Group Identification via Transitional Hypergraph Convolution with Cross-view Self-supervised Learning

Mingdai Yang
University of Illinois at Chicago
Chicago, USA
myang72@uic.edu

Zhiwei Liu
Salesforce AI Research
Palo Alto, USA
zhiweiliu@salesforce.com

Liangwei Yang
lyang84@uic.edu
University of Illinois at Chicago
Chicago, USA

Xiaolong Liu
Chen Wang
xliu262@uic.edu
cwang266@uic.edu
University of Illinois at Chicago
Chicago, USA

Hao Peng*
School of Cyber Science and
Technology, Beihang University,
Beijing, China
Yunnan Key Laboratory of Artificial
Intelligence, Kunming University of
Science and Technology,
Kunming, China
penghao@buaa.edu.cn

Philip S. Yu
University of Illinois at Chicago
Chicago, USA
psyu@uic.edu

ABSTRACT

With the proliferation of social media, a growing number of users search for and join group activities in their daily life. This develops a need for the study on the group identification (GI) task, i.e., recommending groups to users. The major challenge in this task is how to predict users' preferences for groups based on not only previous group participation of users but also users' interests in items. Although recent developments in Graph Neural Networks (GNNs) accomplish embedding multiple types of objects in graph-based recommender systems, they, however, fail to address this GI problem comprehensively. In this paper, we propose a novel framework named **G**roup Identification via **T**ransitional Hypergraph Convolution with **G**raph **S**elf-supervised Learning (GTGS). We devise a novel transitional hypergraph convolution layer to leverage users' preferences for items as prior knowledge when seeking their group preferences. To construct comprehensive user/group representations for GI task, we design the cross-view self-supervised learning to encourage the intrinsic consistency between item and group preferences for each user, and the group-based regularization to enhance the distinction among group embeddings. Experimental results on three benchmark datasets verify the superiority of GTGS. Additional detailed investigations are conducted to demonstrate the effectiveness of the proposed framework.

CCS CONCEPTS

• **Information systems → Collaborative filtering**.

*Corresponding author

KEYWORDS

Recommender System; Graph Neural Network; Hypergraph Learning; Group Recommendation

1 INTRODUCTION

With the development of online shopping, information collection and decision-making have become essential yet overwhelming for individual customers. To seek suggestions for reference before purchasing, a growing number of users decide to join groups on online platforms and communities. For example, an expectant mother would seek suggestions from other experienced mothers in online communities. Online groups provide a space for users to discuss their demands and share their experiences. Moreover, online groups can affect and develop users' interests. For example, a group of movie fans would discuss their anticipated movies, and a group of video game players would debate whether a newly released game is worth buying. In terms of platforms, users' participation and retention rate can be boosted by the attachment to online groups [52]. Therefore, it improves the long-term stickiness of users to platforms if personalized social groups can be suggested for users. In this paper, we define this problem as *group identification*[1] (GI) task, which aims at recommending groups to users.

If regarding users and groups as nodes and their interactions as edges, we may thus resolve the group identification (GI) task as predicting edges over the user-group bipartite graph [23, 48]. Then, existing graph-based recommender systems methods [4, 11, 35, 40] would be adopted, which predict edges based on interactions in the

---
[1]We use *group identification* rather than *group recommendation* is because the later usually refers to recommending items to a group of users.
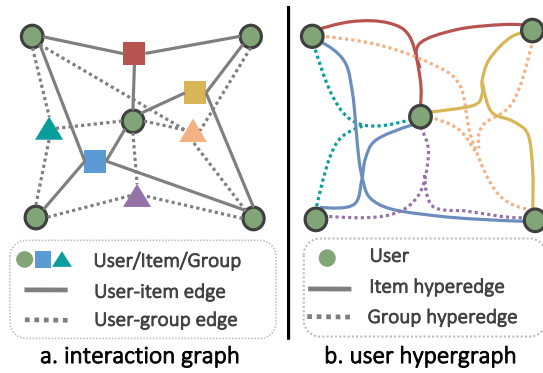
Figure 1: Toy example of how to construct a GI hypergraph from user-group-item interaction graph.

graph. However, GI task is far more complex than this because the group participation of users also depends on the item interests of users. For instance, a user may join a group because other members purchased the item of her interest. Therefore, simultaneously capturing collaborative signals among users, groups, and items is necessary for accurate recommendation in GI task.

The recent developments of graph neural networks (GNNs) have achieved great success in embedding multiple types of nodes in graph-based recommender systems [14, 41, 42, 44, 46]. To name a few, HetGNN [49] integrates Bi-LSTM [12] to capture both structure and content heterogeneity in graphs. GTN [47] learns from a soft selection of edge types and composite relations in heterogeneous graphs through generative meta paths. HGT [13] proposes heterogeneous mutual attention to characterize the attention over different types of edges for modeling heterogeneity. However, edges in most existing graphs only characterize the pair-wise semantics, which contradicts the fact that users join a group as a unified whole. To be more specific, a group is a union of all users in this group. A new user joins this group because of his or her potential relationships with all users in the groups. In this sense, pair-wise connections cannot explicitly express the *union semantics* in groups.

To this end, we adopt hypergraphs [7, 54] for GI task, where hyperedges connect multiple nodes synchronously. We illustrate a toy example of the hypergraph construction for GI task in Figure 1. We construct a GI hypergraph from a user-item-group interaction graph. The hyperedges are items and groups, which are represented as solid and dash lines in Figure 1(b), respectively. The hypergraph structure intrinsically reflects the union semantics of users. For example, the bottom purple triangular in the interaction graph denotes a group, which connects to three users through three edges, whereas, in the GI hypergraph, we convert this group to a hyperedge, denoted as purple dashline, which synchronously connects three nodes, and thus representing the union of three users.

However, previous hypergraph convolution is defective in propagating information on hypergraphs, although it has been widely applied to aggregate information in recommender systems [14, 46]. In previous methods, the information from nodes in a hypergraph is aggregated to hyperedges and then directly aggregated back to those nodes by hypergraph convolution. They neglect the intrinsic

hyperedge information during aggregation, thus being suboptimal for information propagation. For example, if we construct hyperedges to connect the groups joined by the same users in GI problem, previous hypergraph convolution only builds the representations of those groups based on their member composition, but completely ignores the item preferences of their members. Nonetheless, in most e-commercial platforms, the content of a group is up to the popular items attracting the members. For item-related group identification without group-item interaction data available, it is challenging to transmit information on members' item preferences to groups by hypergraph learning.

The other challenge in applying hypergraph learning to GI task is how to harmonize the information learned from different hyperedges for the same nodes, i.e., the information from different views of hypergraphs. For example, although a user's item preferences and group preferences can be modeled through item and group hyperedges discriminatively in the user hypergraph, without integrating these two types of preferences coherently, the recommender system is still unable to deduce her group preferences based on her item preference. We argue that it is crucial to maintain the intrinsic consistency between users' item and group preferences during hypergraph learning. Given no available group-item interaction, this consistency inside each user can be regarded as an alternative group-item connection, which provides necessary information for recommending groups to users.

To this end, we propose a novel framework called Group Identification via Transitional Hypergraph Convolution with Graph Self-supervised Learning (GTGS). GTGS is a hypergraph neural network built upon one user-view group hypergraph, and two item-view and group-view user hypergraphs. We propose a novel transitional hypergraph convolution (THC) layer to generalize existing hypergraph convolution. THC layer endows the model with the ability of characterizing the both hyperedge information propagated from nodes and intrinsically existing in hyperedges. In this way, the intrinsic hyperedge information on users' preferences for items is transmitted to groups. Regarding learning user representation from both group-views, we leverage contrastive self-supervised learning (SSL) to encourage the consistency between item and group preferences of users. After convolution on item-view and group-view user hypergraphs, we obtain two views of embeddings for all users. Then, we develop a novel cross-view self-supervised learning (CSSL) based on InfoNCE contrastive loss [10] to maximize the mutual information between two views of the same user in a self-supervised manner. Additionally, we propose a new group-based regularization term to alleviate the embedding collapse issue for contrastive learning. This group-based regularization is leveraged to enhance the distinction between different groups for constructing distinguishable group representations. The main contributions of this paper are summarized as follows:

- We propose a novel framework GTGS, a model based on hypergraph learning for group identification, which leverages hyperedges to express the union semantics in users and groups.
- We devise a novel THC layer to harness the intrinsic information on hyperedges, which generalizes existing hypergraph convolution and GCN-based models.

- We design a novel CSSL paradigm to optimize user and group representations respectively and propose a group-based regularization to improve the distinction of group embeddings.
- We conduct extensive experiments on three real-world datasets. The significant improvement of GTGS on all datasets indicates its superiority in tackling the GI problem.

2 RELATED WORKS

2.1 Group Recommender Systems

Group recommender systems refer to recommending groups to their potential members. Traditional group recommender systems apply various algorithms to recover user-group membership matrices with available side information. For example, semantic information from descriptions of groups [5] and visual information from photos shared by users [33] can be incorporated with a collaborative filtering framework to perform personalized group recommendations. User behaviors in different time periods [34, 39], such as joining groups, can also be leveraged for recommending groups to users. However, the requirement for side information degrades the performance of those methods when recommending groups to users with only interaction information. Some recent works [17, 18] investigate recommending groups to users with only user-group interactions , while item interaction information is ignored.

Besides recommending groups to users, the term *group recommendation* in literature also refers to recommending items to a group of users [3, 28], which differs from the focus in this paper.

2.2 Hypergraph Learning Based Recommendation

Inspired by the success of hypergraph learning in other graph-based tasks [8, 9], the hypergraph is also introduced in recommender systems due to its capability of modeling complex high-order dependencies through hyperedge-node connections. HGCN-CC [46] improves HGCN [8] with feature crossing and contrastive learning for the user-item recommendation. DHCF [14] utilizes residual connections to consider original features and aggregated related representations simultaneously during modeling the hybrid multi-order correlations between users and items. Hypergraphs can also be constructed based on side information in recommender systems. As an early attempt, MRH [2] builds hyperedges among users, music, and the music tag to construct a hypergraph for music recommendation. MHCN [44] proposes a multi-channel hypergraph convolutional network to enhance social recommendation by leveraging high-order user-user relations. HMF [53] leverages hypergraph structure to describe the interior relation of the social network.

To the best of our knowledge, no previous work has applied hypergraph learning for group recommender systems. Since the hyperedge simultaneously connecting multiple nodes is a natural structure for handling groups of users in recommender systems, some recent works leverage hypergraph learning to predict potential items for given groups of users. HHGR [51] designs hierarchical hypergraph convolution on fine- and coarse-grained hypergraphs. HCR [15] defines an overlap graph of a hypergraph to learn the group's general preference independent of its members' interests.

2.3 Recommendation with Self-Supervised Learning

Self-supervised learning is an emerging paradigm to generate additional supervised signals for improving representation quality. It has reached most recommendation topics and achieved remarkable success. Recommendation models with self-supervised learning can be generally categorized into two branches according to the type of self-supervised learning task: generative methods and contrastive methods. Inspired by the masked language models [6], generative methods learn by reconstructing the input data from its corrupted version [29, 30]. Contrastive methods compare multiple views of input data and contrast positive samples with negative samples to learn discriminative representations [35, 38].

The construction of high-quality self-supervised signals plays a pivotal role in self-supervised learning [31]. Most previous works [30, 35, 43] augment the data to create multiple views as self-supervised signals. However, complex data augmentation cost additional space and time Recent works [21, 45] verify that common data augmentations can bring negative impacts for graph-based tasks. In this work, we design two novel augmentation-free self-supervised learning paradigms to optimize user and group representations.

3 PRELIMINARIES

In this section, we first formulate the problem of group identification (GI), then introduce three hypergraphs for GI and their construction.

DEFINITION 1. *(Ranking-based Group Identification). Given three disjoint node sets, including a user set $\mathcal{U}$, a group set $\mathcal{G}$ and an item set $\mathcal{I}$, and the interactive edges, i.e., user-group edges $E_{\mathcal{U},\mathcal{G}}$ and user-item edges $E_{\mathcal{U},\mathcal{I}}$, an interaction graph is defined as $\mathcal{T} = (V, E)$, where $V = \mathcal{U} \cup \mathcal{I} \cup \mathcal{G}$ and $E = E_{\mathcal{U},\mathcal{G}} \cup E_{\mathcal{U},\mathcal{I}}$. The group identification (GI) for a user u is to predict a ranking list of groups $\{g_1, g_2, \ldots, g_k\}$, with which this user has no interactions in the graph $\mathcal{T}$.*

In other words, we recommend a list of groups that this user u is of potential interest in GI. Note that we distinguish the group as another entity rather than a simple union of users due to its special characteristics, e.g., group information.

DEFINITION 2. *(User-view Group Hypergraph). The user-view group hypergraph of an interaction graph $\mathcal{T}$ is denoted as $\mathcal{T}_g^u = (\mathcal{G}, \mathcal{E}_u)$, where groups $\mathcal{G}$ are nodes and $\mathcal{E}_u$ are hyperedges. We let users be hyperedges. An incidence matrix $\mathbf{H} \in \{0, 1\}^{|\mathcal{G}| \times |\mathcal{U}|}$ is used to represent connections among group nodes.*

Analogously, we define the **Group-view User Hypergraph** and **Item-view User Hypergraph**, denoted as $\mathcal{T}_u^g$ and $\mathcal{T}_u^i$, respectively, whose nodes are users and hyperedges are groups and items, respectively. $\mathcal{T}_u^g$ and $\mathcal{T}_u^i$ are associated with two incidence matrices, denoted as $\mathbf{U}_g \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{G}|}$ and $\mathbf{U}_i \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{I}|}$, respectively. Figure 2 presents an example of constructing the above hypergraphs from an interaction graph.

4 METHODS

In this section, we present the proposed GTGS model for the group identification task. The framework of GTGS is shown in Figure 2. We start by introducing all embedding layers to be trained in this framework. Thereafter, we demonstrate how to leverage users'

preferences for items as prior knowledge in group representation learning. Specifically, we adopt a THC layer to integrate users' preferences learned in the item-view user hypergraph into hyperedge features, and then aggregate these hyperedge features to group nodes. We further design SSL tasks to optimize user and group representations respectively. We propose CSSL to encourage the intrinsic consistency between item-view and group-view user embeddings, and leverage group-based regularization to enhance the distinction between different group embeddings. Lastly, we conduct theoretical analyses on THC and prove it as a general case of graph convolution and hypergraph convolution.

4.1 Embedding layer

We maintain an embedding layer $\mathbf{E} \in \mathbb{R}^{d \times (2|\mathcal{U}| + |\mathcal{G}|)}$, where d is the feature dimension and columns represent all trainable embeddings for users and groups in the three hypergraphs. Each user has two embeddings corresponding to the item-view and group-view user hypergraphs. We denote the initial item-view user embedding as $\mathbf{E}_{\mathbf{u}}^{i,(0)}$, the initial group-view user embedding as $\mathbf{E}_{\mathbf{u}}^{g,(0)}$, and the initial group embedding as $\mathbf{E}_{\mathbf{g}}^{(0)}$.

4.2 Transitional Hypergraph Convolution

Hypergraph convolution is widely used for information propagation among nodes through hyperedges in hypergraphs. However, existing hypergraph convolution methods directly aggregate the information from neighbors connected by hyperedges, while neglecting intrinsic information of hyperedges. Hence, we devise a novel Transitional Hypergraph Convolution (THC) layer to resolve this issue. Specifically, before information propagated from node to node, we first construct the hyperedge information from connected nodes. We denote this initial step as information *construction* of THC layer. Then, we propose an additional *transition* step, which enables the fusion of hyperedge information from other resources. Finally, in the final *propagation* step, fused information is aggregated to nodes for inference of node embedding. Next, to be more concrete, we present the calculation details of a THC layer over a user-view group hypergraph. The illustration is in Figure 2(b).

4.2.1 Information Construction.
In the user-view group hypergraph, group nodes are connected by user hyperedges. Therefore, we construct the information of a hyperedge by aggregating embeddings of all nodes connected by this hyperedge, which is illustrated as in ① of Figure 2(b). The information construction of a hyperedge ϵ is formulated as follows:

$$\mathbf{t}_{\epsilon}^{(l)} = \frac{1}{|\mathcal{N}_{\epsilon}|} \sum_{g' \in \mathcal{N}_{\epsilon}} \mathbf{e}_{g'}^{(l)}, \tag{1}$$

where $\mathbf{t}_{\epsilon}^{(l)}$ denotes temporary embedding of the hyperedge ϵ, $\mathcal{N}_{\epsilon}$ is the set of group nodes connected by this hyperedge, and $\mathbf{e}_{g'}^{(l)}$ is the embedding of those group nodes as input of the l-th THC layer. Eq. (1) represents constructing the hyperedge information via mean-pooling of its connected nodes. We leave other variants such as max-pooling to future work.

4.2.2 Instrinsic Information Transition.
In this step, we fuse the constructed information with intrinsic information of a hyperedge.

Let the intrinsic embedding of a hyperedge ϵ be $\mathbf{c}_{\epsilon}$, we fuse $\mathbf{c}_{\epsilon}$ with $\mathbf{t}_{\epsilon}^{(l)}$ via a *transition layer* as follows:

$$\mathbf{q}_{\epsilon}^{(l)} = \text{Transition}(\mathbf{t}_{\epsilon}^{(l)}, \gamma \mathbf{c}_{\epsilon}^{(l)}), \tag{2}$$

where $\mathbf{q}_{\epsilon}^{(l)}$ is the fused hyperedge embedding, and γ is a scalar hyper-parameter to control the transition intensity. For example, a hyperedge in user-view group hypergraph is a user. Hence, the intrinsic information of the hyperedge is from the user embedding. In this paper, we use add operation as the transition layer and leave the investigation of other variants to future work. In ② of Figure 2(b), we demonstrate this step via adopting the intrinsic information of user hyperedges from the item-view user embeddings, *i.e.*, $\mathbf{C}^{(l)} = \mathbf{E}_{\mathbf{u}}^{i,(l+1)}$, which will be introduced later.

4.2.3 Final Aggregation.
We learn the group embedding by aggregating the fused embedding from all its connected hyperedges. For a target group g, the aggregation step is formulated as follows:

$$\mathbf{e}_{g}^{(l+1)} = \frac{1}{|\mathcal{N}_g|} \sum_{\epsilon \in \mathcal{N}_g} \mathbf{q}_{\epsilon}^{(l)}, \tag{3}$$

where $\mathbf{e}_{g}^{(l+1)}$ denotes the output group embedding, $\mathcal{N}_g$ is the set of hyperedges connected to this group, and $\mathbf{q}_{\epsilon}^{(l)}$ is the fused embedding of its connected hyperedges from Eq. (2). Again, we only investigate the mean-pooling aggregation of hyperedge information and leave other variants to future study. This step is demontrated in Figure 2(b) ③.

4.2.4 THC in Matrix Form.
. To offer a holistic view of convolution, we formulate the matrix form of transitional hypergraph convolution (equivalent to Eq. (1)-(3)) as:

$$\begin{aligned} \mathbf{E}_{\mathbf{g}}^{(l+1)} &= \text{THC}(\mathbf{E}_{\mathbf{g}}^{(l)}, \mathbf{H}, \gamma \mathbf{C}_{\epsilon}^{(l)}) \\ &= \mathbf{D}^{-1}\mathbf{H} \cdot \text{Transition}(\mathbf{B}^{-1}\mathbf{H}^{\top}\mathbf{E}_{\mathbf{g}}^{(l)}, \gamma \mathbf{C}^{(l)}), \end{aligned} \tag{4}$$

where $\mathbf{E}_{\mathbf{g}}^{(l)}$ is the input embedding from l-th layer, $\mathbf{H}$ is the incidence matrix, $\mathbf{C}^{(l)}$ denotes intrinsic hyperedge information, and $\mathbf{E}_{\mathbf{g}}^{(l+1)}$ is the output. $\mathbf{D}$ is the degree matrix of nodes and $\mathbf{B}$ is degree matrix of hyperedges for normalization. THC layer is a more general version of hypergraph convolution, which degrades to existing hypergraph convolution [44] if let $\gamma = 0$ and dismiss the Transition layer.

4.3 Cross-view Self-supervised Learning

Recall that we construct two user hypergraphs, *i.e.,* item-view and group-view user hypergraph. In this section, we introduce how to infer user embeddings from both hypergraphs and how to harness them via contrastive self-supervised learning simultaneously.

4.3.1 User Embeddings.
We employ L THC layers on both item-view and group-view user hypergraphs to infer the item-view user embeddings $\mathbf{E}_u^{i,(L)}$ and the group-view user embeddings $\mathbf{E}_{\mathbf{u}}^{g,(L)}$ as:

$$\begin{aligned} \mathbf{E}_{\mathbf{u}}^{i,(L)} &= \text{THC}^L(\mathbf{E}_{\mathbf{u}}^{i,(0)}, \mathbf{U}_i, 0), \\ \mathbf{E}_{\mathbf{u}}^{g,(L)} &= \text{THC}^L(\mathbf{E}_{\mathbf{u}}^{g,(0)}, \mathbf{U}_g, 0), \end{aligned} \tag{5}$$

where $\mathbf{U}_i$ and $\mathbf{U}_g$ are incidence matrices. $\mathbf{E}_{\mathbf{u}}^{i,(0)}$ and $\mathbf{E}_{\mathbf{u}}^{g,(0)}$ are the input user embeddings. Note that here we eliminate the intrinsic hyperedge information for both views by setting the $\gamma = 0$ in Eq.(2),
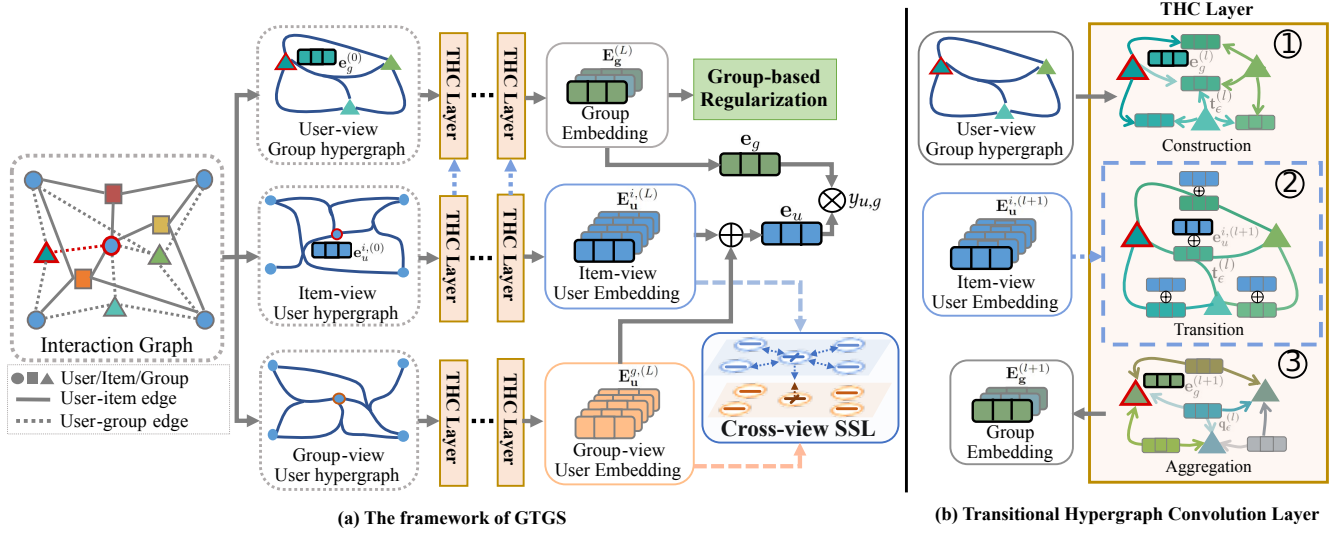
Figure 2: (a) The overall framework of GTGS. First, we construct three hypergraphs and apply THC layers to them. Next, we conduct group-based regularization on the output group embeddings, and employ cross-view SSL to optimize item-view and group-view user embeddings. At last, the inner product of group embedding and item-view user embedding is calculated for prediction; (b) the illustration of the Transitional Hypergraph Convolution (THC) layer.

which retrains simplicity of the model. Intuitively, after L THC layer, the item-view and group-view user embedding characterize the collaborative signals in user-item interaction and user-group interactions, respectively. In the following paper, we ignore the superscript (L) for legibility and simply use $\mathbf{E}_{\mathbf{u}}^i$ and $\mathbf{E}_{\mathbf{u}}^g$ to denote the final embedding from item-view and group-view, respectively. We calculate the final user embeddings via a weighted sum the embeddings from two views as:

$$\mathbf{E}_u = \beta \mathbf{E}_{\mathbf{u}}^i + (1 - \beta) \mathbf{E}_{\mathbf{u}}^g, \tag{6}$$

where β is a hyper-parameter to balance two views.

4.3.2 Cross-view Self-supervised Learning. Though we represent the final user embedding as the sum of two views, the consistency between the two views is however neglected. The intuition is that we should encourage the model to recommend similar groups to users who prefer similar items, and vice versa. We devise a Cross-view Self-Supervised Learning (CSSL) task to enhance the consistency between item-view and group-view user representations as shown in Figure 2(a). To establish consistency between the user representations under the two views, we deploy the InfoNCE loss to maximize the mutual information between the two views:

$$\mathcal{L}_{cssl}^{user} = \sum_{u=0}^{|\mathcal{U}|} - \log \frac{\exp(\text{sim}(\mathbf{e}_u^i, \mathbf{e}_u^g)/\tau_{\mathbf{u}})}{\sum_{v=0}^{|\mathcal{U}|} \exp(\text{sim}(\mathbf{e}_u^i, \mathbf{e}_v^g))/\tau_{\mathbf{u}})}, \tag{7}$$

where $\text{sim}(\cdot, \cdot)$ measures the cosine similarity between two vectors and τ_u is a temperature hyper-parameter. The item-view embedding $\mathbf{e}_u^i$ and group-view embedding $\mathbf{e}_u^g$ is interpreted as users' preference towards items and groups.

This loss function encourages the agreement between the two views of the same user and the distinction between the two views

of different users. In other words, the consistency of the two views is maximized for each individual user.

4.3.3 Group-based Regularization. Inspired by recent works [16, 26] investigating the embedding collapse problem in contrastive learning, we propose a group-based regularization to improve the distinction of group representations. The reason of the embedding collapse issue in our loss function Eq. (7) is that the excessive focus on the closeness of interacted user-group pairs results in a trivial solution of group embeddings collapsing to users who joined plenty of groups. Thus, we apply a regularization to encourage all group embeddings to be uniformly distributed in the space. Concretely, we propose a new variant of the InfoNCE loss to regularize the distribution of group embeddings:

$$\mathcal{L}_{reg}^{group} = \sum_{g=0}^{|\mathcal{G}|} - \log \frac{\exp(1/\tau_{\mathbf{g}})}{\sum_{k=0}^{|\mathcal{G}|} \exp(\text{sim}(\mathbf{e}_g, \mathbf{e}_k)/\tau_{\mathbf{g}})}, \tag{8}$$

where $\mathbf{e}_g$ and $\mathbf{e}_k$ are group embeddings learned by the THC layer, and τ_g is the temperature hyper-parameter. This loss encourages all the group embeddings to be distinct.

4.4 Prediction and Optimization

For GI task, we conduct prediction on the ranking score $y_{u,g}$ of the user-group pair (u, g), which is calculated by the inner product as:

$$y_{u,g} = \mathbf{e}_u \cdot \mathbf{e}_g, \tag{9}$$

where $\mathbf{e}_u$ is final layer user embedding as described in Eq. (6) and $\mathbf{e}_g$ is the final layer group embedding as in Eq. (3). Then we adopt the pairwise Bayesian Personalized Ranking (BPR) loss [27] to optimize

the prediction:

$$\mathcal{L}_{bpr} = \sum_{(u,g,g')\in\mathcal{D}} -\log\sigma(\hat{y}_{u,g} - \hat{y}_{u,g'}), \tag{10}$$

where $\mathcal{D} = \{(u,g,g')|g\in\mathcal{G}_u^+, g'\in\mathcal{G}\backslash\mathcal{G}_u^+\}$ is the training data and group set $\mathcal{G}_u^+$ contains all the groups joined by user u.

Finally, we jointly optimize the recommendation task and the proposed cross-view self-supervised learning as follows:

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda(\mathcal{L}_{cssl}^{user} + \mathcal{L}_{reg}^{group}) + \lambda_\Theta\|\Theta\|_2^2, \tag{11}$$

where λ is the hyper-parameter to control the strength of SSL, and Θ is all trainable parameters in the framework, which is regularized by λ_Θ. Adam [19] is chosen as the optimizer.

4.5 Model Comparison

In this section, we compare the proposed THC with convolution layers in the previous graph and hypergraph convolution based recommendation methods mathematically. We prove that both hypergraph convolution [44] and GCN [20] are special cases of THC.

4.5.1 THC generalizes Hypergraph Convolution. By eliminating the non-linear activation and linear transformation for existing hypergraph convolution works [7, 14, 44], we formulate a hypergraph convolution [44] as:

$$\mathbf{E}^{(l+1)} = \mathbf{D}^{-1}\mathbf{H}\cdot\mathbf{B}^{-1}\mathbf{H}^\top\mathbf{E}^{(l)}. \tag{12}$$

THC is equivalent to hypergraph convolution if we dismiss transition layer **Transition**() and set hyper-parameter $\gamma = 0$ of Eq. (4).

4.5.2 THC generalizes LightGCN. Since graph convolution is a special case of hypergraph convolution when each hyperedge connects only two nodes in hypergraph [1], THC as a general case of hypergraph convolution can also be regarded as a general case of graph convolution. Moreover, THC can be formulated as two consecutive graph convolution layers when each hyperedge connects multiple nodes. We use the LGC layer in LightGCN [11] to represent graph convolution, which is defined as follows:

$$\mathbf{E}^{(l+1)} = (\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{E}^{(l)}, \tag{13}$$

where $\mathbf{E}^{(l)}$ and $\mathbf{E}^{(l+1)}$ are the input and output of a graph convolution layer, and $\mathbf{A}$ is the adjacency matrix reflecting edges. To be more specific, two consecutive graph convolution layers for group-user-group information propagation are denoted as follows:

$$\begin{aligned}\mathbf{E}_{\mathbf{u}}^{(l+1)} &= (\mathbf{D}_u^{-\frac{1}{2}}\mathbf{A}^\top\mathbf{D}_g^{-\frac{1}{2}})\mathbf{E}_{\mathbf{g}}^{(l)}, \\ \mathbf{E}_{\mathbf{g}}^{(l+2)} &= (\mathbf{D}_g^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_u^{-\frac{1}{2}})\mathbf{E}_{\mathbf{u}}^{(l+1)}.\end{aligned} \tag{14}$$

By combining the matrix forms of the two layers, we obtain updated group embeddings as follows:

$$\begin{aligned}\mathbf{E}_{\mathbf{g}}^{(l+2)} &= (\mathbf{D}_g^{-\frac{1}{2}}\mathbf{A}\mathbf{D}_u^{-\frac{1}{2}})(\mathbf{D}_u^{-\frac{1}{2}}\mathbf{A}^\top\mathbf{D}_g^{-\frac{1}{2}})\mathbf{E}_{\mathbf{g}}^{(l)} \\ &= \mathbf{D}_g^{-1}\mathbf{A}\mathbf{D}_u^{-1}\mathbf{A}^\top\mathbf{E}_{\mathbf{g}}^{(l)},\end{aligned} \tag{15}$$

where $\mathbf{A}\in\mathbb{R}^{|G|\times|U|}$ represents interactions between users and groups. Groups sharing a common user are connected together in the group hypergraph. We have its incidence matrix $\mathbf{H} = \mathbf{A}$, node degree matrix $\mathbf{D} = \mathbf{D}_g$ and hyperedge degree matrix $\mathbf{B} = \mathbf{D}_u$. Thus, Eq. (15) is equivalent to Eq. (12), which is a special case of THC.

Table 1: The statistics of datasets.

Dataset	Steam	Beibei	Weeplaces
# users	19,608	11,487	8,550
# groups	46,587	4,035	8,535
# items	3,951	13,814	22,357
# user-group edges	105,271	20,972	16,529
# user-item edges	1,209,979	105,210	152,258
Avg. # groups/user	5.37	1.83	1.93
Avg. # users/group	2.26	5.2	1.94
Avg. # items/user	61.71	9.16	18.44
Avg. # users/item	306.25	7.62	6.81

5 EXPERIMENT

5.1 Experimental Setup

5.1.1 Datasets. We conduct experiments on three real-world datasets: Steam [25], Beibei [50], and Weeplaces [24]. Steam dataset includes users' transaction records and the discussion groups they joined on Steam online game store. Beibei is the largest E-commerce platform for maternal and infant products in China. It records users' purchases and groups they join for group buying. Weeplaces dataset includes users' group-traveling history and their check-in locations. The statistics of three datasets are shown in Table 1. For Steam and Beibei, we randomly select 70% of all groups joined by each user for training and the remaining 30% for testing. For Weeplaces, the split ratio is 80% for training and 20% for testing. We further split 20% validation set from the training set for hyper-parameter tuning. Our source code and the three datasets are released online [2].

5.1.2 Baselines. We compare GTGS with the following baselines:

- **LGCN** [11]. This is the state-of-the-art recommendation method based on GCN [20] by removing feature transformation and nonlinear activation.
- **SGL** [36]. This recommendation framework performs contrastive learning on LGCN to augment node representations by leveraging additional self-supervised signals during training.
- **ENMF** [4]. It is based on a neural matrix factorization architecture leveraging mathematical optimization to train the model efficiently without sampling data for recommendation.
- **HGNN** [7]. It is a method applying hypergraph convolution on a graph for node representation learning. We calculate inner production between user nodes and group nodes as the prediction scores with BPR loss as the objective for recommendation.
- **HCCF** [37]. In this hypergraph learning based recommendation method, contrastive learning is applied to jointly capture local and global collaborative relations.
- **DHCF** [14]. It introduces hypergraph convolution into dual-channel learning for recommendation.
- **LGCN+**. This is a GCN-based recommendation model adapted for GI task. It consists of two LGCN parts to conduct item-user and user-group information propagation consecutively.
- **HGNN+**. This recommendation model is based on HGNN. We adapt it for GI task by employing hypergraph convolution on user-item and user-group bipartite graphs successively.

---
[2]https://github.com/mdyfrank/GTGS

Table 2: Performance comparison on three datasets.

Dataset	Steam				Beibei				Weeplaces			
Metric	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
LGCN	0.0938	0.1236	0.0734	0.0803	0.1042	0.1329	0.0741	0.0821	0.1176	0.1506	0.0705	0.0790
SGL	0.1290	0.1565	0.0879	0.0856	0.1084	0.1377	0.0779	0.0883	0.1192	0.1486	0.0707	0.0784
ENMF	0.1075	0.1386	0.0774	0.0868	0.1154	0.1560	0.0826	0.0959	0.1310	0.1720	0.0734	0.0854
HGNN	0.0398	0.0515	0.0407	0.0463	0.0945	0.1279	0.0864	0.1070	0.0967	0.1367	0.0468	0.0605
HCCF	0.0765	0.0890	0.0698	0.0771	0.0860	0.1236	0.0586	0.0812	0.0818	0.1200	0.0366	0.0505
DHCF	0.1747	0.2082	0.1549	0.1724	0.0828	0.1261	0.0629	0.0876	0.1080	0.1466	0.0633	0.0780
LGCN+	0.1019	<u>0.2619</u>	0.0804	0.1745	0.0925	0.1574	0.0698	0.1007	<u>0.1668</u>	<u>0.2490</u>	<u>0.0824</u>	<u>0.1065</u>
GAT	0.1249	0.1329	0.1155	0.1204	0.0860	0.1701	0.1040	0.1142	0.0350	0.0716	0.0175	0.0298
HGNN+	<u>0.1752</u>	0.1753	<u>0.1985</u>	<u>0.1986</u>	<u>0.1349</u>	<u>0.2003</u>	<u>0.1064</u>	<u>0.1410</u>	0.1500	0.2337	0.0723	0.0956
GTGS	**0.2347**	**0.2903**	**0.2075**	**0.2335**	**0.2125**	**0.2970**	**0.1630**	**0.1931**	**0.2061**	**0.3090**	**0.0994**	**0.1292**
Improv.	33.96%	10.84%	4.53%	17.57%	57.52%	48.28%	53.20%	36.95%	23.56%	24.10%	20.63%	21.31%

- **GAT** [32]. This work leverages a multi-head self-attention architecture to capture influences from each node in graph structures. To adapt it for GI task, we deploy two consecutive graph attention networks for item-user and user-group information aggregation and use pair-wise BPR loss as the objective function.

Since those bipartite recommendation baselines are not designed for graphs with multiple types of interactions, we deploy them with only user-group interactions such that they are adapted to GI task.

5.1.3 Model Settings. The number of THC layers L is set to 1 in our experiments. Learning rate is set to 0.05 for Steam/Beibei and 0.005 for Weeplaces. We set regularization strength λ_θ to 1e-7 for Beibei/Weeplaces and 1e-5 for Steam. Embedding size d is set to 64 for all the datasets. We adopt full-batch training. Early stopping is utilized in all experiments to cope with the over-fitting problem.

5.1.4 Evaluation Metrics. We evaluate GI task by ranking the test groups with all non-interacted groups of users. And we adopt Recall@{10, 20} and NDCG@{10, 20} as evaluation metrics.

5.2 Overall Performance Comparison

Overall comparison results are shown in Table 2. The best results are in boldface, and the second-best results are underlined. The improvement is calculated by subtracting the best performance value of the baselines from that of GTGS and then using the difference to divide the former. We summarize the following key observations:

- The proposed GTGS method achieves the best results and outperforms all the baseline methods by up to 57.52% in the three datasets. We hypothesize these large stable gains result from the abundant and balanced user-item and user-group interactions.
- The baselines integrating both user-item and user-group interaction information, such as LGCN+ and HGNN+, are better than most bipartite recommendation models. It indicates that users' interest in items is able to benefit group identification as side information rather than harmful noise in most cases. However, LGCN+ and GAT are still unable to capture relationships among multiple users or groups synchronously through hypergraphs, thus being worse than GTGS. This observation justifies the necessity of hypergraph learning for GI task.

- Although HGNN+ utilizes hypergraphs to learn user-group and user-item relations, it models these two types of relations independently and neglects the intrinsic consistency between users' preferences for items and groups. Compared with it, GTGS is specifically designed for GI task with transitional hypergraph convolution and cross-view self-supervised learning, which is a better and more stable framework.

5.3 Ablation Study

5.3.1 Hypergraph construction. We demonstrate the performance of GTGS with different item-view user hypergraph constructions on the three datasets in Fig. 3. GTGS utilizes item nodes as hyperedge to construct the item-view user hypergraph. In addition, we investigate three other variants of GTGS: (L) Instead of constructing the item-view user hypergraph, item information is aggregated to user nodes through user-item interactions by one GCN layer; (J1) both item nodes and group nodes are used as hyperedges among user nodes, and hypergraph convolution is employed on both item hyperedges and group hyperedges simultaneously; (J2) same as J1, but hypergraph convolution is first employed on item hyperedges and then on group hyperedges. We only show the result on NDCG since the pattern on Recall is the same.
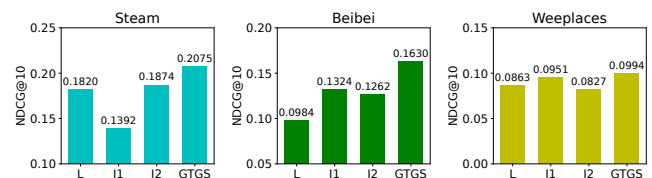
Figure 3: Performance of GTGS w.r.t. different hypergraph constructions.

We observe that GTGS performs the best on three datasets. This justifies the superiority of GTGS with simply item hyperedges that perform best among all item-view user hypergraph constructions. The poor performance of L verifies the limitation of bipartite edges in capturing real-world non-bipartite relations. Since user-item interactions are much denser than user-group interactions in Steam

dataset, the two-stage hypergraph convolution in J2 is able to distinguish the information propagated by item and group hyperedges, which results in better performance than J1. However, given user nodes have been used as hyperedges in the group hypergraph, adding group hyperedges to the item-view user hypergraph increases the risk of over-smoothing between users and groups, so both J1 and J2 are worse than GTGS.

5.3.2 Transitional Hypergraph Convolution Settings. To verify the effectiveness of THC, we show the performance of GTGS with and without THC in Fig. 4. For the model without THC, we set transition intensity $\gamma = 0$ for all THC layers. We can observe that the performance is consistently improved by THC on all three datasets. And the improvement is more obvious in the sparser datasets, such as Beibei and Weeplaces. In these datasets, members' interests in items are more distinct in each group, thus helping the model more accurately capture the characteristics of their group. Therefore, we believe that including users' item preferences in group representations by THC is a better way for GI task.
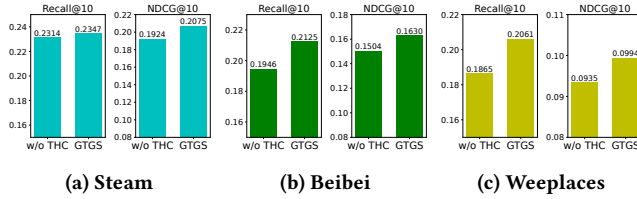

(a) Steam　　　　**(b) Beibei**　　　　**(c) Weeplaces**

Figure 4: Performance of GTGS with and without THC.

Table 3: Ablation study on self-supervised learning settings.

Dataset	Steam		Beibei		Weeplaces	
Metric	R@10	N@10	R@10	N@10	R@10	N@10
w/o SSL	0.2214	0.1829	0.1841	0.1470	0.1821	0.0931
w/o user	0.2242	0.2037	0.1865	0.1367	0.1831	0.0939
w/o group	0.2325	0.1888	0.1900	0.1482	0.1964	0.0989
GTGS	**0.2347**	**0.2075**	**0.2028**	**0.1526**	**0.2061**	**0.0994**

5.3.3 Self-supervised learning settings. Another ablation study is further made to investigate the effectiveness of self-supervised learning. We design three different variants of GTGS. 1) *w/o SSL* is without both CSSL and group-based regularization during the training of GTGS, so the model is optimized only by the BPR loss. 2) *w/o user* is GTGS without the CSSL to maintain the consistency between item-view and group-view user representations. And 3) *w/o group* is GTGS without the group-based regularization to enhance the distinction of group representations. The results are demonstrated in Table 3. We observe that:

- The best setting in general is to apply CSSL and group-based regularization to optimize both user and group representations. This verifies the effectiveness of CSSL applied in GTGS.
- Compared with those variants without the CSSL (i.e., *w/o SSL* and *w/o user* in Table 3), adding it to the model consistently improves the performances in all datasets, which justifies the importance of maintaining the consistency between item and group preferences during the user representation learning.

- Removing the group-based regularization will drop the performance in NDCG@10 largely in the Steam dataset, which has more groups and denser user-group interactions. This indicates that distinction among embeddings is crucial for learning representations of densely connected nodes by hypergraph convolution. Since the embeddings of connected nodes tend to be close in the vector space after convolution [22], it is more important to prevent the features of densely connected nodes from being excessively similar and indistinguishable in such a dataset.
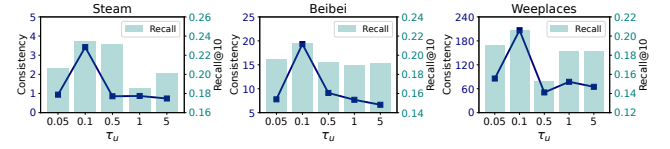
5.4 Cross-view SSL Analysis

Figure 5: Average consistency of user embeddings w.r.t CSSL temperature τ_u. Histograms denote performances in Recall.

As aforementioned, CSSL encourages the intrinsic consistency between item-view and group-view user embeddings for learning comprehensive user representation. We further analyze the influences of user embedding consistency on performance. We define the consistency between item-view and group-view user embeddings as follows:

$$\text{Consistency} = \frac{\mathbb{E}_{u \sim p_{user}} \text{sim}(\mathbf{E}_{\mathbf{u}}^{i,(L)}, \mathbf{E}_{\mathbf{u}}^{g,(L)})}{\mathbb{E}_{u,v \sim p_{user}} \text{sim}(\mathbf{E}_{\mathbf{u}}^{i,(L)}, \mathbf{e}_{\mathbf{v}}^{i,(L)})}, \quad (16)$$

where $p_{user}(\cdot)$ is the distribution of user data and $\text{sim}(\cdot)$ denotes the consine similarity. This consistency score is the ratio of the expected similarity between the two views of a user to the expected similarity between any two users in the dataset. A larger consistency means that the two views of each user learned by GTGS are more consistent. Consistency among user representations learned by CSSL with different temperatures τ_u is shown in Fig. 5 together with corresponding model performances.

We observe that the model has the best performance when consistency achieves the peak, which indicates that the proposed CSSL is able to boost recommendation performance by enhancing the agreement between item-view and group-view user embeddings. Besides, the relatedness between consistency and performance is more obvious in smaller Beibei and Weeplaces datasets. Since each user only interacted with several items in such datasets, her group preference can be clearly revealed by her interests in items. This justifies the CSSL in our design.

5.5 Embedding Analysis

We conduct analyses of the group embeddings learned by GTGS to verify that those embeddings can reflect the user composition of groups through hypergraph learning. We retrieve all pairs of groups. We calculate their relatedness and corresponding common

user ratio. The ratio for each group pair (g_a, g_b) is computed as:

$$r_{ab} = \frac{|\mathcal{N}_{g_a}^{(u)} \cap \mathcal{N}_{g_b}^{(u)}|}{|\mathcal{N}_{g_a}^{(u)} \cup \mathcal{N}_{g_b}^{(u)}|}, \tag{17}$$

where $\mathcal{N}_{g_a}^{(u)}$ and $\mathcal{N}_{g_b}^{(u)}$ is the set of users in group g_a and g_b, respectively. For a simple illustration purpose, we sort all group pairs w.r.t. the pair-wise cosine similarity as the relatedness score and split them into 100 equal size subsets, and represent each subset as the average scores. The scatter plots between relatedness and common user ratio sharing ratio on three datasets are shown in Figure 6. We also draw a regression line and compute the Pearson correlation coefficient p for each dataset. We observe an obvious
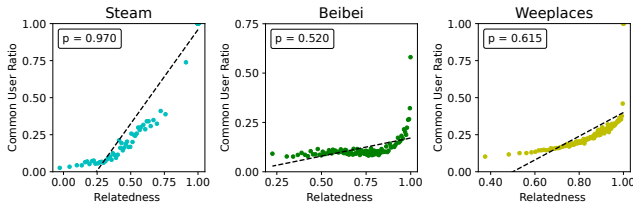
Figure 6: Common user ratio w.r.t relatedness of group pairs based on group embeddings.

overall tendency that the common user ratio increases with the growth of relatedness. This tendency indicates that two groups sharing more common members have higher relatedness based on their embeddings, which justifies the efficacy of hypergraph learning in modeling group representations.

5.6 Cold-start Group Identification
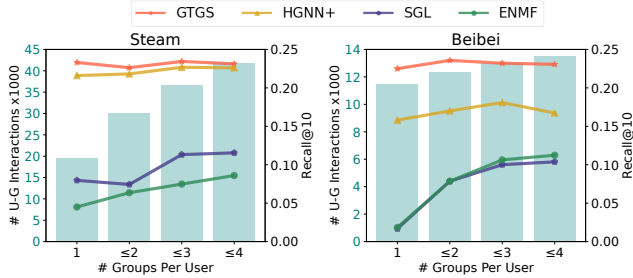
Figure 7: Cold-start performance of different methods. The background histograms denote the number of user-group interactions left in the training set with different thresholds. The solid lines indicate the performances of different methods with respect to different thresholds.

As shown in Table 1, many users have few group interactions, and the cold-start issue in GI task is severe. Hence, we conduct a detailed analysis regarding the ability of GTGS to tackle cold-start group recommendation. We randomly remove some user-group edges for each user in the training set such that the number of neighbor groups of each user is no greater than a threshold k. For comparison, we choose SGL, ENMF, and HGNN+ as the three baseline

models, and perform the experiments with threshold $k \in \{1, 2, 3, 4\}$. The threshold indicates the maximum number of groups per user. The results are shown in Figure 7. We report the Recall performance with respect to different thresholds as the solid line and the number of user-group interactions in the background histograms. First, We observe that GTGS as the method leveraging user's item preferences for GI task significantly outperforms SGL and ENMF as the bipartite user-group recommendation models. The reason is that GTGS can leverage both the item and group interactions to learn node embeddings. Despite the few group participation of users, item interactions complement the cold-start issue. Even if only one user-group edge is left, GTGS is able to achieve outstanding performance in Steam and Beibei datasets. Secondly, although HGNN+ also utilizes hypergraph learning to model user-group relations with user-item relations, it ignores the consistency between users' preferences for items and groups. Thus it relies on user-group edges more heavily and still performs worse than GTGS in Beibei dataset with sparse user-item interactions.

Additionally, we observe that sometimes GTGS even performs better with fewer user-group edges, such as $k = 1$ in Steam dataset. We hypothesize that when the number of groups per user is few, GTGS can well characterize the group interests of users from their item interactions. Therefore, it verifies the GTGS is a better framework to comprehensively integrate item and group information for users and can successfully complete the GI task.

6 CONCLUSION

In this paper, we propose a novel framework GTGS for GI task. GTGS leverages hyperedges to express the union semantics in users and groups for recommendation. A THC layer is devised to transmit information on members' item preferences to groups, such that users' interests in items serve as prior knowledge during group identification. In addition, CSSL is leveraged to guarantee the intrinsic consistency between item and group preferences for each user, and group-based regularization is used to enhance the distinction among group representations. We conduct extensive experiments and detailed analyses on three datasets to verify the effectiveness of GTGS. In the future, we may explore how to design other variants of transition layers in THC for the transition of prior knowledge, so that we can improve the functionality of THC layers.

7 ACKNOWLEDGEMENTS

REFERENCES

[1] Song Bai, Feihu Zhang, and Philip H. S. Torr. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* 110 (2021), 107637. https://doi.org/10.

1016/j.patcog.2020.107637

[2] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. 2010. Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content. In *Proceedings of the 18th ACM International Conference on Multimedia* (Firenze, Italy) *(MM '10)*. Association for Computing Machinery, New York, NY, USA, 391–400. https://doi.org/10.1145/1873951.1874005

[3] 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 and Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 645–654.

[4] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient Neural Matrix Factorization without Sampling for Recommendation. *ACM Trans. Inf. Syst.* 38, 2, Article 14 (Jan 2020), 28 pages.

[5] Wen-Yen Chen, Dong Zhang, and Edward Y. Chang. 2008. Combinational Collaborative Filtering for Personalized Community Recommendation *(KDD '08)*. Association for Computing Machinery, New York, NY, USA, 115–123. https://doi.org/10.1145/1401890.1401909

[6] 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, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[7] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* AAAI Press, 3558–3565. https://doi.org/10.1609/aaai.v33i01.33013558

[8] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* AAAI Press, 3558–3565. https://doi.org/10.1609/aaai.v33i01.33013558

[9] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. 2022. Hypergraph Learning: Methods and Practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 5 (2022), 2548–2566. https://doi.org/10.1109/TPAMI.2020.3039374

[10] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010 (JMLR Proceedings, Vol. 9)*, Yee Whye Teh and D. Mike Titterington (Eds.). JMLR.org, 297–304. http://proceedings.mlr.press/v9/gutmann10a.html

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. *LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation.* Association for Computing Machinery, New York, NY, USA, 639–648.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 2704–2710. https://doi.org/10.1145/3366423.3380027

[14] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual Channel Hypergraph Collaborative Filtering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2020–2029. https://doi.org/10.1145/3394486.3403253

[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)*. 260–269. https://doi.org/10.1109/ICDM51629.2021.00036

[16] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2021. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348* (2021).

[17] Davod Karimpour, Mohammad Ali Zare Chahooki, and Ali Hashemi. 2021. GroupRec: Group Recommendation by Numerical Characteristics of Groups in Telegram. In *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*. 115–120. https://doi.org/10.1109/ICCKE54056.2021.9721494

[18] Davod Karimpour, Mohammad Ali Zare Chahooki, and Ali Hashemi. 2021. Telegram group recommendation based on users' migration. In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*. 1–6. https://doi.org/10.1109/CSICC52343.2021.9420581

[19] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

[20] 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.*

[21] Namkyeong Lee, Junseok Lee, and Chanyoung Park. 2022. Augmentation-Free Self-Supervised Learning on Graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022.* AAAI Press, 7372–7380. https://ojs.aaai.org/index.php/AAAI/article/view/20700

[22] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 433, 8 pages.

[23] Xin Li and Hsinchun Chen. 2013. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decis. Support Syst.* 54, 2 (2013), 880–890. https://doi.org/10.1016/j.dss.2012.09.019

[24] 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, 2014*, Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang (Eds.). ACM, 739–748. https://doi.org/10.1145/2661829.2662002

[25] Mark O'Neill, Elham Vaziripour, Justin Wu, and Daniel Zappala. 2016. Condensing Steam: Distilling the Diversity of Gamer Behavior. In *Proceedings of the 2016 Internet Measurement Conference* (Santa Monica, California, USA) *(IMC '16)*. Association for Computing Machinery, New York, NY, USA, 81–95. https://doi.org/10.1145/2987443.2987489

[26] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining.* 813–823.

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[28] Aravind Sankar, Yanhong Wu, Yuhang Wu, Wei Zhang, Hao Yang, and Hari Sundaram. 2020. *GroupIM: A Mutual Information Maximization Framework for Neural Group Recommendation.* Association for Computing Machinery, New York, NY, USA, 1279–1288.

[29] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019. Pre-training of Graph Augmented Transformers for Medication Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 5953–5959. https://doi.org/10.24963/ijcai.2019/825

[30] 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* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1441–1450. https://doi.org/10.1145/3357384.3357895

[31] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What Makes for Good Views for Contrastive Learning?. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/4c2e5eaae9152079b9e95845750bb9ab-Abstract.html

[32] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[33] Jingdong Wang, Zhe Zhao, Jiazhen Zhou, Hao Wang, Bin Cui, and Guojun Qi. 2012. Recommending Flickr Groups with Social Topic Model. *Inf. Retr.* 15, 3–4 (jun 2012), 278–295. https://doi.org/10.1007/s10791-012-9193-0

[34] Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. 2016. Recommending Groups to Users Using User-Group Engagement and Time-Dependent Matrix Factorization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 1331–1337.

[35] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-Supervised Graph Learning for Recommendation. In

Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 726–735. https://doi.org/10.1145/3404835.3462862

[36] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 726–735.

[37] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy X. Huang. 2022. Hypergraph Contrastive Collaborative Filtering. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 70–79. https://doi.org/10.1145/3477495.3532058

[38] 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. https://ojs.aaai.org/index.php/AAAI/article/view/16578

[39] Lu Yang, Yezheng Liu, Yuanchun Jiang, Le Wu, and Jianshan Sun. 2021. Predicting personalized grouping and consumption: A collaborative evolution model. *Knowledge-Based Systems* 228 (2021), 107248. https://doi.org/10.1016/j.knosys.2021.107248

[40] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S Yu. 2021. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*. 2141–2145.

[41] Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S Yu. 2022. Large-scale personalized video game recommendation via social-aware contextualized graph neural network. In *Proceedings of the ACM Web Conference 2022*. 3376–3386.

[42] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S Yu. 2023. Ranking-based group identification via factorized attention on social tripartite graph. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 769–777.

[43] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix X. Yu, Ting Chen, Aditya Krishna Menon, Lichan Hong, Ed H. Chi, Steve Tjoa, Jieqi (Jay) Kang, and Evan Ettinger. 2021. Self-supervised Learning for Large-scale Item Recommendations. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 4321–4330. https://doi.org/10.1145/3459637.3481952

[44] 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*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 413–424. https://doi.org/10.1145/3442381.3449844

[45] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary?: Simple Graph Contrastive Learning for Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 1294–1303. https://doi.org/10.1145/3477495.3531937

[46] Huanhuan Yuan, Jian Yang, and Jiajin Huang. 2022. Improving Hypergraph Convolution Network Collaborative Filtering with Feature Crossing and Contrastive Learning. *Applied Intelligence* 52, 9 (jul 2022), 10220–10233. https://doi.org/10.1007/s10489-021-03144-1

[47] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2019. Graph Transformer 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*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 11960–11970. https://proceedings.neurips.cc/paper/2019/hash/9d63484abb477c97640154d40595a3bb-Abstract.html

[48] Hongyuan Zha, Xiaofeng He, Chris H. Q. Ding, Ming Gu, and Horst D. Simon. 2001. Bipartite Graph Partitioning and Data Clustering. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*. ACM, 25–32. https://doi.org/10.1145/502585.502591

[49] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 793–803. https://doi.org/10.1145/3292500.3330961

[50] Jun Zhang, Chen Gao, Depeng Jin, and Yong Li. 2021. Group-Buying Recommendation for Social E-Commerce. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*. IEEE, 1536–1547. https://doi.org/10.1109/ICDE51399.2021.00136

[51] 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* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 2557–2567. https://doi.org/10.1145/3459637.3482426

[52] Mingli Zhang, Lingyun Guo, Mu Hu, and Wenhua Liu. 2017. Influence of customer engagement with company social networks on stickiness: Mediating effect of customer value creation. *International Journal of Information Management* 37, 3 (2017), 229–240.

[53] Xiaoyao Zheng, Yonglong Luo, Liping Sun, Xintao Ding, and Ji Zhang. 2018. A Novel Social Network Hybrid Recommender System Based on Hypergraph Topologic Structure. *World Wide Web* 21, 4 (jul 2018), 985–1013. https://doi.org/10.1007/s11280-017-0494-5

[54] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with Hypergraphs: Clustering, Classification, and Embedding. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, Bernhard Schölkopf, John C. Platt, and Thomas Hofmann (Eds.). MIT Press, 1601–1608. https://proceedings.neurips.cc/paper/2006/hash/dff8e9c2ac33381546d96deea9922999-Abstract.html