensure the accuracy of recommendations [7, 32].

Echoing the methods of conventional recommender systems, latent factor models are pivotal in the domain of group recommendations. Specifically, groups and items are represented as vectors by mapping into the same semantic space. Then estimating the suitable items closes with the group decision-making. Recent works [1–6, 11, 14, 24] on group recommendations have largely concentrated on aggregating the preferences of members to predict the decision-making of a group. Classic aggregation techniques rely on pre-defined heuristic rules, such as averaging [3], minimizing misery [1], or maximizing satisfaction [2]. More recently, there are also advanced deep-learning models [4, 5, 11, 14, 24] that employ attention mechanisms to facilitate preference aggregation. For example, AGREE [4] proposes an attentive group recommendation that dynamically aggregates the preferences of users within a group, and HyperGroup [10] explores the superior representational capabilities of hypergraphs for group recommendation. CubeRec [6] leverages the geometrically rich structure of hypercubes [20, 33] to integrate multifaceted member preferences. It is worth mentioning that ConsRec [26] realizes that group consensus cannot be integrated by

[*] Corresponding author.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10…$15.00
https://doi.org/10.1145/3627673.3679697

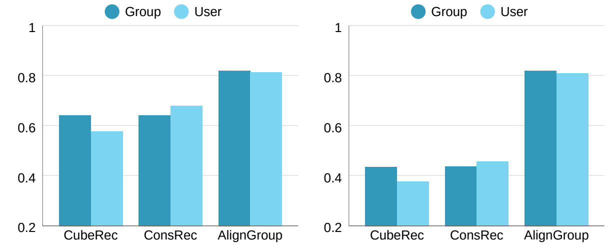
group users' preferences alone; it reveals the consensus hidden behind group behavior, achieving state-of-the-art performance.

Nevertheless, the majority of approaches lack the capability to adequately capture group consensus, instead simply inferring the group decision-making by aggregating individual user preferences. Group consensus is not simply the aggregation of member preferences. For example, if a family of three goes to the movies, the kid likes animated movies, the parents like action movies, and ultimately the group consensus drives them to choose an educational movie. Although CubeRec adopts a geometric perspective, harnessing hypercubes in the semantic space to represent group consensus, the substantial volume of these hypercubes may compromise the accuracy of consensus representation. ConsRec extracts group consensus through multi-views of information. However, it greatly smoothes out the individual preferences of group members. Fig 1 displays the performance comparison of CubeRec, ConsRec, and our AlignGroup on the CAMRa2011 dataset for both group and user recommendation tasks. It is observed that CubeRec and ConsRec perform poorly on the Normalized Discounted Cumulative Gain (NDCG) compared with the Hit Ratio (HR). It is worth noting that the NDCG additionally considers the position of the demand item in the model's recommendation list compared to HR. Therefore, we point out that CubeRec and ConsRec are too coarse in inferring the group decision-making, and we further attribute this to the fact that both CubeRec and ConsRec only consider the group consensus but ignore individualized preferences for users within a group.

To this end, we propose a novel group recommendation method **AlignGroup**, which focuses on both group consensus and individual preferences of group members to infer the group decision-making. Specifically, we innovatively propose a hypergraph neural network to capture group consensus from both intra- and inter-group relations. The intra-group relations are learned by aggregating group member preferences and enhanced from high-order related neighbor groups' information through hypergraph convolution operations. For inter-group relations, we effectively capture the group relations from a member overlap perspective. Besides, to effectively infer user preferences and item properties in the same semantic space, we augment user and item representations by aggregating the related group consensus for each user or item. Furthermore, we introduce a novel and simple self-supervised alignment task to effectively capture the fine-grained group decision-making by aligning the group consensus with members' common preferences. In Section 4.2, we discuss different strategies and scopes for calculating members' common preferences, and further evaluate them in Section 5.3.2.

In a nutshell, the contributions of our work are as follows:

- We propose a novel framework AlignGroup for the group recommendation, which can effectively infer the fine-grained group decision-making by considering both group consensus and members' preferences.
- We develop a new hypergraph neural network, which well-extracts group consensus from both intra- and inter-group relations.
- We design a self-supervised alignment task, effectively reducing the gap between member preferences and group consensus.
- We perform comprehensive experiments on two real-world datasets to validate both the effectiveness and efficiency of AlignGroup.



(a) HR@5 (b) NDCG@5

Figure 1: Performance comparison of CubeRec, ConsRec, and our AlignGroup on both group recommendation and user recommendation tasks in terms of Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) on CAMRa2011.

2 RELATED WORK

In this section, we review three related lines of research that contribute to group recommendation, namely methods for preference aggregation, hypergraph learning, and self-supervised learning.

2.1 Preference Aggregation

Earlier works use pre-defined heuristic rules to aggregate the preferences of users within a group to infer the group decision-making, such as averaging [3], minimizing misery [1], or maximizing satisfaction [2]. However, in the real world, each member has a different status and role within the group. Therefore, pre-defined aggregation rules cannot dynamically adjust the weights. Recently, many deep learning-based aggregation methods have been proposed, which dynamically adjust the weight share of each user in a group through a learnable weight matrix. For example, MoSAN [24] first utilizes the attention mechanism to capture fine-grained interactions among group members dynamically. Nevertheless, it only considers users as group members. AGREE [4] broadens the scope of group members based on MoSAN, which considers both users and items as group members, and SoAGREE [5] further exploits the relationships among users to enhance the user representation in AGREE. SIGR [29] exploits and integrates the global and local social influence of users. CAGR [30] first tries to learn centrality-aware user representations and then leverages it to refer to group decision-making. GroupIM [22] adopts mutual information maximization between users and groups to overcome the data sparsity problem between items and groups. HyperGroup [10] proposes a hyperedge embedding-based method to enhance the representations. It is worth noting that some recent works break the constraint of aggregating information, such as CubeRec [6] exploits the geometric representational power of hypercubes to frame the scope of group consensus, and ConsRec [26] utilizes multilevel views to collectively expose group consensus to infer group decision-making. However, we note that relying solely on group consensus or aggregating the preferences of group members is insufficient. Therefore, only by taking into account both individual preferences and group consensus can group decision-making be effectively inferred.

2.2 Hypergraph Learning

The relationship between the edge and vertex in the hypergraph is naturally similar to the relationship between group and member.

Recent works [10, 15, 26, 32] utilize hypergraphs to improve group recommendation performance. HyperGroup [10] utilizes the edges of the hypergraph to represent groups, effectively exploiting the hypergraph structure to propagate in-group user collaboration signals. HCR [15] utilizes a hypergraph structure to simultaneously propagate collaboration signals between users and items. S^2-HHGR [32] further utilizes the hypergraph to capture both in-group and out-group user relationships while utilizing self-supervised signals to align user representations at different granularities. ConsRec [26] propagates multi-view collaborative signals from group and member perspectives to achieve state-of-the-art performance.

2.3 Self-supervised Learning

Recent works point out that self-supervised learning can also be effective in mitigating the data sparsity problem on the group recommendation task, GroupIM [22] utilizes self-supervised signals to maximize the mutual information of users and groups to improve the representation, S^2-HHGR [32] utilizes self-supervised signals to efficiently align user representations at different granularities, and CubeRec [6] proposes a novel self-supervised signal to capture the common preferences of users within a hypercube.

3 PRELIMINARY

Formally, we use bold lowercase letters (e.g., $\mathbf{e}$) and bold capital letters (e.g., $\mathbf{E}$) to represent vectors and matrices, respectively. None-bold letters (e.g., x) and squiggle letters (e.g., $\mathcal{X}$) are used to denote scalars and sets, respectively.

3.1 Task Definition

Let $\mathcal{U} = \{u_1, ..., u_{|\mathcal{U}|}\}$, $\mathcal{I} = \{i_1, ..., i_{|\mathcal{I}|}\}$, and $\mathcal{G} = \{g_1, ..., g_{|\mathcal{G}|}\}$ be the sets of users, items, and groups respectively. There are two types of observed interactions. We use $\mathbf{R}^{\mathbf{U}} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$ to denote the user-item interactions between $\mathcal{U}$ and $\mathcal{I}$, where the element $\mathbf{R}^{\mathbf{U}}(\mathbf{n},\mathbf{m})$ = 1 if user u_n has interacted with item i_m otherwise $\mathbf{R}^{\mathbf{U}}(\mathbf{n},\mathbf{m})$ = 0. Similarly, we use $\mathbf{R}^{\mathbf{G}} \in \mathbb{R}^{|\mathcal{G}|\times|\mathcal{I}|}$ to denote the group-item interactions between $\mathcal{G}$ and $\mathcal{I}$. The group $g \in \mathcal{G}$ consists of a set of user members $\mathcal{U}_g = \{u_1, ..., u_p, ..., u_{|\mathcal{U}_g|}\}$ where $u_p \in \mathcal{U}$. We denote the interacted item set of g as $\mathcal{I}_g = \{i_1, ..., i_q, ..., i_{|\mathcal{I}_g|}\}$ where $i_q \in \mathcal{I}$. The task for group recommendation is defined as recommending items that target group g may be interested in.

3.2 Hypergraph

Hypergraph is a more complex and enriched topological structure. Each hyperedge can contain more than one vertex. Formally, we define the hypergraph as $G = (\mathcal{V}, \mathcal{E}, \mathcal{H})$ where $\mathcal{V} = \mathcal{U} + \mathcal{I}$ is the vertex set, $\mathcal{E}$ is the hyperedge set, and $\mathcal{H} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ depicts the connectivity of the hypergraph as $\mathcal{H}(v, e) = 1$ if the hyperedge e connects the vertex v otherwise $\mathcal{H}(v, e) = 0$. Furthermore, let $\mathcal{E}_v$ denote a set of related hyperedges that connect to the node v (i.e., $\mathcal{E}_v = \{e \in \mathcal{E}|\mathcal{H}(v, e) = 1\}$) and $\mathcal{V}_e$ denote a set of nodes that connect to the hyperedge e (i.e., $\mathcal{V}_e = \{v \in \mathcal{V}|\mathcal{H}(v, e) = 1\}$). In the group recommendation scenario, all members of a group together form a hyperedge, each member is a vertex, and each vertex can belong to more than one hyperedge. Therefore, we use $g \in \mathcal{E}$ to represent hyperedge instead of $e \in \mathcal{E}$ in some cases.

4 METHODOLOGY

In this section, we elaborate on AlignGroup architecture and describe each component in detail. Specifically, AlignGroup first provides a hypergraph neural network to capture both group consensus embedding and member embedding (including both users and items) via both intra-group relation learning (IntraRL) and inter-group relation learning (InterRL). To ensure that the user's personalization preferences are not compromised too much for group consensus, we further introduce a self-supervised alignment signal to steer the group consensus closer to the users' preferences. Fig 2 shows the overall architecture of AlignGroup.

4.1 Hypergraph Neural Network

In this subsection, we introduce a hypergraph neural network that aims to capture both group consensus and member preferences. Specifically, this hypergraph neural network aims to capture group consensus embedding $\dagger\mathbf{E}_g$, refined user embedding $\dagger\mathbf{E}_u$, and refined item embedding $\dagger\mathbf{E}_i$ via graph propagation.

4.1.1 Group Consensus Embedding Learning via Hypergraph Neural Network. The group consensus is determined by the joint decision of all relevant members of the group, including users and items. At the same time, there is some similarity between groups with overlapping members.

We aim to extract both intra- and inter-group relations to effectively capture the group consensus.

Intra-group Relation Learning (IntraRL): First, we consider intra-group relation learning. We represent each group as a hyperedge $g \in \mathcal{E}$, including both users and items. However, user and item nodes preserve different semantic information. Therefore, we split the aggregation process of the hypergraph neural network into user-side and item-side. We feed the initial user embedding $\mathbf{E}_u \in \mathbb{R}^{|\mathcal{U}|\times d}$, initial item embedding $\mathbf{E}_i \in \mathbb{R}^{|\mathcal{I}|\times d}$, and initial group embedding $\mathbf{E}_g \in \mathbb{R}^{|\mathcal{G}|\times d}$ to the hypergraph neural network.

Formally, for hyperedge g, we compute the message for the user/item side within group g via aggregation process:

$$\mathbf{e}_{u_p} = \mathbf{E}_u(p, :), \quad \mathbf{e}_{i_q} = \mathbf{E}_i(q, :), \tag{1}$$

$$\hat{\mathbf{e}}_{g,u} = \text{Aggr}(\mathbf{e}_{u_p}|u_p \in \mathcal{U}_g) = \sum_{u_p \in \mathcal{U}_g} \alpha_u \mathbf{e}_{u_p}, \tag{2}$$

$$\hat{\mathbf{e}}_{g,i} = \text{Aggr}(\mathbf{e}_{i_q}|i_q \in \mathcal{I}_g) = \sum_{i_q \in \mathcal{I}_g} \alpha_i \mathbf{e}_{i_q}, \tag{3}$$

where α_i and α_u are the simple average operation (i.e., $\alpha_u = 1/|\mathcal{U}_g|$ and $\alpha_i = 1/|\mathcal{I}_g|$). Then we fuse the user-side message and the item-side message through a linear transformation:

$$\hat{\mathbf{e}}_g = \text{Concat}(\hat{\mathbf{e}}_{g,u}|\hat{\mathbf{e}}_{g,i})W, \tag{4}$$

where $W \in \mathbb{R}^{2d\times d}$ denotes the trainable weight matrix for message fusion and Concat is the concatenation operation. Then we stack all $\hat{\mathbf{e}}_g (g \in \mathcal{G})$ to construct group consensus embeddings $\hat{\mathbf{E}}_g$.

Many works [12, 16, 27, 31] verify that multiple-layer Graph Convolution Networks (GCNs) can improve the expressiveness of representation. Both user/item representation and group consensus representation can benefit from high-order neighbors. To this end, we average the embedding obtained at each layer to compute the final group representation $\dagger\mathbf{E}_g$:

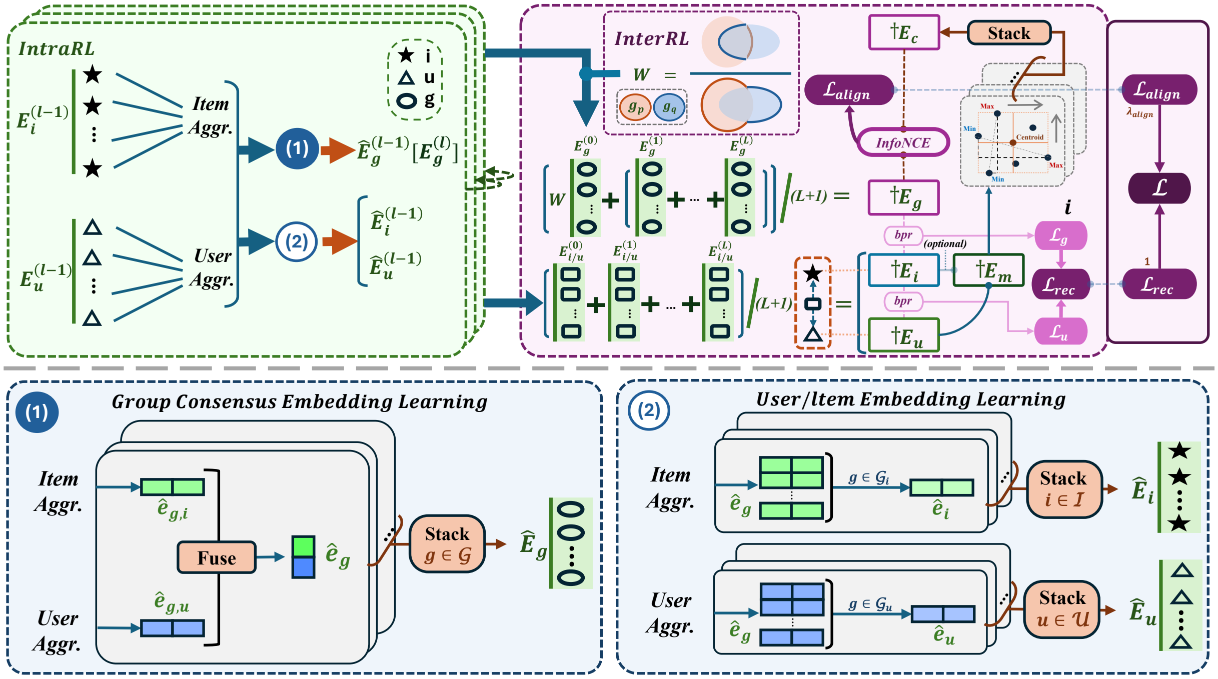


Figure 2: AlignGroup overview. We construct a hypergraph neural network for both IntraRL and InterRL to capture group consensus. We further propose a self-supervised alignment task to align group consensus and members' common preferences. Steps (1) and (2) are the group consensus embedding learning and user/item embedding learning components within our hypergraph neural network.

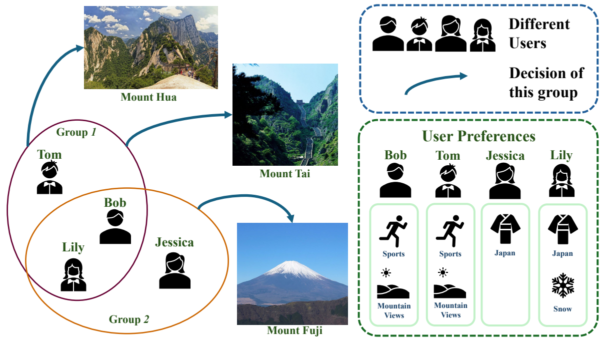


Figure 3: An illustrative example of the inter-group relations.

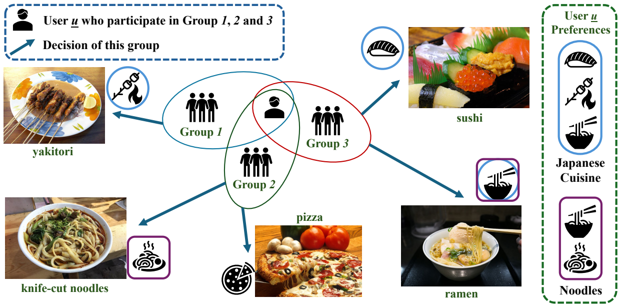


Figure 4: An illustrative example of the user preferences.

$$\mathbf{E}_g^{(l+1)} = \hat{\mathbf{E}}_g^{(l)}, \quad \dagger\mathbf{E}_g = \frac{1}{L+1}\sum_{l=0}^{L}\mathbf{E}_g^{(l)}, \tag{5}$$

where L is the total number of convolutional layers, $\hat{\mathbf{E}}_g$ is the next layer representation of $\mathbf{E}_g$.

We fully explore intra-group relations through the above steps. However, inter-group relations also deserve further consideration.

Inter-group Relation Learning (InterRL): Inspired by ConsRec [26], we noticed that groups may have their inter-group relations. For example, as depicted in Fig 3, since Bob and Tom both like sports and mountain views, group 1 always chooses to climb the steeper Mount Hua and Mount Tai. However, in group 2, Bob is the only one who likes sports, and both Jessica and Lily like Japan, so they finally choose to enjoy Mount Fuji in Japan, which has a beautiful mountain view. It means that the degree of member overlap plays a significant role in the similarity between groups. To capture the group inherent preference among similar groups, we introduce a group overlap graph to capture the similarity between groups from a member overlap perspective.

Formally, we build an adjacency matrix W_g to discriminate relevance between groups. $W_g(g_p, g_q)$ means the overlap ratio between group g_p and group g_q:

$$W_g(g_p, g_q) = \frac{|\mathcal{V}_{g_p} \cap \mathcal{V}_{g_q}|}{|\mathcal{V}_{g_p} \cup \mathcal{V}_{g_q}|}. \tag{6}$$

We further combine inter- and intra-group relations. Specifically, we utilize a group overlap graph to explore the inter-group relations together with intra-group consensus embeddings $\mathbf{E}_g$, formally:

$$\dagger\mathbf{E}_g = \frac{1}{L+1}(\sum_{l=1}^{L}\mathbf{E}_g^{(l)} + \mathbf{E}_g^{(0)} W_g). \tag{7}$$

The first layer of GCN plays the most important role, so we effectively combine inter- and intra-group relations in the first layer of GCN. We further verify the importance of InterRL through an ablation study in Section 5.3.1.

4.1.2 User/Item Embedding Learning via Hypergraph Neural Network. Intuitively, the preferences of a user or the properties of an item can be jointly modeled by the behavior of all groups to which it belongs. For example, as described in Fig 4, the preferences of user u can be modeled by group consensus. Specifically, we can infer the preference of user u for Japanese cuisine by observing their observed liking for yakitori in Group 1 and for ramen and sushi in Group 3. Additionally, the preference of user u for noodles can be deduced from their fondness for knife-cut noodles in Group 2 and ramen in Group 3. It is worth noting that the pizza in Group 2 may mean user u chooses to compromise with the group consensus.

To enhance the representation of user/item preference, we aggregate all related representations of each user/item by collecting the message from related hyperedges:

$$\hat{\mathbf{e}}_u = \mathrm{Aggr}(\hat{\mathbf{e}}_g | g \in \mathcal{E}_u) = \sum_{g\in\mathcal{E}_u} \frac{1}{|\mathcal{E}_u|} \frac{\sum_{g'\in\mathcal{E}_u}(|\mathcal{U}_{g'}|+|\mathcal{I}_{g'}|)}{|\mathcal{E}_u|(|\mathcal{U}_g|+|\mathcal{I}_g|)}\hat{\mathbf{e}}_g, \quad (8)$$

$$\hat{\mathbf{e}}_i = \mathrm{Aggr}(\hat{\mathbf{e}}_g | g \in \mathcal{E}_i) = \sum_{g\in\mathcal{E}_i} \frac{1}{|\mathcal{E}_i|} \frac{\sum_{g'\in\mathcal{E}_i}(|\mathcal{U}_{g'}|+|\mathcal{I}_{g'}|)}{|\mathcal{E}_i|(|\mathcal{U}_g|+|\mathcal{I}_g|)}\hat{\mathbf{e}}_g, \quad (9)$$

where $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_i$ means the enhanced embedding of user u and item i, respectively. In the real world, the more people in a group, the lower the correlation between group consensus and individual preferences. Therefore, we utilize group size to dynamically adjust the influence weights of different group consensus on individual preferences. $\sum_{g'\in\mathcal{E}_u}(|\mathcal{U}_{g'}|+|\mathcal{I}_{g'}|)/|\mathcal{E}_u|(|\mathcal{U}_g|+|\mathcal{I}_g|)$ and $\sum_{g'\in\mathcal{E}_i}(|\mathcal{U}_{g'}|+|\mathcal{I}_{g'}|)/|\mathcal{E}_i|(|\mathcal{U}_g|+|\mathcal{I}_g|)$ are the adjust weights. Then we stack all $\mathbf{e}_u(u \in \mathcal{U})$ and $\mathbf{e}_i(i \in \mathcal{I})$ to construct user and item embeddings $\hat{\mathbf{E}}_u$ and $\hat{\mathbf{E}}_i$, respectively.

We further enhance the expressiveness of $\hat{\mathbf{E}}_u$ and $\hat{\mathbf{E}}_i$ by GCNs. Specifically, we average the embedding obtained at each layer to get the final user representation $\dagger\mathbf{E}_u$ and item representation $\dagger\mathbf{E}_i$:

$$\mathbf{E}_u^{(l+1)} = \hat{\mathbf{E}}_u^{(l)}, \quad \dagger\mathbf{E}_u = \frac{1}{L+1}\sum_{l=0}^{L}\mathbf{E}_u^{(l)}, \quad (10)$$

$$\mathbf{E}_i^{(l+1)} = \hat{\mathbf{E}}_i^{(l)}, \quad \dagger\mathbf{E}_i = \frac{1}{L+1}\sum_{l=0}^{L}\mathbf{E}_i^{(l)}, \quad (11)$$

where L is the total number of convolutional layers, $\hat{\mathbf{E}}_u$ and $\hat{\mathbf{E}}_i$ are the next layer representation of $\mathbf{E}_u$ and $\mathbf{E}_i$, respectively.

Algorithm 1 shows the process of our hypergraph neural network.

4.2 Self-supervised Alignment Task

Unfortunately, in many cases, members have to compromise their individual preferences to reach the group consensus. To satisfy both group consensus and member preferences, we propose a self-supervised alignment task to align group consensus and members' common preferences.

Two important problems need to be identified, **a) the scope of group members**, and **b) how to calculate the common preferences of members within a group**.

Algorithm 1 Hypergraph Neural Network

```
Input: $\mathcal{U}$, $\mathcal{I}$, $\mathcal{G}$, GCN Layer number $L$, $\mathcal{V}$, $\mathcal{E}$, $\mathcal{H}$
Output: User embedding $\dagger\mathbf{E}_u$, Item embedding $\dagger\mathbf{E}_i$, Group embedding $\dagger\mathbf{E}_g$
Initialize $\mathbf{E}_u$, $\mathbf{E}_i$, $\mathbf{E}_g$;
$\mathbf{E}_u^{(0)} \leftarrow \mathbf{E}_u$; $\mathbf{E}_i^{(0)} \leftarrow \mathbf{E}_i$; $\mathbf{E}_g^{(0)} \leftarrow \mathbf{E}_g$;
for $l = 1...L$ do
    for $g \in \mathcal{G}$ do
        $\hat{\mathbf{e}}_{g,u} \leftarrow \mathrm{Aggr}(\mathbf{e}_{u_p} | u_p \in \mathcal{U}_g)$ with Eq.2;
        $\hat{\mathbf{e}}_{g,i} \leftarrow \mathrm{Aggr}(\mathbf{e}_{i_q} | i_q \in \mathcal{I}_g)$ with Eq.3;
        $\hat{\mathbf{e}}_g \leftarrow \mathrm{Concat}(\hat{\mathbf{e}}_{\mathrm{g,u}}, \hat{\mathbf{e}}_{\mathrm{g,i}})W$;
    end for
    $\hat{\mathbf{E}}_g \leftarrow \mathrm{Stack}(\hat{\mathbf{e}}_g | g \in \mathcal{G})$;
    for $u \in \mathcal{U}$ do
        $\hat{\mathbf{e}}_u \leftarrow \mathrm{Aggr}(\hat{\mathbf{e}}_g | g \in \mathcal{E}_u)$ with Eq.8;
    end for
    $\hat{\mathbf{E}}_u \leftarrow \mathrm{Stack}(\hat{\mathbf{e}}_u | u \in \mathcal{U})$;
    for $i \in \mathcal{I}$ do
        $\hat{\mathbf{e}}_i \leftarrow \mathrm{Aggr}(\hat{\mathbf{e}}_g | g \in \mathcal{E}_i)$ with Eq.9;
    end for
    $\hat{\mathbf{E}}_i \leftarrow \mathrm{Stack}(\hat{\mathbf{e}}_i | i \in \mathcal{I})$;
    $\mathbf{E}_g^{(l)} \leftarrow \hat{\mathbf{E}}_g$; $\mathbf{E}_u^{(l)} \leftarrow \hat{\mathbf{E}}_u$; $\mathbf{E}_i^{(l)} \leftarrow \hat{\mathbf{E}}_i$;
end for
for $(g_p, g_q) \in \mathcal{G}$ do
    $W_g(g_p, g_q) \leftarrow \frac{|\mathcal{V}_{g_p} \cap \mathcal{V}_{g_q}|}{|\mathcal{V}_{g_p} \cup \mathcal{V}_{g_q}|}$;
end for
$\dagger\mathbf{E}_g \leftarrow \frac{1}{L+1}(\sum_{l=1}^{L}\mathbf{E}_g^{(l)} + \mathbf{E}_g^{(0)}W_g)$;
$\dagger\mathbf{E}_u \leftarrow \frac{1}{L+1}(\sum_{l=0}^{L}\mathbf{E}_u^{(l)})$; $\dagger\mathbf{E}_i \leftarrow \frac{1}{L+1}(\sum_{l=0}^{L}\mathbf{E}_i^{(l)})$;
```

We present our solution to achieve effective alignment between group consensus and member preferences. Specifically, we introduce two different scopes of group members and two different strategies for calculating the common preferences of members within a group. We further testify these various scopes and strategies via ablation study in Section 5.3.2.

4.2.1 The Scope of Group Members. In the hypergraph neural network section, we treat users and items as members of groups equally. However, we cannot deny that user preferences can be compromised for group consensus, while properties of items are fixed properties. Therefore, we propose two scopes in computing common member preferences: **only users**, **both users and items**.

4.2.2 How to Calculate the Common Preferences of Members within a Group. For a given group $g \in \mathcal{G}$, it consists of a set of members $\mathcal{M}_g = \{m_1, ...m_k, ..., m_{|\mathcal{M}_g|}\}$, where $m_k \in \mathcal{U}_g$ or $m_k \in \mathcal{U}_g \cup \mathcal{I}_g$.

Thanks to the hypergraph neural network part, we have projected the embedding of item, user, and group into the same semantic space. Therefore, we consider defining the common preferences of group members from a physical space perspective.

Geometric Centroid: The geometric centroid for group g can be computed as the following:

$$\mathbf{e}_{g,c} = \frac{\mathrm{Max}(\dagger\mathbf{e}_{m_k} | m_k \in \mathcal{M}_g) + \mathrm{Min}(\dagger\mathbf{e}_{m_k} | m_k \in \mathcal{M}_g)}{2}, \quad (12)$$

where Max and Min both operate element-wise. $\dagger\mathbf{e}_{m_k}$ is the vector of $\dagger\mathbf{E}_{m_k} \in \dagger\mathbf{E}_u \cup \dagger\mathbf{E}_i$.

Geometric Barycenter: The geometric barycenter for group g can be computed as the following:

$$\mathbf{e}_{g,c} = \mathrm{Avg}(\dagger\mathbf{e}_{m_k} | m_k \in \mathcal{M}_g), \tag{13}$$

where Avg operate element-wise. $\dagger\mathbf{e}_{m_k}$ is the vector of $\dagger\mathbf{E}_{m_k}$.

Then we stack all $\mathbf{e}_{g,c} (g \in \mathcal{G})$ to construct members' common preferences embeddings $\dagger\mathbf{E}_c$.

4.2.3 *Loss Function.*

We adopt InfoNCE [19] to align group consensus embeddings $\dagger\mathbf{E}_g$ and members' common preferences embeddings $\dagger\mathbf{E}_c$. Formally, the alignment learning loss is defined as:

$$\mathcal{L}_{align} = \sum_{g_p \in \mathcal{G}} -\log \frac{\exp\left(\dagger\mathbf{e}_{g_p,c} \cdot \dagger\mathbf{e}_{g_p} / \tau\right)}{\sum_{g_q \in \mathcal{G}} \exp\left(\dagger\mathbf{e}_{g_q,c} \cdot \dagger\mathbf{e}_{g_q} / \tau\right)}, \tag{14}$$

where $\dagger\mathbf{e}_{g_p,c}$ and $\dagger\mathbf{e}_{g_q,c}$ are the vectors of $\dagger\mathbf{E}_c$, and $\dagger\mathbf{e}_{g_p}$ and $\dagger\mathbf{e}_{g_q}$ are the vectors of $\dagger\mathbf{E}_g$. τ is the temperature hyper-parameter.

Overview steps can be found in Algorithm 2.

Algorithm 2 Self-supervised Alignment Task

```
Input: $\mathcal{U}$, $\mathcal{I}$, $\mathcal{G}$, $\dagger\mathbf{E}_u$, $\dagger\mathbf{E}_i$, $\dagger\mathbf{E}_g$
Output: Self-supervised alignment loss $\mathcal{L}_{align}$
for $g \in \mathcal{G}$ do
    $\mathcal{M}_g \leftarrow \mathcal{U}_g$ or $\mathcal{M}_g \leftarrow \mathcal{U}_g \cap \mathcal{I}_g$;
    if Geometric Centroid then
        $\mathbf{e}_{g,c} \leftarrow$ Compute with Eq 12 # Geometric Centroid;
    else
        $\mathbf{e}_{g,c} \leftarrow$ Compute with Eq 13 # Geometric Barycenter;
    end if
end for
$\dagger\mathbf{E}_c \leftarrow \mathrm{Stack}(\mathbf{e}_{g,c} | g \in \mathcal{G})$;
$\mathcal{L}_{align} \leftarrow \mathrm{InfoNCE}(\dagger\mathbf{E}_g, \dagger\mathbf{E}_c)$ with Eq 14;
```

4.3 Optimization

We introduce our optimization strategy that jointly learns user-item and group-item interactions. Specifically, we provide a shared Multi-layer Perceptron (MLP) as a prediction function to compute both user-item final scores and group-item final scores. Formally:

$$\mathrm{MLP}(\mathbf{e}) = \mathrm{LeakyReLU}(\mathbf{e}W_1)W_2, \tag{15}$$

$$\hat{y}_{u,i} = \sigma(\mathrm{MLP}(\dagger\mathbf{e}_u \odot \dagger\mathbf{e}_i)), \quad \hat{y}_{g,i} = \sigma(\mathrm{MLP}(\dagger\mathbf{e}_g \odot \dagger\mathbf{e}_i)), \tag{16}$$

where $W_1 \in \mathbb{R}^{d \times 8}$ and $W_2 \in \mathbb{R}^{8 \times 1}$ denote the trainable weight matrices, $\dagger\mathbf{e}_u$, $\dagger\mathbf{e}_i$, and $\dagger\mathbf{e}_g$ are the vectors of $\dagger\mathbf{E}_u$, $\dagger\mathbf{E}_i$, and $\dagger\mathbf{E}_g$, respectively. σ denotes the Sigmoid function, and $\odot$ is element-wise multiplication. We choose LeakyReLU [18] to solve the neuron death problem for ReLU.

With both the user-item interaction and group-item interaction data, we utilize the Bayesian Personalized Ranking (BPR) [21] loss for optimization:

$$\mathcal{L}_u = -\sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{D}_u|} \sum_{(p,n) \in \mathcal{D}_u} \hat{y}_{u,p} - \hat{y}_{u,n}, \tag{17}$$

$$\mathcal{L}_g = -\sum_{g \in \mathcal{G}} \frac{1}{|\mathcal{D}_g|} \sum_{(p,n) \in \mathcal{D}_g} \hat{y}_{g,p} - \hat{y}_{g,n}, \tag{18}$$

where $\mathcal{D}_u$ and $\mathcal{D}_g$ denote the user-item training set sampled for user u and group-item training set sampled for group g, respectively. For each (p, n) pair, p and n denote the observed item and unobserved item, respectively.

We jointly train $\mathcal{L}_u$ and $\mathcal{L}_g$ on all the user-item and group-item interactions as:

$$\mathcal{L}_{rec} = \mathcal{L}_u + \mathcal{L}_g. \tag{19}$$

Finally, we combine the $\mathcal{L}_{rec}$ with auxiliary self-supervised alignment task $\mathcal{L}_{align}$ to jointly training:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{align}\mathcal{L}_{align}, \tag{20}$$

where λ_{align} is the balancing hyper-parameter.

5 EXPERIMENT

In this section, we conduct a comprehensive experiment to validate the effectiveness of our proposed AlignGroup. Specifically, we aim to answer the following research questions (RQs):

- RQ1: Does AlignGroup outperform the state-of-the-art group recommendation methods?
- RQ2: Whether the InterRL is necessary for AlignGroup?
- RQ3: How do different strategies and scopes of our self-supervised alignment task affect the performance?
- RQ4: Can AlignGroup effectively align group consensus and member preferences?
- RQ5: How efficient is AlignGroup compared with other group recommendation methods?
- RQ6: How do different hyper-parameter settings impact the performance of AlignGroup?

Table 1: Statistics of the experimental datasets.

Dataset	# Users	# Items	# Groups	# U-I interactions	# G-I interactions	Avg. group size
Mafengwo	5275	1513	995	39761	3595	7.19
CAMRa2011	602	7710	290	116344	145068	2.08

5.1 Experimental Settings

5.1.1 *Datasets.*

To evaluate our AlignGroup[1] in both group and user recommendation tasks, we conduct extensive experiments on two real-world public datasets, Mafengwo and CAMRa2011 [4].

1. Mafengwo: The Mafengwo dataset encompasses travel records from a social tourism platform, detailing 5,275 users, 995 groups, and 1,513 locations with interactions between users and visited sites. This dataset was curated by selecting groups with at least two members who have visited three or more venues, averaging 7.19 users per group.

2. CAMRa2011: The CAMRa2011 dataset provides a collection of movie ratings from individual and household users, comprising 602 users in 290 groups with a total of 7,710 movies rated on a scale from 0 to 100. To focus on communal viewing preferences, only users belonging to groups were retained, with ratings converted into binary feedback, resulting in 116,344 user-item and 145,068 group-item interactions, with an average group size of 2.08.

The key statistics of these two datasets are delineated in Table 1.

[1] The code is available on https://github.com/Jinfeng-Xu/AlignGroup.

Table 2: Performance comparison of baselines and AlignGroup on group recommendation in terms of H@K and N@K. The superscript * indicates the improvement is statistically significant where the p-value is less than 0.01.

Dataset	Metric	Pop	NCF	AGREE	HyperGroup	HCR	GroupIM	S^2-HHGR	CubeRec	ConsRec	AlignGroup	*Improv.*
Mafengwo	H@5	0.3115	0.4701	0.4729	0.5739	0.7759	0.7377	0.7568	0.8613	0.8844	**0.9678***	9.43%
	H@10	0.4251	0.6269	0.6321	0.6482	0.8503	0.8161	0.7779	0.9025	0.9156	**0.9688***	5.81%
	N@5	0.2169	0.3657	0.3694	0.4777	0.6611	0.6078	0.7322	0.7574	0.7692	**0.8817***	14.63%
	N@10	0.2537	0.4141	0.4203	0.5018	0.6852	0.6330	0.7391	0.7708	0.7794	**0.8821***	13.18%
CAMRa2011	H@5	0.4324	0.5803	0.5879	0.5890	0.5883	0.6552	0.6062	0.6400	0.6407	**0.8193***	25.05%
	H@10	0.5793	0.7693	0.7789	0.7986	0.7821	0.8407	0.7903	0.8207	0.8248	**0.8448**	0.49%
	N@5	0.2825	0.3896	0.3933	0.3856	0.4044	0.4310	0.3853	0.4346	0.4358	**0.8180***	87.70%
	N@10	0.3302	0.4448	0.4530	0.4538	0.4670	0.4914	0.4453	0.4935	0.4945	**0.8259***	67.02%

Table 3: Performance comparison of baselines and AlignGroup on user recommendation in terms of H@K and N@K. The superscript * indicates the improvement is statistically significant where the p-value is less than 0.01.

Dataset	Metric	Pop	NCF	AGREE	HyperGroup	HCR	GroupIM	S^2-HHGR	CubeRec	ConsRec	AlignGroup	*Improv.*
Mafengwo	H@5	0.4047	0.6363	0.6357	0.7235	0.7571	0.1608	0.6380	0.1847	0.7725	**0.8159***	5.62%
	H@10	0.4971	0.7417	0.7403	0.7759	0.8290	0.2497	0.7520	0.3734	0.8404	**0.8643***	2.84%
	N@5	0.2876	0.5432	0.5481	0.6722	0.6703	0.1134	0.4637	0.1099	0.6884	**0.7605***	10.47%
	N@10	0.3172	0.5733	0.5738	0.6894	0.6937	0.1420	0.5006	0.1708	0.7107	**0.7763***	9.23%
CAMRa2011	H@5	0.4624	0.6119	0.6196	0.5728	0.6262	0.6113	0.6153	0.5754	0.6774	**0.8116***	19.81%
	H@10	0.6026	0.7894	0.7897	0.7601	0.7924	0.7771	0.8173	0.7827	**0.8412**	0.8382	-
	N@5	0.3104	0.4018	0.4098	0.4410	0.4195	0.4064	0.3978	0.3751	0.4568	**0.8083***	76.95%
	N@10	0.3560	0.4535	0.4627	0.5016	0.4734	0.4606	0.4641	0.4428	0.5104	**0.8167***	60.01%

5.1.2 *Evaluation Metrics.*

To evaluate the recommendation task performance fairly, we adopt two well-established metrics following previous settings [4, 5]: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). We report the average metrics of all users in the test dataset under HR@5 (H@5), HR@10 (H@10), NDCG@5 (N@5), and NDCG@10 (N@10).

5.1.3 *Baselines.*

We compare AlignGroup against the following state-of-the-art recommendation methods:

- **Popularity (Pop)** [8] is a non-personalized method that ranks items for users and groups according to how often items are interacted with, providing a non-personalized standard for comparing personalized recommendation methods.
- **NCF** [13] is adopted as treating each group as a virtual user for prediction.
- **AGREE** [4] is a group recommendation method that uses an attention mechanism to aggregate user preferences within a group.
- **HyperGroup** [10] treats a group as a hyperedge and introduces a method for learning group representations based on hyperedge embeddings.
- **HCR** [15] is a notable group recommendation model that uses a dual-channel hypergraph convolutional network to analyze both member and group preferences.
- **GroupIM** [22] combines user preferences into group preferences using attention and includes self-supervised learning to address data sparsity by linking users with their groups.
- **S^2-HHGR** [32] is a state-of-the-art method that combines hypergraph and self-supervised learning for improved prediction of group preferences.
- **CubeRec** [6] utilizes the geometric expressiveness of hypercubes to adaptively aggregate members' interests.
- **ConsRec** [26] reveals the group consensus by aggregate member-level, item-level, and group-level views.

5.1.4 *Implementation Details.*

Following the basic settings of ConsRec [26], we implement AlignGroup in PyTorch and optimize with Adam optimizer. We apply Xavier initialization [9] for all initial random embedding. For hidden layers, we randomly initialize their parameters with a Gaussian distribution of a mean of 0 and a standard deviation of 0.1. As for hyper-parameter settings on AlignGroup, we perform a grid search on the number of convolutional layers L in {1, 2, 3, 4}, temperature hyper-parameter τ in {0.2, 0.4, 0.6, 0.8}, and the balancing hyper-parameter λ_{align} in {1e-1, 1e-2, 1e-3}. We fix the learning rate with 1e-3.

5.2 Overall Performance (RQ1)

Table 2 and Table 3 show the performance of our proposed AlignGroup and other baseline methods on the group recommendation and user recommendation tasks, respectively. We have the following key observations:

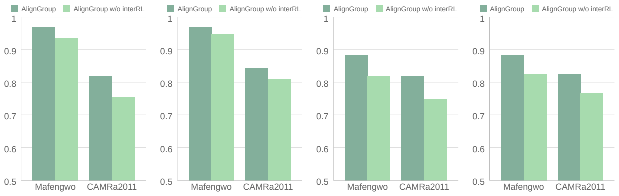
- For the group recommendation task shown in Table 2, AlignGroup achieves higher performance than all baselines, which demonstrates the effectiveness of our AlignGroup. We attribute this to the excellent ability of our IntraRL and InterRL in hypergraph neural networks to capture group consensus. In addition, we note that the improvement of AlignGroup compared to other baselines is huge for NDCG, which is more concerned with fine-grained ranking order than HR. Therefore, we believe this implies that merely obtaining group consensus lacks fine-grained inference over group behavior, such as ConsRec and CubeRec. On the contrary, our self-supervised alignment task can effectively capture the fine-grained behavior of the group by aligning the group consensus with the members' common preferences.
- For the user recommendation task shown in Table 3, AlignGroup still outperforms all baselines under most evaluation metrics. We owe our superiority to the proposed self-supervised alignment task and our group user jointly optimization strategy. The poor

performance of GroupIM and CubeRec verifies the effectiveness of the joint optimization strategy compared with the two-stage optimization strategy.

- It is worth mentioning that hypergraph-based methods (HyperGroup and ConsRec) generally perform better than traditional aggregation methods (AGREE), thanks to the excellent representation capabilities of hypergraphs. Also, GroupIM benefits from the self-supervised task to achieve excellent performance. However, our AlignGroup achieves the best performance with excellent motivations that are more in line with the real world.

5.3 Ablation Study (RQ2 & RQ3 & RQ4)

In this section, we conduct exhaustive experiments to examine the effectiveness of various modules in our AlignGroup.



(a) HR@5 (b) HR@10 (c) NDCG@5 (d) NDCG@10

Figure 5: Histograms in dark green show the performance of AlignGroup with both IntraRL and InterRL, and histograms in light green show the performance of AlignGroup w/o InterRL with only IntraRL. The performance metrics are HR@5, HR@10, NDCG@5, and NDCG@10 on both Mafengwo and CAMRa2011 datasets.

5.3.1 *Effectiveness of InterRL (RQ2).*

To verify the effectiveness of the InterRL in our hypergraph neural network, we design the following variant of AlignGroup.

AlignGroup w/o InterRL: We remove the InterRL part and only retain intra-group relation learning. Formally, we utilize Eq 5 to compute group representation $\dagger E_g$ instead of Eq 7.

As shown in Fig 5, we can observe that removing InterRL degrades the performance, showing that both intra-group and inter-group relations play a distinct role in group consensus learning.

5.3.2 *Performance of Different Strategies and Scopes of Self-supervised Alignment Task (RQ3).*

To analyze the performance of different strategies and scopes of our self-supervised alignment task, we introduce an ablation study. In terms of scope, we refer to considering only the user as "small" and considering both the user and item as "big". Therefore we try to combine two different scopes and two different strategies for computing common preferences:

Centroid-small: We choose **geometric centroid** as common preferences for each group including **small** scope members.

Centroid-big: We choose **geometric centroid** as common preferences for each group including **big** scope members.

Barycenter-small: We choose **geometric barycenter** as common preferences for each group including **small** scope members.

Barycenter-big: We choose **geometric barycenter** as common preferences for each group including **big** scope members.

Table 4: Ablation study on different strategies and scopes of self-supervised alignment task with both group and user recommendation results reported.

Task	Dataset	Variant	H@5	H@10	N@5	N@10
Group	Mafengwo	Centroid-small	**0.9678**	**0.9688**	**0.8817**	**0.8821**
		Centroid-big	0.9075	0.9286	0.7755	0.7825
		Barycenter-small	0.9332	0.9401	0.8371	0.8529
		Barycenter-big	0.8999	0.9102	0.7493	0.7678
	CAMRa2011	Centroid-small	**0.8193**	**0.8448**	**0.8180**	**0.8259**
		Centroid-big	0.6572	0.8172	0.5292	0.5812
		Barycenter-small	0.8002	0.8285	0.7892	0.8021
		Barycenter-big	0.6428	0.8071	0.5246	0.5763
User	Mafengwo	Centroid-small	0.8159	0.8643	0.7605	0.7763
		Centroid-big	**0.8660**	**0.9037**	**0.8294**	**0.8416**
		Barycenter-small	0.7889	0.8273	0.7297	0.7382
		Barycenter-big	0.8525	0.8891	0.8113	0.8302
	CAMRa2011	Centroid-small	**0.8116**	**0.8382**	**0.8083**	**0.8167**
		Centroid-big	0.6867	0.8239	0.5962	0.6407
		Barycenter-small	0.7884	0.8491	0.7429	0.7601
		Barycenter-big	0.6612	0.8091	0.5817	0.6287

We report the experimental results in Table 4. Selecting the geometric centroid to calculate shared member preferences yields superior results compared to using the geometric barycenter for both group and individual user recommendation tasks across all evaluation metrics. In most cases, the small scope outperforms the big scope. However, for user recommendation tasks, the big scope shows better results on the Mafengwo dataset. We speculate that this is because the Mafengwo dataset has far fewer items than users, and taking items into common preference computation can make the semantics of users and items more similar, which benefits user recommendation tasks. On the other hand, for the CAMRa2011 dataset, where the number of items greatly exceeds that of users, taking items into common preference computation may significantly alter the semantic representation of users, leading to poorer performance in user recommendation tasks.

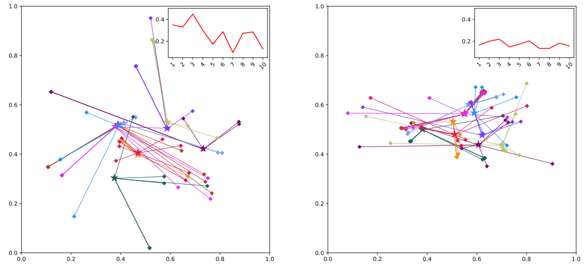
5.3.3 *Effectiveness of Self-supervised Alignment Task (RQ4).*

We further verify the effectiveness of our self-supervised alignment task through visualization.

Specifically, we randomly sample all users in 10 groups from the Mafengwo dataset. Then, we utilize t-SNE [23] to map their embedding to the 2-dimension space. In Fig 6, we observe that our self-supervised alignment task effectively reduces the gap between member preferences and group consensus. To show this result more clearly, we average the gap between group consensus and member preferences within the group in the upper right corner of Fig 6. Our self-supervised alignment task effectively reduces the average gap between group consensus and member preferences.

5.4 Efficiency Study (RQ5)

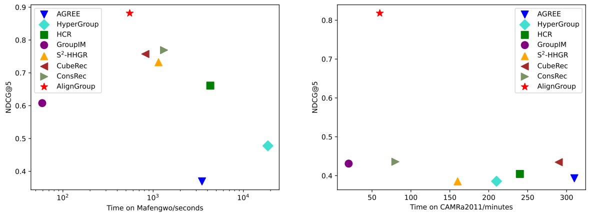
Following ConsRec [26] settings, we estimate the efficiency of AlignGroup by directly comparing the total running time (including both training and testing) with all baselines. Fig 7 illustrates the performance (NDCG@5) and runtime (in seconds or minutes) for the group recommendation task on two experimental datasets. Our proposed AlignGroup achieves exceptional efficiency and performance, outperforming all baseline methods in terms of efficiency—with the sole exception of GroupIM—and significantly surpassing all baseline methods in performance.



(a) AlignGroup w/o self-supervised alignment task

(b) AlignGroup

Figure 6: Visualization of (a) AlignGroup w/o self-supervised alignment task and (b) AlignGroup. Pentagram and Prism denote the group consensus and member preferences, respectively. The same color indicates the group consensus and member preferences of the same group. The line graph in the upper right corner represents the average gap between member preferences and group consensus for each of the 10 groups. It shows that AlignGroup in (b) can align member preferences with group consensus much better.



(a) Mafengwo

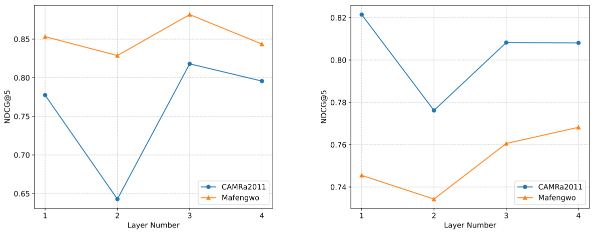
(b) CAMRa2011

Figure 7: Efficiency Study in terms of NDCG@5 on both Mafengwo and CAMRa2011 datasets.

5.5 Hyper-parameter Analysis (RQ6)

5.5.1 Effect of the Layer Number L of the Hypergraph Neural Network. Layer number affects the performance of the hypergraph neural network. More layers can aggregate more information from high-order neighbors. However, as the layer number increases, it faces the over-smoothing problem [17] where the discriminative of node representations is insufficient. Fig 8 illustrates the performance trends of AlignGroup with different settings of L. Considering both the group recommendation and user recommendation tasks, $L = 3$ is the optimal setting on both Mafengwo and CAMRa2011 datasets.

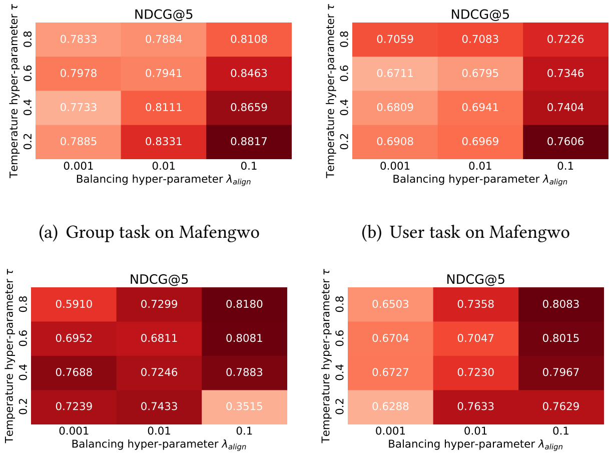
5.5.2 Effect of the Balancing Hyper-parameters λ_{align} and τ of the Self-supervised Alignment Task. The balancing hyper-parameter λ_{align} and the temperature hyper-parameter τ jointly control the degree of our self-supervised alignment task. We tune the λ_{align} from {1e-1,1e-2,1e-3}, and τ from {0.2, 0.4, 0.6, 0.8}. Fig. 9 shows that the best performances for both group recommendation and user recommendation tasks are achieved with λ_{align} = 1e-1 and τ = 0.2 on Mafengwo dataset. On CAMRa2011 dataset, the optimal result is achieved with λ_{align} = 1e-1, τ = 0.8.



(a) Group recommendation task

(b) User recommendation task

Figure 8: Performance of AlignGroup with respect to different layer numbers L of hypergraph neural network.



(a) Group task on Mafengwo

(b) User task on Mafengwo

(c) Group task on CAMRa2011

(d) User task on CAMRa2011

Figure 9: Performance of AlignGroup with respect to different hyper-parameter pairs (λ_{align},τ). Darker color denotes better performance of recommendation.

6 CONCLUSION

In this paper, we propose a novel group recommendation method, named AlignGroup. It extracts group consensus by learning intra- and inter-group relations and further refining the group decision-making by aligning group consensus and members' common preferences. In particular, AlignGroup effectively captures the group consensus from both intra- and inter-group relations by a well-designed hypergraph neural network. Moreover, AlignGroup proposes a novel and simple self-supervised alignment task to capture fine-grained group decision-making by aligning the group consensus with members' common preferences. Experimental results on two real-world datasets rigorously verify the effectiveness and efficiency of AlignGroup for both group and user recommendation tasks.

ACKNOWLEDGMENTS

This work was supported by the Hong Kong UGC General Research Fund no. 17203320 and 17209822, and the project grants from the HKU-SCF FinTech Academy.

REFERENCES

[1] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. 2009. Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment* 2, 1 (2009), 754–765.

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

[3] Ludovico Boratto and Salvatore Carta. 2011. State-of-the-art in group recommendation and new approaches for automatic identification of groups. In *Information retrieval and mining in distributed environments*. Springer, 1–20.

[4] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive group recommendation. In *The 41st International ACM SIGIR conference on research & development in information retrieval*. 645–654.

[5] Da Cao, Xiangnan He, Lianhai Miao, Guangyi Xiao, Hao Chen, and Jiao Xu. 2019. Social-enhanced attentive group recommendation. *IEEE Transactions on Knowledge and Data Engineering* 33, 3 (2019), 1195–1209.

[6] Tong Chen, Hongzhi Yin, Jing Long, Quoc Viet Hung Nguyen, Yang Wang, and Meng Wang. 2022. 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*. 1664–1673.

[7] Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 891–900.

[8] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. 39–46.

[9] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[10] Lei Guo, Hongzhi Yin, Tong Chen, Xiangliang Zhang, and Kai Zheng. 2021. Hierarchical hyperedge embedding-based representation learning for group recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 1 (2021), 1–27.

[11] Lei Guo, Hongzhi Yin, Qinyong Wang, Bin Cui, Zi Huang, and Lizhen Cui. 2020. Group recommendation with latent voting mechanism. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 121–132.

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. 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*. 639–648.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[14] Zhixiang He, Chi-Yin Chow, and Jia-Dong Zhang. 2020. GAME: Learning graphical and attentive multi-view embeddings for occasional group recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 649–658.

[15] Renqi Jia, Xiaofei Zhou, Linhua Dong, and Shirui Pan. 2021. Hypergraph convolutional network for group recommendation. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 260–269.

[16] Yangqin Jiang, Chao Huang, and Lianghao Huang. 2023. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 4252–4261.

[17] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[18] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. Atlanta, GA, 3.

[19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[20] H Ren, W Hu, and J Leskovec. 2020. Query2box: Reasoning Over Knowledge Graphs In Vector Space Using Box Embeddings. In *International Conference on Learning Representations (ICLR)*.

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[22] Aravind Sankar, Yanhong Wu, Yuhang Wu, Wei Zhang, Hao Yang, and Hari Sundaram. 2020. 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*. 1279–1288.

[23] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[24] Lucas Vinh Tran, Tuan-Anh Nguyen Pham, Yi Tay, Yiding Liu, Gao Cong, and Xiaoli Li. 2019. 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*. 255–264.

[25] Yinwei Wei, Wenqi Liu, Fan Liu, Xiang Wang, Liqiang Nie, and Tat-Seng Chua. 2023. Lightgt: A light graph transformer for multimedia recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1508–1517.

[26] Xixi Wu, Yun Xiong, Yao Zhang, Yizhu Jiao, Jiawei Zhang, Yangyong Zhu, and Philip S Yu. 2023. Consrec: Learning consensus behind interactions for group recommendation. In *Proceedings of the ACM Web Conference 2023*. 240–250.

[27] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Hewei Wang, and Edith C-H Ngai. 2024. MENTOR: Multi-level Self-supervised Learning for Multimodal Recommendation. *arXiv preprint arXiv:2402.19407* (2024).

[28] Qi Yang, Sergey Nikolenko, Alfred Huang, and Aleksandr Farseev. 2022. Personality-Driven Social Multimedia Content Recommendation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 7290–7299.

[29] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social influence-based group representation learning for group recommendation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 566–577.

[30] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, and Xiaofang Zhou. 2020. Overcoming data sparsity in group recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 7 (2020), 3447–3460.

[31] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[32] Junwei Zhang, Min Gao, Junliang Yu, Lei Guo, Jundong Li, and Hongzhi Yin. 2021. Double-scale self-supervised hypergraph learning for group recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 2557–2567.

[33] Shuai Zhang, Huoyu Liu, Aston Zhang, Yue Hu, Ce Zhang, Yumeng Li, Tanchao Zhu, Shaojian He, and Wenwu Ou. 2021. Learning user representations with hypercuboids for recommender systems. In *Proceedings of the 14th ACM international conference on web search and data mining*. 716–724.