

Double-Scale Self-Supervised Hypergraph Learning for Group Recommendation

Junwei Zhang

Chongqing University
jw.zhang@cqu.edu.cn

Lei Guo

Shandong Normal University
leiguo.cs@gmail.com

Min Gao*

Chongqing University
gaomin@cqu.edu.cn

Junliang Yu

The University of Queensland
jl.yu@uq.edu.au

Jundong Li

University of Virginia
jl6qk@virginia.edu

Hongzhi Yin

The University of Queensland
h.yin1@uq.edu.au

ABSTRACT

With the prevalence of social media, there has recently been a proliferation of recommenders that shift their focus from individual modeling to group recommendation. Since the group preference is a mixture of various predilections from group members, the fundamental challenge of group recommendation is to model the correlations among members. Existing methods mostly adopt heuristic or attention-based preference aggregation strategies to synthesize group preferences. However, these models mainly focus on the pairwise connections of users and ignore the complex high-order interactions within and beyond groups. Besides, group recommendation suffers seriously from the problem of data sparsity due to severely sparse group-item interactions. In this paper, we propose a self-supervised hypergraph learning framework for group recommendation to achieve two goals: (1) capturing the intra- and inter-group interactions among users; (2) alleviating the data sparsity issue with the raw data itself. Technically, for (1), a hierarchical hypergraph convolutional network based on the user- and group-level hypergraphs is developed to model the complex tuplewise correlations among users within and beyond groups. For (2), we design a double-scale node dropout strategy to create self-supervision signals that can regularize user representations with different granularities against the sparsity issue. The experimental analysis on multiple benchmark datasets demonstrates the superiority of the proposed model and also elucidates the rationality of the hypergraph modeling and the double-scale self-supervision.

CCS CONCEPTS

- Information systems → Recommender systems.

KEYWORDS

Group Recommendation, Graph Neural Networks, Hypergraph Learning, Self-Supervised Learning

*Min Gao is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8446-9/21/11...\$15.00

<https://doi.org/10.1145/3459637.3482426>

ACM Reference Format:

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 and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3459637.3482426>

1 INTRODUCTION

Owing to the rapid development of social media in recent years, users with similar interests now have opportunities to form various online groups [34, 43]. Traditional recommender systems designed for individuals can no longer serve the interest of groups. There is an urgent need to develop practical group recommender systems [3, 6, 50]. The group recommendation aims to reach a consensus among group members, generating suggestions that can cater to most group members [21, 29]. However, the influences of users vary from group to group, leading to hard decision-making. An illustrative example is shown in Figure 1: Alice and her colleagues (Bob and Tony) form a group that are fond of slow food. However, when Alice and her boss Allen (who likes fast food) go for lunch, fast food would be the first choice due to Allen's higher position. Meanwhile, since Allen has an appetite for pizza, pizza is also recommended to Alice and Bob. It is natural that collective decisions tend to be dynamic, i.e., a group's preference may vary due to the influence of other user members and other groups' preferences. As such, a critical issue is to model the complex interactions of users and groups in the research of group recommendations.

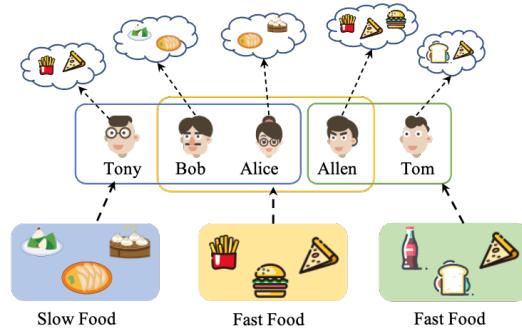


Figure 1: An illustrative example of group recommendation.

However, existing group recommendation models fail to model the interactions of users and groups well. Many of them adopt

heuristic rules, such as average [3, 4], least misery [1], and maximum satisfaction [5], which ignore the impact of user interactions. With the success of deep learning, several group recommendation models employ the attention mechanism over user-item graphs to exploit user interactions and synthesize group preferences [20, 35, 44]. Nevertheless, these models keep attention on the pairwise connections of users and ignore the complex high-order interactions within and beyond groups. Besides, since many groups are formed temporally, the group-item interaction records are very sparse, making it more difficult to learn an accurate group preference directly from the interaction data [20, 44, 45]. To tackle the aforementioned two limitations, in this paper, we propose a self-supervised hypergraph learning framework for group recommendation to achieve two goals: (1) capturing the intra- and inter-group interactions among users; (2) alleviating the data sparsity issue with the raw data itself. To handle the interaction issue, we explicitly model the complex tuplewise relationships as a hierarchical hypergraph, which aggregates the correlated users within and beyond groups. Distinct from the edge in simple graphs, the hyperedge can connect multiple nodes, which is a natural way to capture tuplewise relations. The proposed hierarchical hypergraph consists of two levels: user- and group-level, where each level of hypergraphs contains the corresponding type of nodes. In the user-level hypergraph, each hyperedge connects the set of users belonging to the same group. To capture the intra-group interactions among different users, we adopt the hypergraph convolutional network to aggregate the related users and generate the dynamic user embeddings. Meanwhile, for exploiting the informative inter-group interactions among users, we adopt triangular motifs [55] which allow each group to select the most relevant groups to acquire information to form the hyperedges in the group-level hypergraph.

Despite the benefits of hypergraph modeling, the group recommendation performance is still compromised by the data sparsity problem, which not only exists in group interaction but also in user interaction. To alleviate this issue, we integrate self-supervised learning (SSL), which can discover self-supervision signals from the raw data, into the training of the hierarchical hypergraph convolutional network. Many existing SSL-based models commonly conduct node/edge dropout in the user-item graph to augment supervised signals [16, 31, 47]. However, this may drop some essential information, especially when the raw group is very sparse, leading to unserviceable supervision signals. Therefore, we propose a double-scale node dropout strategy to create supervisory labels with different and finer granularities. The intuition behind is that not all user-user interactions significantly affect the final group decision, and leveraging partial information can help learn invariant representations. Meanwhile, two different-grained views may to some degree prevent the loss of essential information (e.g. drop important nodes). Concretely, at a coarse granularity, we randomly remove some nodes from the user-level hypergraph structure. As a result, when a removed node belongs to multiple groups, it will disappear in all related hyperedges. As for the fine-grained node dropout, we remove some member nodes only from a specific group, which does not affect this node's existence in other groups. We then maximize the mutual information between the node representations learned from two granularities to regularize the user and group representation against the sparsity issue. Finally, we unify

the recommendation task and the self-supervised learning task to optimize model parameters by a two-stage training approach.

In summary, the main contributions of this paper are as follows:

- We devise a hierarchical hypergraph learning framework to capture the intra- and inter-group interactions among users in the group recommendation.
- We propose a SSL strategy with different granularities to enhance user and group representations and alleviate the data sparsity problem, which is seamlessly coupled with the hierarchical design of the hypergraph convolutional network.
- We conduct extensive experiments on three group recommendation datasets to exhibit the superiority of the proposed model over some recent baselines and elucidate why the hypergraph learning and the double-scale SSL strategy can improve group recommendation. The code implementation of our model is released at <https://github.com/0411tony/HGGR>.

The remainder of this paper is organized as follows. We first provide preliminaries of the hypergraph and the definition of the group recommendation in Section 2. Then, we present our proposed model in detail in Section 3. Section 4 describes experimental research. Next, we further discuss the related work in Section 5. Finally, we summarize our work and look forward to future work in Section 6.

2 PRELIMINARIES

To facilitate understanding, we present the definition of the hypergraph and the task of the group recommendation in this section.

2.1 Definition of Hypergraph

The hypergraph is defined as $G = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the vertex set containing M unique vertices, and \mathcal{E} is the hyperedge set containing N hyperedges. Each hyperedge $\epsilon \in \mathcal{E}$ contains multiple vertices. The hypergraph can be represented by an incidence matrix $\mathbf{H} \in \mathbb{R}^{M \times N}$, where $\mathbf{h}_{ve} = 1$ if the hyperedge ϵ contains the vertex $v \in \mathcal{V}$, otherwise $\mathbf{h}_{ve} = 0$. For each hypergraph, we use the diagonal matrix \mathbf{D} to denote the degrees of vertex, where $\mathbf{d}_v = \sum_{\epsilon=1}^N \mathbf{w}_\epsilon \mathbf{h}_{ve}$. $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the diagonal matrix of the weight of the hyperedge. Let diagonal matrix \mathbf{B} denote the degree of hyperedge, where $\mathbf{b}_\epsilon = \sum_{v=1}^M \mathbf{h}_{ve}$ represents the number of vertices connected by the hyperedge ϵ .

2.2 Task of Group Recommendation

The task of the group recommendation is to predict the item ranking list for the given group that has multiple members. We consider $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ and $\mathcal{I} = \{i_1, i_2, \dots, i_{|\mathcal{I}|}\}$ to denote the user set and item set, respectively. Let $\mathcal{G} = \{g_1, g_2, \dots, g_{|\mathcal{G}|}\}$ be the group set, where $g_m \in \mathcal{G}$ is the m -th group. There are three observable interaction behaviors: user-item interaction, group-item interaction, and user-group interaction. Let $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ denote the user-item interaction matrix, where $r_{ui} = 1$ if the user u consumed item i , otherwise $r_{ui} = 0$. $\mathbf{S} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{I}|}$ denotes the group-item interaction matrix, where $s_{gi} = 1$ if the group g consumed the item i , otherwise $s_{gi} = 0$. We embed each user u into the space vector $\mathbf{p}_u \in \mathbb{P}^{|\mathcal{U}| \times d}$ of the dimension d . Let $\mathbf{q}_i \in \mathbb{Q}^{|\mathcal{I}| \times d}$ represent the vector representation of item i , which is randomly initialized with

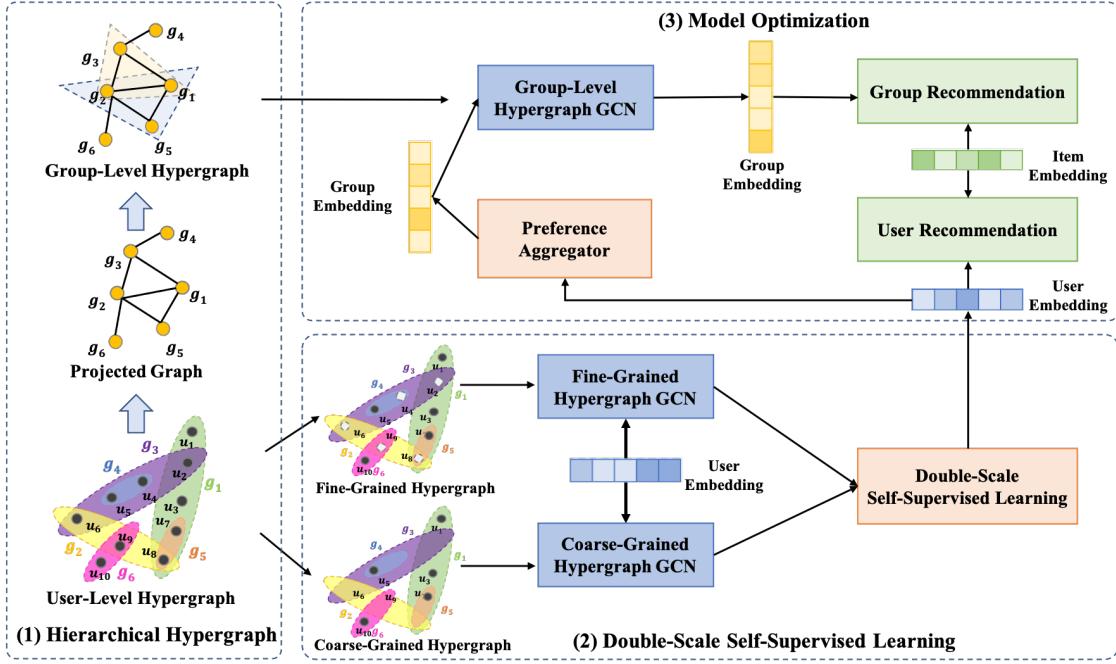


Figure 2: The overview of our proposed framework for group recommendation.

a d -dimensional vector, $z_g \in Z^{|\mathcal{G}| \times d}$ denotes the representation of group g . We use bold capital letters (e.g., X) and bold lowercase letters (e.g., x) to represent matrices and vectors, respectively.

3 THE PROPOSED METHOD

In this section, we first introduce the details of the proposed model **HHGR** (short for "Hierarchical Hypergraph Learning Framework for Group Recommendation"). Then, we present its self-supervised variant, **S²-HHGR**, with the double-scale dropout strategy. Finally, we conduct the complexity analysis on **S²-HHGR** to demonstrate its scalability. The proposed framework is shown in Figure 2, which includes three components: 1) Hierarchical hypergraph, which captures the user interactions within and beyond groups by propagating information from user level to group level; 2) Double-scale self-supervised learning, which contains coarse- and fine-grained node dropout strategies to refine the user and group representations and alleviate the data sparsity problem; and 3) Model optimization, which unifies the objectives of group recommendation and self-supervised learning to enhance both tasks.

3.1 Hierarchical Hypergraph Convolution

By organizing users and groups in a hierarchical hypergraph, we can leverage their connectives to help exhibit high-order tuplewise user interactions within and beyond groups, which is of crucial importance to group recommendation. Inspired by [10, 23], we devise a new hierarchical hypergraph convolution to perform the embedding propagation mechanism over our hierarchical hypergraph on two levels, user-level and group-level. In what follows, we elaborate on these two ingredients.

3.1.1 User-level hypergraph. Hypergraph construction. Since users are correlated within and beyond groups in group recommendation scenario, it is vital to define appropriate connections among them and exploit the users' mutual interactions. The conventional graph structure can only support pairwise relations between users, which is not fit for this case. Thus, we propose to model such user interactions with a hypergraph, which is shown in the bottom-left corner of Figure 2. In the user-level hypergraph G_{ul} , multiple users can be connected through one hyperedge (or group).

User representation learning. We aim to exploit the tuplewise user interactions for learning their influences, in which the correlated users should be affected by the intra-group users. To achieve that, we introduce the hypergraph convolution operation to capture user interactions among their neighbors and learn each user's dynamic representation. For the specific user node v_u , the hypergraph convolutional network first learns the representations of all its connected hyperedges e_j , which is from the gathered node features. Then the node v_u 's representation would integrate the related hyperedge feature information. Simple graphs use the adjacency matrix to represent relationships between two connected nodes, whereas hypergraphs introduce the incidence matrix H_{ul} to describe the relationship between nodes and hyperedges, which is defined in Section 2.1. Referring to the spectral hypergraph convolution proposed in HGNN [10, 38], the hypergraph convolution is defined as follows:

$$P^{(l+1)} = \sigma(D_{ul}^{-1} H_{ul} W_{ul} B_{ul}^{-1} H_{ul}^T P^{(l)} \Theta^{(l)}), \quad (1)$$

where D_{ul} denotes the vertex degree matrix of the user-level hypergraph, and B_{ul} denotes the hyperedge degree matrix. W_{ul} is regarded as the weight matrix of the hyperedges. In this paper, we initialize the weight matrix W_{ul} with the identity matrix yielding

equal weights for all hyperedges. D_{ul} and B_{ul} play a role of the normalization in hypergraph convolutional operator. Θ is the parameter matrix between two convolutional layers. $P^{(l)}$ is the users' embeddings in the l -th hypergraph convolutional network, where $P^{(0)} = \tilde{P}$. \tilde{P} is the initial vectors of all users u with d -dimension. Since the nonlinear activation is found to be redundant in recent research [38], we remove this part in Eq. (1), which is shown as follows:

$$P^{(l+1)} = D_{ul}^{-1} H_{ul} B_{ul}^{-1} H_{ul}^T P^{(l)} \Theta^{(l)}. \quad (2)$$

Group representation learning. Our target is to obtain the group embeddings to estimate their preferences on items. Since group members have different importance, we perform a weighted sum on the representations of user members to generate the attentive group representation \tilde{Z} . The weights can reflect the users' contribution to the group's decision-making.

$$\tilde{z}_g = \sum_{u \in g_m} \alpha_u p_u W_{agg}, \quad (3)$$

$$\alpha_u = \frac{\exp(p_u W_{agg} x^T)}{\sum_{j \in g_m} \exp(p_j W_{agg} x^T)}, \quad (4)$$

where α_u is the weight of a user u in the group decision, p_j denotes other members vectors in the same group g , and $W_{agg} \in \mathbb{R}^{d \times d}$ and $x \in \mathbb{R}^d$ are parameters used to compute the weights.

3.1.2 Group-level hypergraph. Hypergraph construction. A user often belongs to multiple groups, which means she is connected with other users of different groups. However, not all user interactions affect user preferences. To select informative user interactions and optimize inter-group interactions among users, we adopt the triadic motif relation as the motif-induced hyperedge to construct the group-level hypergraph. Since the triadic motif can increase homophily cohesion in the social community [11, 26], its interaction is a stronger bond, which is highly beneficial to modeling user interactions. Specifically, we first convert the user-level hypergraph to a projected graph, where group hyperedges act as nodes. In the projected graph, if two group hyperedges have common user members, they will be connected, ensuring the relevance of preferences between groups. Then, we adopt the triadic motif to select the most relevant groups in the group-level hypergraph. If any three groups conform to the definition of the motif, we will define these three groups to be divided into one motif-induced hyperedge. The group-level hypergraph is represented as G_{gl} .

Group representation learning. Intuitively, the motif can be regarded as a closed path connecting three different nodes. Referring to the motif-based PageRank model proposed in [52], when we do not consider the self-connection, motif-based adjacency matrix can be calculated as Eq. (5), which can be represented as $H_{gl} H_{gl}^T$.

$$H_{gl} H_{gl}^T = (CC) \odot C, \quad (5)$$

where H_{gl} denotes the motif incidence matrix of the group-level hypergraph. $C \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{G}|}$ is the symmetric adjacency matrix of the projected graph, where $c_{ij} = 1$ if there is a connection between group i and group j , otherwise $c_{ij} = 0$. CC denotes the paths connecting three vertices, and $\odot C$ transforms the paths into closed triangles.

Similar to the hypergraph convolution proposed in the user-level hypergraph, the transformed hypergraph convolutions in the group-level hypergraph can be defined as Eq. (6). Following the LightGCN and MHCN [18, 49], since the self-connection has little effect on performance, the Eq. (6) is equivalent to Eq. (2).

$$Z^{(l+1)} = D_{gl}^{-1} H_{gl} H_{gl}^T Z^{(l)} \Psi^{(l)}, \quad (6)$$

where $D_{gl} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{G}|}$ is the degree matrix of the hypergraph motif $H_{gl} H_{gl}^T H_{gl} H_{gl}^T$ can be regarded as the motif-induced adjacency matrix. $\Psi^{(l)}$ is the parameter matrix in the l -th layer. $Z^{(l)}$ is the group representation in the l -th layer of the hypergraph convolutional network, where $Z^{(0)} = \tilde{Z}$. The \tilde{Z} learns from the attention-based group preference aggregator.

3.1.3 Loss function. Here, we adopt the pairwise learning task loss function to optimize user and item representations in the user-level hypergraph, which is designed as follows:

$$\begin{aligned} \mathcal{L}_{UI} = & - \sum_{(u,i,j) \in O} (\hat{r}_{ui} - \hat{r}_{uj} - 1)^2, \\ \hat{r}_{ui} = & p_u \tilde{q}_i^T, \end{aligned} \quad (7)$$

where \tilde{p}_u represents the embedding vector for user u . \tilde{q}_i represents the embedding vector for item i . O represents the training set, in which each one includes the user u , interacted item i , and unobserved items j . Our target is to make the margin between the positive samples (u, i) and negative samples (u, j) as close to 1 as possible.

For the group-level hypergraph, its loss function can be achieved as follows:

$$\begin{aligned} \mathcal{L}_{GI} = & - \sum_{(g,i,j) \in O'} (\hat{s}_{gi} - \hat{s}_{gj} - 1)^2, \\ \hat{s}_{gi} = & z_g \tilde{q}_i^T, \end{aligned} \quad (8)$$

where z_g is the group representation learned from the attention-based preference aggregation strategy. Similar to the user-level loss, O' represents the training set, which includes the group g , interacted item i , and unobserved items j . To enhance the learning process of these two recommendation methods, we apply a joint training strategy, whose loss function consists of two parts: user preference loss and group preference loss. It is given by

$$\mathcal{L} = \mathcal{L}_{UI} + \mathcal{L}_{GI}. \quad (9)$$

3.2 Boosting Group Recommendation with Double-Scale Self-Supervision

Despite the remarkable capability of hypergraph modeling, as groups may occasionally form, only minimal group-item interactions can be observed (i.e., the data sparsity issue), which may lead to sub-optimal recommendation performance [14, 20, 44]. Several models utilize the user member preferences to synthesize group preferences and alleviate the sparsity issue. However, they overlook the fact that the user-item interaction is also sparse. To address this problem and refine the representations of users and groups, we propose a double-scale node dropout strategy on the hierarchical hypergraph (including coarse and fine granularity) to augment the raw data and create two types of self-supervision signals for

contrastive learning to boost the performance of HHGR. A self-supervised variant of HHGR, S^2 -HHGR, is presented as follows. It should be noted that, for S^2 -HHGR, we upgrade the design in Section 3.1.1 and employ two new hypergraph convolutional networks to learn user representations, which can be seamlessly coupled with the double-scale self-supervised strategy.

Coarse-grained node dropping strategy. We drop a certain portion of users at a coarse-grained granularity, where the magnitude of missing nodes is a hyperparameter. Unlike node dropping in a simple graph accompanied by deleting connected edges, the hyperedge may not be deleted when discarding some nodes in the hypergraph. Besides, when a deleted node belongs to multiple group hyperedges, the node would be deleted in all certain hyperedges. We define the coarse-grained node dropping function f_{coarse} , as follows:

$$\mathbf{h}_c = f_{coarse}(\mathbf{a}_c, \mathbf{h}_{ul}) = \mathbf{a}_c \odot \mathbf{h}_{ul}, \quad (10)$$

where \mathbf{h}_c represents the column vector of the coarse-grained hypergraph incidence matrix \mathbf{H}_c , and $\mathbf{a}_c \in \{0, 1\}^{|\mathcal{U}|}$ is a mask vector with its entries being 0 at a given probability, controlling the dropout magnitude of the nodes in \mathbf{h}_{ul} . \odot represents the element-wise product, which means the mask vector would multiply with each column in the user-level hypergraph incidence matrix \mathbf{h}_{ul} . After the coarse-grained node dropping, we can obtain a perturbed user-level hypergraph. We encode it through a new hypergraph convolutional network $g_c(\cdot)$ to get the coarse-grained user representation \mathbf{P}' , which can be as one data augmentation of the raw user representations.

$$\mathbf{P}'^{(l+1)} = g_c(\mathbf{P}'^{(l)}) = \mathbf{D}_c^{-1} \mathbf{H}_c \mathbf{B}_c^{-1} \mathbf{H}_c^T \mathbf{P}'^{(l)} \Gamma^{(l)}, \quad (11)$$

where \mathbf{D}_c and \mathbf{B}_c are the vertex degree and hyperedge degree diagonal matrices of the coarse-grained hypergraph. Γ denotes the parameter matrix in the coarse-grained hypergraph convolutional network. $\mathbf{P}'^{(l)}$ is the user representation in the l -th layer of the coarse-grained hypergraph neural network, where $\mathbf{P}'^{(0)} = \tilde{\mathbf{P}}$. $\tilde{\mathbf{P}}$ is the randomly initialized d-dimensional representation matrix.

Fine-grained node dropping strategy. We perturb the hypergraph construction by dropping a certain number of users in a particular group hyperedge. Analogously, we define a fine-grained node dropping function f_{fine} , which is represented as follows:

$$\mathbf{h}_f = f_{fine}(\mathbf{a}_f, \mathbf{h}_{ul}) = \mathbf{a}_f \odot \mathbf{h}_{ul}, \quad (12)$$

where \mathbf{h}_f represents the column vector of the fine-grained hypergraph incidence matrix \mathbf{H}_f , and $\mathbf{a}_f \in \{0, 1\}^{|\mathcal{U}|}$ is the fine-grained mask vector, which will drop nodes with a fixed probability in only one hyperedge. Different from the coarse-grained strategy, each time \mathbf{a}_f is multiplied by \mathbf{h}_f , it is reassigned. Next, we encode $\mathbf{P}''^{(l)}$ through $g_f(\cdot)$ to obtain a new user representation:

$$\mathbf{P}''^{(l+1)} = g_f(\mathbf{P}''^{(l)}) = \mathbf{D}_f^{-1} \mathbf{H}_f \mathbf{B}_f^{-1} \mathbf{H}_f^T \mathbf{P}''^{(l)} \Phi^{(l)}, \quad (13)$$

where \mathbf{D}_f and \mathbf{B}_f are the vertex and hyperedge degree diagonal matrices of the fine-grained hypergraph convolutional network. \mathbf{H}_f is the incidence matrix. Φ represents the parameter matrix. $\mathbf{P}''^{(l)}$ is the user representation in the l -th layer of the fine-grained hypergraph neural network, where $\mathbf{P}''^{(0)} = \tilde{\mathbf{P}}$. To avoid the loss of some essential information, we add two representations at the two granulates to get the final user representation: $\mathbf{P} = \mathbf{P}' + \mathbf{P}''$.

Contrastive learning. Having established different granularity augmented views of nodes, we treat the granularity of the same node as the positive pairs and any different granularity of nodes as the negative pairs. We hope that the distribution of the user representation vectors from two pretext tasks can be as close as possible. Hence, we design a discriminator function $f_{\mathcal{D}}(\cdot)$ to learn a score between two input vectors and assign higher scores to positive pairs compared with negative samples. We adopt the cross-entropy loss as contrastive loss function to enforce maximizing the agreement between positive pairs, which is defined as follows:

$$\mathcal{L}_{UU} = - \sum_{i \in U} \left[\log \sigma(f_{\mathcal{D}}(\mathbf{p}'_i, \mathbf{p}''_i)) + \sum_{j=1}^n \left[\log \sigma(1 - f_{\mathcal{D}}(\mathbf{p}'_j, \mathbf{p}''_i)) \right] \right], \quad (14)$$

$$f_{\mathcal{D}}(\mathbf{p}'_i, \mathbf{p}''_i) = \sigma(\mathbf{p}'_i \mathbf{W}_{\mathcal{D}} \mathbf{p}''_i^T), \quad (15)$$

where \mathbf{p}'_j is the representation of user j in the coarse-grained hypergraph; n is the number of negative samples randomly selected from the same batch.

3.3 Model Optimization

To optimize the S^2 -HHGR's parameters, we unify the loss functions of the task of group recommendation and self-supervised learning. We first update the loss function of self-supervised learning, \mathcal{L}_{UU} , as the pre-training strategy to optimize the user representations. Then we optimize the supervised loss function of \mathcal{L}_{UI} and \mathcal{L}_{GI} to obtain the representations of users and items. Meanwhile, the model will be fine-tuned and updated during the process of self-supervised learning. The learned user embedding would be used to generate group representations, which are updated in the objective function \mathcal{L}_{GI} . Specifically, we train our model using the Adam algorithm. The overall objective is defined as follows:

$$\mathcal{L} = \beta \mathcal{L}_{UU} + \mathcal{L}_{UI} + \mathcal{L}_{GI}, \quad (16)$$

where β is the hyper-parameter to balance the task of self-supervised learning and supervised learning task for group recommendation.

3.4 Complexity Analysis

In this section, we discuss the complexity of S^2 -HHGR from model size and time complexity.

Model size. The parameters of our model consist of three parts: 1) hierarchical hypergraph convolutional network, 2) attention-based group preference aggregator, and 3) discriminator network. The coarse-grained and fine-grained user-level hypergraph convolutional network have parameters of size $2[l \times d \times d]$. The parameters size of the group-level hypergraph convolutional network is $l \times d \times d$. The parameter size of the trainable weighted matrix in the preference aggregator is $3 \times d \times d$. The discriminator measures the similarity between user representations of different granularities, the weight matrix in $f_{\mathcal{D}}(\cdot)$ has a trainable weighted matrix $\mathbf{W}_{\mathcal{D}}$ with the shape $d \times d$.

Time complexity. The computational cost mainly comes from two parts: the hierarchical hypergraph convolution and the double-scale self-supervised learning strategy. The time complexity of hierarchical hypergraph is mainly from the information propagation consumption, through l , is less than $O(l \times |\mathcal{H}| \times d)$, where $|\mathcal{H}|$

denotes the number of nonzero elements in the incidence matrix H . Since our model contains a user-level hypergraph convolutional network and a group-level hypergraph convolutional network, the time complexity of the hierarchical hypergraph convolutional neural network is about $O(l^2 \times |H|^2 \times d^2)$. As for the self-supervised learning strategy, the cost derives from the hypergraph generation of different perspectives and contrastive learning. The time cost of hypergraph generation is less than $O(2 \times M \times M)$. The time cost of contrastive learning is less than $O(|U| \times n)$, where n is the number of negative samples.

Table 1: The statistics of datasets.

Dataset	#User	#Item	#Group	#U-I Feedback	#G-I Feedback
Douban	2,964	39,694	2,630	823,851	463,040
Weeplaces	8,643	25,081	22,733	1,358,458	180,229
GAMRa2011	602	7,710	290	116,344	145,068

4 EXPERIENCE AND RESULTS

In this section, we conduct extensive experiments to justify our model’s superiority and reveal the reasons for its effectiveness. Specifically, we will answer the following research questions to unfold the experiments.

RQ1: Compared with the state-of-the-art group recommendation models, how does our model perform?

RQ2: What are the benefits of each component (i.e., the hierarchical hypergraph and the self-supervised learning) in our model?

RQ3: How do the hyper-parameters influence the effectiveness of the S^2 -HHGR?

4.1 Experimental Settings

4.1.1 Datasets. We conduct experiments on three public datasets: Weeplaces [32], CAMRa2011 [28], and Douban [44]. Weeplaces dataset includes the users’ check-in history in a location-based social network. We follow GroupIM [32] to construct group interactions by using the user check-in records and their social network. As for the CAMRa2011, it is a movie rating dataset containing individual users and households records. We follow the idea of AGREE [6] and convert the explicit rating to implicit preference, where the rating records are regarded as 1. Douban dataset is from the Douban platform, which consists of a variety of group social activities. As the Douban dataset does not contain explicit group information, we follow the idea of SIGR [44] to extract implicit group data. Specifically, we regard users’ friends who have participated in the same activity as the group members. The statistical information of the three datasets is shown in Table 1. We randomly split the set of all groups into training (70%), validation (10%), and test sets (20%).

4.1.2 Baselines. To answer the **RQ1**, we compare the proposed model with the following models:

- **Popularity.** This method ranks items according to their popularity.
- **NeuMF** [19]. NeuMF is a neural network-based collaborative filtering model to benchmark the recommendation performance.

We treat all groups as virtual users and utilize group-item interactions to generate the group preferences.

- **AGREE** [6]. This method utilizes attentional preference aggregation to compute group member weights and adopts neural collaborative filtering to learn the group-item interaction.
- **MoSAN** [35]. MoSAN is a neural group recommender that employs a collection of sub-attentional networks to learn each user’s preference and model member interactions.
- **SIGR** [44]. This is a state-of-the-art group recommendation model, which introduces a latent variable and the attention mechanism to learn users’ local and global social influence. It also utilizes the bipartite graph embedding model to alleviate the data sparsity problem.
- **GroupIM** [32]. This model aggregates the users’ preferences as the group preferences via the attention mechanism. It maximizes the mutual information between the user representations and its belonged group representations to alleviate the data sparsity problem.

HHGR is the vanilla version of our proposed model, and S^2 – HHGR represents the self-supervised version.

4.1.3 Evaluation metrics. To measure the performance of all methods, we employ the widely adopted metrics $NDCG@K$ and $Recall@K$ with $k = \{20, 50\}$. $NDCG@K$ evaluates the ranking of true items in the recommendation list. $Recall@K$ is the fraction of relevant items that have been retrieved in the Top-K relevant items.

4.1.4 Parameter settings. For the general settings, the embedding size is 64, the batch size for the mini-batch is 512, and the number of negative samples is 10. During the training process of the double-scale self-supervised learning, the initial learning rate is $5e-4$. For the group-level hypergraph training, the learning rate is $1e-4$. The user-level and group-level hypergraph neural network structure is two-layer and one-layer, respectively. For the baseline models, we refer to their best parameter setups reported in the original papers and directly report their results if we use the same datasets and evaluation settings.

4.2 Recommendation Performance Comparison (RQ1)

4.2.1 Overall performance comparison. In this part, we validate the superiority of HHGR and S^2 -HHGR on three datasets. Table 2 shows the experimental results of the proposed models’ performance compared with the baselines. We highlight the best results of all models in boldface. According to the results, we note the following key observations:

- 1) Among these methods, the attention-based group models outperform baseline recommenders (i.e., Popular and NeuMF) on most datasets due to their ability to dynamically model the user interactions within groups and learn various weights of different members in the group.
- 2) In the models based on attention mechanism, HHGR is better than most models (including AGREE, MoSAN, and SIGR). We believe that the performance improvement of HHGR verifies the effectiveness of hypergraph and hypergraph neural network modules to exploit high-order user interactions.
- 3) SIGR performs better than AGREE and MoSAN due to considering the form of bipartite graph to represent user-item interactions, especially on

Table 2: The general recommendation performance comparison on three datasets.

Dataset	Weeplaces					CAMRa2011					Douban		
	Metric	N@20	N@50	R@20	R@50	N@20	N@50	R@20	R@50	N@20	N@50	R@20	R@50
Baseline recommender													
Popular	0.063	0.074	0.126	0.176	0.099	0.122	0.172	0.226	0.003	0.005	0.009	0.018	
NeuMF	0.193	0.244	0.271	0.295	0.305	0.367	0.393	0.450	0.144	0.187	0.268	0.366	
Attention-based group recommender													
AGREE	0.224	0.267	0.354	0.671	0.307	0.418	0.529	0.688	0.226	0.264	0.321	0.536	
MoSAN	0.287	0.334	0.548	0.738	0.423	0.466	0.572	0.801	0.193	0.239	0.424	0.639	
SIGR	0.357	0.391	0.524	0.756	0.499	0.524	0.585	0.825	0.249	0.255	0.481	0.673	
GroupIM	0.431	0.456	0.575	0.773	0.637	0.659	0.753	0.874	0.257	0.284	0.523	0.696	
HHGR	0.379	0.398	0.554	0.764	0.517	0.532	0.703	0.830	0.254	0.267	0.507	0.677	
S^2-HHGR	0.456	0.478	0.592	0.797	0.645	0.671	0.779	0.883	0.279	0.294	0.561	0.741	

the CAMRa2011 dataset. Besides, MoSAN also achieves better results because of its expressive power of preference aggregators to capture different personal weights in group-item interactions. 4) On the other hand, although HHGR is slightly inferior to GroupIM, the enhanced model S^2 -HHGR beats the most advanced group recommendation models on three datasets, which verifies the effectiveness of self-supervised learning strategies.

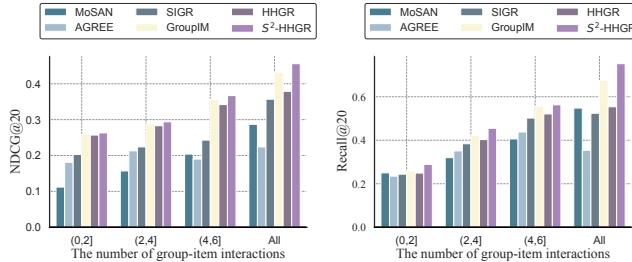


Figure 3: Performance comparison of attention-based group recommendation models on sparsity datasets.

4.2.2 Performance on sparsity datasets. We adopt the self-supervised learning strategy to alleviate the data sparsity problem. To validate the effectiveness of the proposed double-scale node dropout strategy, we study experiments on sparse datasets. We split the dataset into four groups based on the number of group-item interactions. We find that the performance of all models varies with the number of group interactions. Due to the limited space, we only show the performance of attention-based models on the Weeplaces dataset. The results are shown in Figure 3.

As shown in Figure 3, NDCG and Recall generally increase as the number of interacting groups of an item increases, since more interactions will provide more information for learning the group preference representation. Without the auxiliary information, models that adopt self-supervised learning (GroupIM and S^2 -HHGR) are competitive to the models without self-supervised learning. Besides, we can observe that S^2 -HHGR shows the best performance compared with other models. We argue that the effectiveness of S^2 -HHGR can be attributed to the double-scale node dropout strategy in hypergraph neural networks, which can enhance the representations of group preferences.

4.3 Ablation Study (RQ2)

4.3.1 Investigation of the hierarchical hypergraph. To investigate the effectiveness of the hierarchical hypergraph, we conduct experiments on Weeplaces and CAMRa2011 for two HHGR variants, each of which has one of the level removed. **HHGR-wg** and **HHGR-wu** to denote the ablated model without group level or user level. HHGR-wg only considers the user-level hypergraph, which means the group preference representations generated from the preference aggregator are the final group preference. HHGR-wu adopts the group-item interaction information to initialize the group preference and only considers the group level hypergraph to update the group preference. The experimental results are demonstrated in Table 3. From the results, we can observe that the performance of HHGR-wu falls to the performance of HHGR-wg and HHGR on two datasets. Compared with the Weeplace dataset, the performance of the HHGR and its variants on the CAMRa2011 is better. The possible reason might be that the CAMRa2011 dataset is denser, which is beneficial to selecting informative inter- and intra-group user interactions.

Table 3: Comparison between HHGR and its variants.

Method	Weeplaces		CAMRa2011		
	Metric	N@50	R@50	N@50	R@50
HHGR-wu		0.288	0.683	0.495	0.797
HHGR-wg		0.378	0.751	0.511	0.815
HHGR		0.398	0.764	0.532	0.830

4.3.2 Investigation of self-supervised learning. Different types of self-supervised learning. To investigate the feasibility and efficiency of self-supervised learning, we conduct an ablation study to investigate each component's contributions. We propose two variants of S^2 -HHGR: **HHGR-F** and **HHGR-C**. **HHGR-F** means that only fine-grained node dropping is considered. **HHGR-C** means only coarse-grained node dropping is used. For the above two variant models, we maximize the mutual information between the initial node representations and the dropping ones. Considering the limited space of the paper, we only compare these models with S^2 -HHGR on Weeplaces and CAMRa2011.

Figure 4 shows that S^2 -HHGR achieves the best performance over independent granularity models, where self-supervised learning strategies contribute to the recommendation performance. When we only use fine-grained node dropping, the HHGR-F performs slightly worse than S^2 -HHGR but better than HHGR-C, which implies the fine-grained node dropping contributes more to the node representation learning. We argue that the fine-grained task would generate more hard samples, strengthening model training. When we only use the coarse-grained node dropout strategy, the group recommendation performance may be slightly better than HHGR. Besides, without the self-supervised learning strategy would result in a performance decline in most cases on two datasets. The performance of HHGR is worse than HHGR-F and HHGR-C in most cases, indicating that the self-supervised strategy is effective. Overall, the experimental results show that self-supervised learning is useful.

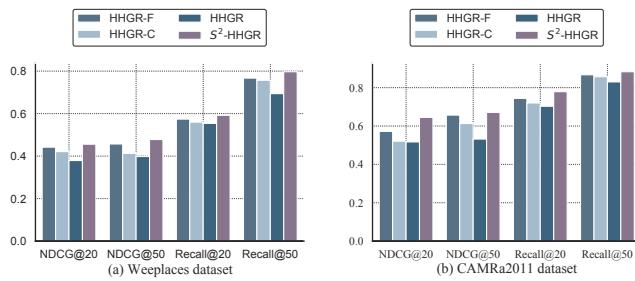


Figure 4: The influence of different self-supervised learning strategies.

Different extent of self-supervised learning. We note that the use of different granularities of node dropping benefits the node representation. We further analyze the extent of coarse and fine granularity node dropout. Specifically, we set different dropping rates from 0.1 to 0.9 in two granularity tasks. When we change the rate of one granularity task, we will fix another granularity's rate. As presented in Figure 5, with the increase of dropping rate, there is a significant increase in performance. When the rate reaches the peak, approximately 0.3 and 0.2 on fine and coarse granularity dropping, respectively, it decreases. It suggests that node dropping can improve the HHGR performance with increasing dropping strength, but too many deletions will worsen the data sparsity problem.

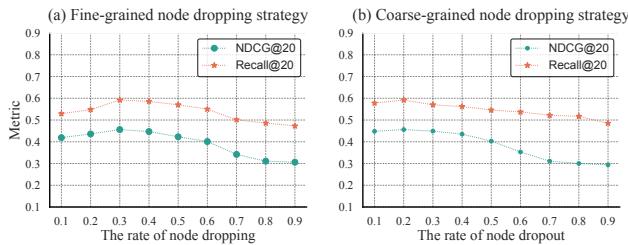


Figure 5: The performance of different extent of two pretext tasks.

4.4 Parameter Sensitivity Analysis (RQ3)

This part focuses on pointing out which hyper-parameters affect our model. The analyzed hyper-parameters include the learning rate, the depth of hypergraph convolutional layers, the number of batch sizes, and the number of negative samples. Due to limited space, we only show the experimental results on the Weeplaces dataset. Specifically, we search the proper value in a small interval and set the learning rate as $\{1e-5, 5e-5, 1e-4, 5e-4, 1e-3\}$, and report the performance of S^2 -HHGR with five values $\{1, 2, 3, 4, 5\}$ of the hypergraph convolutional layer. Besides, the batch size changes from 16 to 512 at intervals of increasing powers of 2, and the number of negative samples is set from 5 to 30 at intervals of 5.

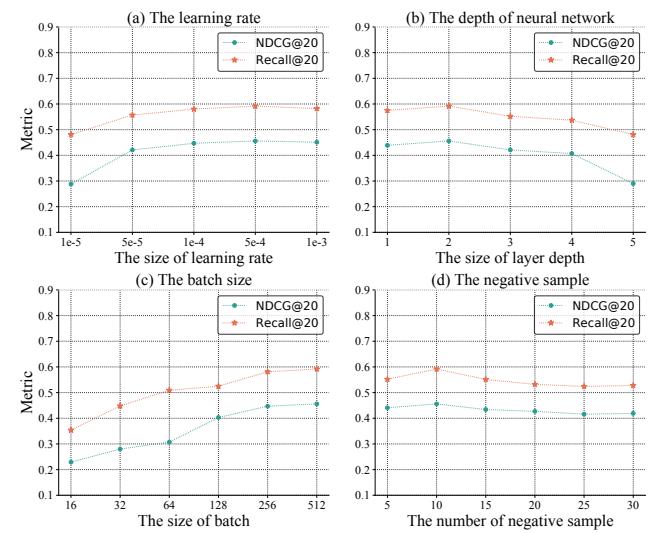


Figure 6: The influence of the model parameters.

As shown in Figure 6 (a), we observe $5e-4$ is sufficient for the learning rate. With the increase of the rate, the performance shows a trend of rising first and then falling, reaching the peak at $5e-4$. According to Figure 6 (b), when the depth of hypergraph convolutional layers is 2, the model reaches the performance peak. As the depth of convolutional layers increases, the performance of S^2 -HHGR steadily declines. A possible reason is that a multi-layer hypergraph convolutional network would lead to the over-smoothing problem. HHGR model also encounters this problem, but it tends to over-smoothing at lower layers. We believe that a self-supervised learning strategy can alleviate the data sparsity problem because it would delete some redundant information to enhance the model performance. Besides, the parameter of batch size is also important. Figure 6 (c) depicts the change of model performance with respect to different batch sizes from 16 to 512. The optimal size is 512. Therefore, we set the batch size to 512 when we train the model. As shown in Figure 6 (d), when the number of negative samples is 10, our model achieves the best performance in the Weeplaces dataset. We can conclude that a small number of negative samples can promote the recommendation task, while a bigger one would mislead it.

5 RELATED WORK

5.1 Group Recommendation

Early work on group recommendation is usually based on the preference integration of group members or the score aggregation of an item across users. The three most common aggregations include the average [5], the least misery [1], and the maximum satisfaction strategy [3, 4]. The average method takes the mean score of all members in the group. The least misery is committed to filtering all members' smallest score to please everyone. The maximum satisfaction strategy aims to reach the maximum satisfaction of the member in the group. These three aggregating methods have a significant drawback: they are oversimplified and ignore user interactions in groups.

Recently, with the successful development of deep learning, models based on the attention mechanism are becoming more and more popular for modeling intra-group user interactions and learning different users' influences [6, 35]. For example, GAME [20] utilizes the heterogeneous information network and attention mechanism to learn the nodes' multi-view embeddings and members' weights. Yin *et al.* [44] propose to take the attention mechanism to exploit each user's intra-group user interactions and adopt the bipartite graph embedding to mitigate the data sparsity problem. Although the above methods based on the attention mechanism have made remarkable achievements in group recommendation, they hardly consider the high-order user interactions.

5.2 Hypergraph-Based Recommendation

Although the graph neural network approaches have achieved successful results on capturing high-order relations in various tasks [7, 25, 56], these approaches are only appropriate in pairwise connections, which has limitation in expressing complex structures of data. Hypergraph has shown promising potential in modeling complex high-order relations [36, 46, 51, 53]. Recently, some studies try to combine the hypergraph with the recommender systems to improve their performances. For example, DHCF constructs two hypergraphs for users and items to model their high-order correlations and enhance recommendation models based on collaborative filtering [22]. Xia *et al.* [39] model session-based data as a hypergraph and use the hypergraph neural network to enhance session-based recommendation. Wang *et al.* [37] incorporate the hypergraph into next-item recommendation systems to represent the short-term item correlations. Yu *et al.* [49] propose a multi-channel hypergraph convolutional network and design multiple motif-induced hypergraphs to exploit high-order user relations patterns.

Despite the tremendous efforts devoted to these models, they just adopt the paradigm of hypergraph representation learning to model high-order relations. Our models differ significantly from previous approaches in that we use the motif to select informative group interactions.

5.3 Self-Supervised Learning in RS

Self-supervised learning offers a new angle to generate additional supervised signals via transferring the unlabeled data [2, 9, 15, 24, 48]. Generally, there are two types of self-supervised learning approaches: generative models and contrastive models. Generative models learn to reconstruct the input data and make it similar to

raw data [12, 27]. Contrastive models learn discriminative representations by contrasting positive and negative samples [8, 13, 17, 30]. Recently, some studies attempt to explore the self-supervised learning framework for the recommendation [41, 42, 54]. For example, Zhou *et al.* [54] propose the S^3 -Rec model and design some self-supervised training strategies to pre-train the model. Bert4Rec [33] adopts the Cloze objective to the sequential recommendation by predicting the random masked items in the sequence. Xin *et al.* [41] adopt the cross-entropy as self-supervised Q-learning to fine-tune the next-item recommendation model. Xie *et al.* [40] propose three data augmentation operations: item crop, item mask, and item reorder, as self-supervision signals to enhance sequential recommendation.

The closest work to ours is GroupIM [32], which maximizes the mutual information between users and groups to optimize the representation of both. Unlike the above approaches, our work is the first to consider the double-scale node dropout strategies in hypergraph as the supervised signals in group recommendation. We maximize the mutual information between the same hypergraph's different views to enhance data representations and improve recommendation performance.

6 CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel neural group recommendation with the hierarchical hypergraph convolutional network and self-supervised learning strategy to capture inter- and intragroup user interactions and alleviate the data sparsity problem. With the triadic motifs, the proposed model can obtain more reliable user interactions beyond groups for more accurate group representations. We innovatively designed a double-scale node dropout strategy as the supervision signals against the data sparsity problem. We conducted extensive experiments on three public datasets to evaluate our models' performance. The experimental results verified the superiority of HHGR and its enhanced version based on self-supervised learning (S^2 -HHGR) compared with other state-of-the-art models. In the future work, we tend to deepen the application of self-supervised learning in group recommendation models and design more general auxiliary tasks for the recommendation to improve the recommendation performance.

ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China (6217022345), Natural Science Foundation of Chongqing, China (cstc2020jcyj-msxmX0690), Fundamental Research Funds for the Central Universities of Chongqing University (2020CDJ-LHZZ-039), ARC Discovery Project (DP190101985), and ARC Future Fellowship (FT210100624).

REFERENCES

- [1] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawla, Gautam Das, and Cong Yu. 2009. Group Recommendation: Semantics and Efficiency. *Proc. VLDB Endow.* 2, 1 (2009), 754–765.
- [2] Alexei Baevski and Abdelrahman Mohamed. 2020. Effectiveness of Self-Supervised Pre-Training for ASR. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4–8*. IEEE, 7694–7698.
- [3] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26–30*. ACM, 119–126.

- [4] Shlomo Berkovsky and Jill Freyne. 2010. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30*. ACM, 111–118.
- [5] 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*. Vol. 324. 1–20.
- [6] 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, SIGIR 2018, Ann Arbor, MI, USA, July 08-12*. ACM, 645–654.
- [7] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7*. Association for Computational Linguistics, 4171–4186.
- [9] Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. [n.d.]. CogLTX: Applying BERT to Long Texts. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS 2020, December 6-12, 2020, virtual*.
- [10] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph Neural Networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1*. AAAI Press, 3558–3565.
- [11] Adrien Friggeri, Guillaume Chelius, and Eric Fleury. 2011. Triangles to Capture Social Cohesion. In *2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*. IEEE Computer Society, 258–265.
- [12] Min Gao, Junwei Zhang, Junliang Yu, Jundong Li, Junhao Wen, and Qingyu Xiong. 2021. Recommender systems based on generative adversarial networks: A problem-driven perspective. *Inf. Sci.* 546 (2021), 1166–1185.
- [13] Priya Goyal, Dhruv Mahajan, Abhimanyu Gupta, and Ishan Misra. 2019. Scaling and Benchmarking Self-Supervised Visual Representation Learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2*. IEEE, 6390–6399.
- [14] Lei Guo, Hongzhi Yin, Tong Chen, Xiangliang Zhang, and Kai Zheng. 2021. Hierarchical Hyperedge Embedding-based Representation Learning for Group Recommendation. *CoRR* abs/2103.13506 (2021).
- [15] Hakim Hafidi, Mounir Ghogho, Philippe Ciblat, and Ananthram Swami. 2020. GraphCL: Contrastive Self-Supervised Learning of Graph Representations. *CoRR* abs/2007.08025 (2020).
- [16] Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4116–4126.
- [17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19*. IEEE, 9726–9735.
- [18] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 639–648.
- [19] 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, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 173–182.
- [20] 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 649–658.
- [21] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Wei Cao. 2014. Deep Modeling of Group Preferences for Group-Based Recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*. AAAI Press, 1861–1867.
- [22] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual Channel Hypergraph Collaborative Filtering. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2020, Virtual Event, CA, USA, August 23-27*. ACM, 2020–2029.
- [23] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. 2019. Dynamic Hypergraph Neural Networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16*. ijcai.org, 2635–2641.
- [24] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. 2020. Self-supervised Learning on Graphs: Deep Insights and New Direction. *CoRR* abs/2006.10141 (2020). arXiv:2006.10141 https://arxiv.org/abs/2006.10141
- [25] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- [26] Geon Lee, Jihoon Ko, and Kijung Shin. 2020. Hypergraph Motifs: Concepts, Algorithms, and Discoveries. *Proc. VLDB Endow.* 13, 11 (2020), 2256–2269.
- [27] Xia Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised Learning: Generative or Contrastive. *CoRR* abs/2006.08218 (2020).
- [28] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting Geographical Neighborhood Characteristics for Location Recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7*. ACM, 739–748.
- [29] Ritu Meena and Sonajharia Minz. 2020. Group Recommender Systems - An Evolutionary Approach Based on Multi-expert System for Consensus. *J. Intell. Syst.* 29, 1 (2020), 1092–1108.
- [30] Ishan Misra and Laurens van der Maaten. 2020. Self-Supervised Learning of Pretext-Invariant Representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19*. IEEE, 6706–6716.
- [31] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2020, CA, USA, August 23-27*. ACM, 1150–1160.
- [32] 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, SIGIR 2020, China, July 25-30*. ACM, 1279–1288.
- [33] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 1441–1450.
- [34] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 214–221.
- [35] 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, SIGIR 2019, Paris, France, July 21-25*. ACM, 255–264.
- [36] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. 2018. Structural Deep Embedding for Hyper-Networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, Louisiana, USA, February, 2-7*. AAAI Press, 426–433.
- [37] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. [n.d.]. Next-item Recommendation with Sequential Hypergraphs. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 1101–1110.
- [38] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.
- [39] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 4503–4511.
- [40] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2010.14395* (2020).
- [41] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. 2020. Self-Supervised Reinforcement Learning for Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 931–940.

- [42] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2020. Self-supervised Learning for Large-scale Item Recommendations. *arXiv preprint arXiv:2007.12865* (2020).
- [43] Hongzhi Yin and Bin Cui. 2016. *Spatio-Temporal Recommendation in Social Media*. Springer.
- [44] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social Influence-Based Group Representation Learning for Group Recommendation. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11*. IEEE, 566–577.
- [45] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, and Xiaofang Zhou. 2020. Overcoming Data Sparsity in Group Recommendation. *CoRR* abs/2010.00813 (2020).
- [46] Se-eun Yoon, HyungSeok Song, Kijung Shin, and Yung Yi. 2020. How Much and When Do We Need Higher-order Information in Hypergraphs? A Case Study on Hyperedge Prediction. *CoRR* abs/2001.11181 (2020). arXiv:2001.11181 <https://arxiv.org/abs/2001.11181>
- [47] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12*.
- [48] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 2084–2092.
- [49] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *WWW '21: The Web Conference 2021, Virtual Event /Ljubljana, Slovenia, April 19-23, 2021*. ACM /IW3C2, 413–424.
- [50] Quan Yuan, Gao Cong, and Chin-Yew Lin. 2014. COM: a generative model for group recommendation. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA - August 24-27, 2014*. ACM, 163–172.
- [51] Ruochi Zhang, Yuesong Zou, and Jian Ma. 2020. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30*. OpenReview.net.
- [52] Huan Zhao, Xiaogang Xu, Yangqiu Song, Dik Lun Lee, Zhao Chen, and Han Gao. 2018. Ranking Users in Social Networks With Higher-Order Structures. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 232–240.
- [53] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with Hypergraphs: Clustering, Classification, and Embedding. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7*. MIT Press, 1601–1608.
- [54] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. [n.d.]. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 1893–1902.
- [55] Le-kun Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueteng Zhuang. 2018. Dynamic Network Embedding by Modeling Triadic Closure Process. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 571–578.
- [56] Difan Zou, Ziniu Hu, Yewen Wang, Song Jiang, Yizhou Sun, and Quanquan Gu. 2019. Layer-Dependent Importance Sampling for Training Deep and Large Graph Convolutional Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 11247–11256.