POI Recommendation for Random Groups Based on Cooperative Graph Neural Networks

Zhizhong Liu[a], Lingqiang Meng[a,*], Quan Z. Sheng[b], Dianhui Chu[c], Jian Yu[d] and Xiaoyu Song[a]

[a]*School of Computer and Control Engineering, Yantai University, Yantai, 264005, China*
[b]*School of Computing, Macquarie University, Sydney, NSW 2109, Australia*
[c]*College of Computer Science and Technology, Harbin Institute of Technology, Weihai, 264209, China*
[d]*Department of Computer Science, Auckland University of Technology, Auckland 1142, New Zealand*

ARTICLE INFO

Keywords:
POI recommendation
Random group recommendation
Weighted signed bipartite graph neural networks
Session-based graph neural networks

ABSTRACT

Group Point-of-Interests (POI) recommendation devotes to find the optimal POIs for groups, which has extracted extensive attention. This work first brings forward a novel POI recommendation model for random groups based on Cooperative Graph Neural Networks (named as CGNN-PRRG). We have done three innovative work. (1) We propose a new fitted presentation learning method for generating the fitted representations of random groups. (2) To conquer the cold start issues in recommending POI for a new random group, we propose to take similar users' (which have the similar representations with that of the random group) POI interaction data as the learning data. (3) We propose an Edge-learning enhanced Bipartite Graph Neural Network (EBGNN) to learn similar users' POI comprehensive interaction preferences. Specially, EBGNN can learn the information on the edges of the graph. Meanwhile, we propose to learn similar users' POI transfer preferences with the Session-based Graph Neural Networks (SRGNN). We verify our proposed model on the three public benchmark datasets (Foursquare, Gowalla and Yelp), which contain 124,933 to 860,888 POI check-in records. The comparison between our proposed model and ten representative baseline models demonstrates the outstanding performance of CGNN-PRRG. In terms of Precision@K and NDCG@K, our model achieves about 24.9% and 62.5% improvement compared with the best baseline models on the three benchmark datasets averagely. Adequate ablation experiments prove the effectiveness of the fitted representation generation method, similar users' POI comprehensive interaction preferences learning method and the method for overcoming the cold start problem. The source code of the CGNN-PRRG model is available on github[1].

1. Introduction

Currently, information technology and different social networks platforms have experienced rapid development (Dokoupil and Peska, 2023; Qi et al., 2019; Sun et al., 2016; El Yebdri et al., 2021; Kajan and Maamar, 2021). In this process, more and more people tend to participate in various activities in groups (Krouska et al., 2023; Ji and Ma, 2023; Zou et al., 2018). For example, some people form a group to take a trip, or to watch movies that they are interested in. The emergence of a large number of group activities has made group-oriented recommendations increasingly important. As well as personalized recommendation (Ye et al., 2023; Liu et al., 2023; Zhan and Xu, 2023; Wang et al., 2023a; Thaipisutikul and Chen, 2023; Gan and Ma, 2023; He et al., 2023), group recommendation devotes to find useful information for groups from a large amount of disorganized information (Khazaei and Alimohammadi, 2018). For instance, as a crowd of people is planning zing to have a trip, group recommendation technology can provide the group with some useful information about the traveling, thereby to overcome the information overload problem for the group.

In the past few years, research on group recommendation has been carried out in many aspects, such as movie recommendation (Seo et al., 2021), music recommendation (Ghazarian and Nematbakhsh, 2015), travel scheme recommendation (Huang et al., 2021b) and others items recommendation (Bahari Sojahrood and Taleai, 2022). Recently, with the emergence of Location Based Social Networks (LBSNs), POI recommendation for groups has gained increasing attention, which aims to recommend POIs to groups while satisfying members' preferences.

[1]https://github.com/IamMichaelMeng/CGNN_PRRG

*Corresponding author: School of Computer and Control Engineering, Yantai University, Yantai, 264005, China.
mlqlsn@126.com (L. Meng)
ORCID(s):

For example, Liu et al. (Liu et al., 2021) developed a POI group recommendation model based on collaborative filtering, while considering group divergence. Sojahrood et al. (Bahari Sojahrood and Taleai, 2022) proposed a POI group recommendation model, which integrates geographical proximity of POIs into users' locations. In work (Zhu et al., 2018), researchers proposed a POI recommendation model, they improved the POI recommendation effect by employing distance-based pre-filtering and distance-based ranking adjustment. Schiaffino et al. (Schiaffino et al., 2020) presented a multi-agent aggregation strategy to conduct the POI group recommendation task.

Actually, in human society, there exists two main types of groups: fixed groups and random groups (Khazaei and Alimohammadi, 2018). A fixed group refers to a number of users who flock together because of the similar and stable interests, which will exist for a longer period of time. While a random group refers to a group of people who get together at a particular moment for a common purpose or joining an activity, this group will dissolve when the goal is achieved or the activity ends. Currently, with the support of rapid development of information technology and socioeconomic, the emergence of random groups is becoming more and more common. Let's consider a scenario where some travelers from all over the word get together at Auckland to explore the city, thus these travelers construct a typical random group (illustrated as Fig.1). For this random group, it would be promising if the recommendation system could recommend some high-quality POIs that can satisfy each member's preferences. Undoubtedly, like the above example, POI recommendation for random groups is wide prevalence in many domains, such as business, education, tourism, and so on.

Figure 1: An example of POI recommendation for a random group.

Random groups have the following typical features: random groups are formed stochastically and its construction is unrepeatable. Moreover, the live period of a random group is usually short. These characteristics lead to the cold start problem when performing POI recommendation for random groups. Besides, for a random group, all members do not have any social relationships with other members, which increases the difficulty of member preference learning.

Currently, although some research has been carried out on POI recommendation for fixed groups, few studies have considered POI recommendation for random groups. Since random groups differ from fixed groups greatly, thus existing POI recommendation approaches for fixed groups cannot be applied to POI recommendation for random groups directly. In all, there are some challenges to be tackled for realizing effective POI recommendation for random groups, which are presented as follows:

- **How to conquer the cold start issues**. Since the construction of random groups is stochastic and unrepeatable, and the lifespans of random groups are usually short. So, there is none POI interaction data of a new random group, which results in cold start problem for POI recommendation for random groups. Thus, how to solve the cold start problem is the first challenge to be tackled.

- **How to generate the fitted representation of a random group**. Existing group recommendation methods always apply the static strategies (Yu et al., 2006; Amer-Yahia et al., 2009; Berkovsky and Freyne, 2010) and the dynamic strategies (Sojahrood and Taleai, 2021; Li et al., 2020) to generate the fitted representation of groups. However, since there are no social relationships among random group members, these two strategies are not suitable for generating the fitted representation of random groups. Therefore, how to generate the fitted representation of a random group is another challenge to be solved.
- **How to learn users' comprehensive POI interaction preferences**. Actually, users' POI interaction preferences usually hide in the graph of users' POI interaction, which play a great impact on recommendation performance. Specifically, the edges of this graph also contain important information. However, existing research just learns the users' POI interaction preferences through node-level learning with GNN-based models, but ignore learning from edges (Yin et al., 2019; Cao et al., 2018; Zan et al., 2021). Thus, how to learn users' POI interaction preferences comprehensively is still a challenge.

To tackle the above issues, this work develops a novel POI recommendation model for random groups based on cooperative graph neural networks (CGNN-PRRG). Specifically, CGNN-PRRG first generates the fitted representation of the random group with our proposed fitted representation learning method. Secondly, to overcome the cold start problem, CGNN-PRRG selects users with the similar representations with that of the random group, and then takes the similar users' POI interaction data as the learning data. Thirdly, with our proposed Edge-learning enhanced Bipartite Graph Neural Networks (EBGNN), CGNN-PRRG learns POIs' representations from the POI interaction bipartite graph, which contains users' POI interaction preferences. Meanwhile, with the Session-based Graph Neural Networks (SRGNN), CGNN-PRRG learns POIs' representations from the POIs transfer directed graph, which imply users' POI transfer preferences. Finally, CGNN-PRRG finishes POI recommendation based on the fitted representation of the random group and the representations of POIs. The contributions are summarized as follows:

- We develop a novel POI recommendation model for random groups based on the cooperative graph neural networks (CGNN-PRRG). To the best of our knowledge, this is the first work considers POI recommendation for random groups.
- To overcome the cold start problem in POI recommendation for random groups, we first detect users with similar representations to that of the random group using the cosine similarity, and then take the similar users' POI interaction data as the CGNN-PRRG's the learning data.
- We propose a method for generating the fitted representation of the random group (FRGRG). FRGRG first computes members' relative influences based on members' POI interaction data. Then, FRGRG calculates members' absolute influences based on weights comparison. Finally, FRGRG generates the fitted representation of the random group based on members' representations and their absolute influences.
- We propose an edge-learning enhanced bipartite graph neural networks (EBGNN) to learn users' comprehensive POI interaction preferences. Different from the existing GNN-based preferences learning models, EBGNN not only learns the node-level information, but also learns the edge-level information in the graph. Thus, EBGNN can learn users' POI preferences comprehensively.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the research objective, problem statements and CGNN-PRRG model in detail. Section 4 reports the experimental results and analysis. Section 5 presents the theoretical and practical implications and discusses our future work.

2. Related Works

2.1. Group Recommendation

Group recommendation aims to recommend a list of items for a group of users based on members' preferences. Recently, group recommendation has become one of the most popular research topics in LBSNs, which not only can enhance users experience, but also can help advertisers to release advertisement for groups (Liu et al., 2021). Currently, there are two different strategies for generating recommendation results for fixed groups: the first strategy conducts recommendations based on members' preferences aggregation, and the second one performs recommendation through predictions aggregation (Liu et al., 2021; Khazaei and Alimohammadi, 2019). In the following sections, we will elaborate the works related to these two strategies respectively.

2.1.1. Group Recommendation Based on Aggregation of Members' Preferences

Actually, group preference is important for performing recommendation for fixed groups. The idea of the preferences aggregation strategy is to recommend items for groups based on the group's preference. The process of this strategy works as follows: it first extracts preferences of group members, and then generates the group's preference by aggregating members' preferences. Secondly, the group is regarded as a virtual user with the aggregated preference, and then it conducts recommendation with a personalized recommendation method (Khazaei and Alimohammadi, 2019). Huang et al. (Huang et al., 2020) proposed a group event recommendation method (named as MAGRM), which first learns the group representation and then adopts the attention mechanism to obtain the recommendation results. However, MAGRM ignores members' personal influence in the group, which makes the group representation learned by MAGRM unable to accurately reflect the characteristics of the group. To improve the precision of recommendations, Liu et al (Liu et al., 2021) developed a POI group recommendation method based on collaborative filtering techniques, which first establishes a group feature preference model to learn the group preference from members' check-in data, and then computes the preference rating of group for each candidate POI based on collaborative filtering. Zhang et al. (Zhang et al., 2021) proposed a self-supervised hypergraph learning model for group recommendation, which applies a hypergraph GCN to get the group representation, and then provides the recommendation for the group based on the group representation and item representations. However, this method did not consider users' interaction preferences learning in the recommendation process.

Work (Guo et al., 2021) presented a group event recommendation model (named as HyperGroup), which connects all groups to construct a hypergraph and converts the group preference learning to embedding hyperedges in the constructed hypergraph. Although this work proposed a joint training method for learning both group-event and user-event interactions, it did not consider the members' influence in the group representation learning. Xu et al. (Xu et al., 2021) proposed a travel recommendation method for groups. In this work, group representation is first constructed based on members' representations, then group recommendations are generated according to members' social influence and global trust. The disadvantage of this method is that it did not learn users' interaction preferences. PGR-ELM (Zhao et al., 2020) is a POI group recommendation method combined with extreme learning machine (ELM) technique. PGR-ELM first extract group's features and POIs' features from members' historical check-in data, and then input the extracted features into a ELM classifier to get the POI recommendation for the group. Our previous work (Meng et al., 2024) proposed a POI recommendation model for occasional groups (named as PROG-HGNN), which first generates the fitted representation of the occasional based on social information of the members, and then recommends POIs for the group by learning group members' POI interaction preferences and POI transfer preferences from members' POI interaction data.

2.1.2. Group Recommendation Based on Predictions Aggregation

For the strategy of group recommendation based on predictions aggregation, it first generates prediction ratings of group members on the items based on members' preferences, and then aggregates the ratings of all members to obtain recommendation results for the fixed groups (Liu et al., 2021). Seo et al. (Seo et al., 2021) proposed a movie recommendation model for groups, which first obtain the members' rating for movies based on the movie genre information, and then the recommendation results for the group are generated by aggregating members' ratings. Although this model can reduce the time complexity in group recommendation, it cannot handle the problem of data sparse and cold start. In work (Logesh et al., 2019), a model named USRiGRM was developed to recommend suitable attractions for traveling groups, which first generates recommendation results for individuals, and then aggregates the personal recommendation list to create the results for groups. Although the performance of model USRiGRM is good, it failed to consider members' influences when generating the fitted representation of the group. Work (Sojahrood and Taleai, 2021) presented a POI group recommendation model, which first generates recommendation results for group members by matrix factorization, and then gets the group recommendation results based on weighted averaging strategy. Ghazarian et al. (Ghazarian and Nematbakhsh, 2015) developed a memory-based technique to realize music recommendation for groups.

By using a support vector machine learning method, the proposed model can calculate the similarity between different musics and predict all unrated musics in the user-music matrix, thus, it can handle the data sparsity problem effectively. However, since it was proposed based on the collaborative filtering method, so, it cannot deeply learn users' interaction preferences from users' history recordings. Zhu et al. (Zhu et al., 2018) proposed a POI group recommendation method which considers the rationality of the POI and the intra-group influence when making group recommendation. It first generates POI recommendation for individuals, and then uses group consensus functions to aggregate the recommendation results for the group. GTSAR-RNN (Li et al., 2020) is a POI group recommendation method which combines user's sentiment information with spatial-temporal contexts. GTSAR-RNN first predict the user's visit intent by calculating the inner product of user embedding and location embedding. and then recommends the potential POIs to users based on the prediction results derived from the first step.

2.2. Research on Recommendation for Random Groups

Random group refers to a group of people who share an environment in a particular moment, without explicit social relationship that linking them (Boratto and Carta, 2011). The nature of the random group is heterogeneous and members always own different preferences (Khazaei and Alimohammadi, 2018). For example, people in a traveling group is one typical random group. In the past decades, some research has been carried out on the random groups and achieved some wonderful results. Wu et al. (Wu et al., 2002) examined the random group effect in housing demand studies, and proposed a two-stage estimation technique to estimate housing demand with the random group effect. Skaggs et al. (Skaggs, 2005) investigated the effectiveness of equating with very small samples using the random groups design. Work (Lehmann and Feldman, 2008) studied the co-evolution of culturally inherited altruistic helping and cultural transmission under the random group formation. Work (Tu et al., 2009) proposed a novel random group mobility model for mobile ad hoc networks, which aims to investigate mobility model in mobile networking.

Recently, some research has been conducted on the recommendation for random groups. Guo et al. (Guo et al., 2019) proposed a random group recommendation model, which firstly creates a preference relation with a multi-variate extreme learning method to forecast unknown preference relations, and then adopts the borda-voting rule to conduct recommendation for the random group. However, this model ignores members' social influence in the group, and it cannot learn users' interaction preferences. Considering members' rating information, work (Ding et al., 2020) developed a movie recommendation model (named as GRMFC) for random groups. GRMFC first converts the random group into a virtual user, and then applies a personalized recommendation method to recommend the most suitable movies for this virtual user. The disadvantage of GRMFC is that it did not consider members' social influence when generating group's representation. To recommend travel schemes for random groups, Wang et al. (Wang et al., 2016) proposed a recommendation model (named as MCS). MCS creates recommendation results for random groups according to the contributions of each members. In work (Pujahari and Sisodia, 2020), a movie recommendation method for random groups was proposed, which leverages the preference relation based matrix factorization to produce unknown ratings, and then aggregates the preferences of the group members using the graph aggregation strategy. Although some wonderful results have been achieved in recommendation for random groups, none research has been conduct on POI recommendation for random groups.

3. POI Recommendation for Random Groups

Currently, POI recommendation for random groups has become a critical challenge, to address this issue, we propose a POI recommendation model for random groups based on cooperative graph neural networks (named as CGNN-PRRG). In this section, we first present the research objective and the problem statement about our work. Then, we introduce the details of our proposed model CGNN-PRRG.

3.1. Research Objective

Recently, with the population of LBSNs, how to recommend suitable POIs for random groups has become a critical problem to be solved urgently. Moreover, existing POI recommendation methods are not suitable for this problem. Our research goal is to address the problem of POI group recommendation for random groups. To realize this goal, the first objective is to generate the fitted representation of the random group and convert the random group into a virtual user (Ding et al., 2020). This strategy is the mainstream in group recommendation research field (Jiang et al., 2019; Chen et al., 2020). For this objective, we propose a fitted representation generation method for generating the fitted representation of the random group based on members' absolute influences. The second objective is to overcome the cold start problem in POI recommendation for random group. For this objective, we take the similar users' POI interaction data as the learning data. The third objective is to learn POIs' representations containing users' complex POI interaction preferences, including user's POI interaction preferences (Lang et al., 2022) and POI transition preferences (Zhang et al., 2022). For this objective, we propose an Edge-learning enhanced Bipartite Graph Neural Networks (EBGNN) to learn POIs' representations from the interaction bipartite graphs between users and POIs, which contains users' POI interaction preferences. Meanwhile, we adopt the Session-based Graph Neural Networks (SRGNN) (Wu et al., 2019) to learn POIs' representations form POIs' transition graphs, which imply users' POI transfer preferences. Our final objective is to recommend the most suitable POIs to the random group based on the above learned fitted representation of the random group and POIs' representations.

3.2. Problem Statement

Definition 1: Users Set. Let $U = \{u_1, u_2, \dots, u_m\}$ indicate users set, m means the number of users, and each user u in U has a unique ID.

Definition 2: POIs Set. A POI is a geographic location with some functionalities that can meet user's demands (such as a coffee shop or a supermarket). A POI has three attributes, including unique id, category c and geographic information d=($long$, lat), where $long$ and lat denotes the longitude and latitude of a POI, respectively. Let $P = \{p_1, p_2, \dots, p_n\}$ represent a set of POIs, and n is the number of POIs.

Definition 3: Random Group. A random group refers to a group of users who get together stochastically because of a brief goal or event. The construction of a random group is usually stochastic and unrepeatable, there is no social relationships between members in a random group. Here, a random group is defined as RG= $\{u_i, u_j, \dots, u_r\}$, where r represents the size of the RG, and RG is a subset of U.

Definition 4: Similar Users' Check-in. Let $S = \{S_i, S_j, \dots, S_r\}$ be the set of POI check-in sequences of users who have similar representations to that of the RG, where $S_i = \{s_1, s_2, \dots, s_i\}$ means the set of check-in sequences of the similar user u_i, and $s_i = \{p_1, \dots, p_k\}$ ($k \in [0, n]$) indicates the i_{th} POI check-in sequence of similar user u_i.

Problem Formulation: Given Users set $U = \{u_1, u_2, \dots, u_m\}$, POIs set $P = \{p_1, p_2, \dots, p_n\}$, $RG = \{u_i, u_j, \dots, u_r\}$, and similar users' check-in $S = \{S_i, S_j, \dots, S_r\}$. Then, the problem studied in this work is how to recommend an optimal POI list for the RG, which can satisfy members' preferences.

3.3. The Framework of model CGNN-PRRG

The structure of our proposed model CGNN-PRRG is illustrated as Fig. 2. CGNN-PRRG consists of four modules, which are fitted representation learning module, similar users' POI interaction preferences learning module, similar users' POI transfer preferences learning module, and POI prediction module. CGNN-PRRG works as follows. Firstly, CGNN-PRRG generates the fitted representation of the random group with Module I. Then, CGNN-PRRG selects users which have the similar representations with that of the random group, and takes similar users' POI interaction data as model's learning dataset. Next, CGNN-PRRG learns POIs' representations which imply similar users' POI interaction preferences and POI transfer preferences with Module II and Module III. Finally, CGNN-PRRG recommends the optimal POIs to the random group with Module IV. The technical details of each module will be introduced in the following sections.

3.3.1. Fitted Representation Learning Module

For group recommendation, the representation of a group is usually calculated based on members' representations and members' weights. Most existing research always calculating members' weights by fusing members' representations and the features of a target item (Yin et al., 2019), we call this kind of weights as the Relative Weight. However, the relative weights cannot reflect member's true importance in a group, because existing research ignores the comparison between members and a representative member in the group when computing members' weights (Zan et al., 2021). Actually, one member in a group will deserve a larger weight when this member owns a greater importance compared with a representative member. Moreover, the weights' comparison between members and a representative member can reflect the fair importance of a member in the group (Zan et al., 2021). Here, we call the weights of members calculated through weight comparison between members and a representative member as the absolute weights.

To generate better fitted representation of a group, based on work (Zan et al., 2021), we propose a new fitted representation generation method for random groups based on members' absolute weights and members' representations (named as FRGRG). Specifically, FRGRG first learns members' representations and relative weights based on members' POI interaction data. Then, FRGRG randomly selects a representative member from the random group, and then calculates members' absolute weights by comparing members' relative weights with the representative member's relative weight. Finally, FRGRG generates the fitted representation of the random group based on the members' absolute weights and representations. The calculation process of FRGRG is illustrated as Fig. 3, which is detailed as follows.

[2] The similar users' POI interaction graph is drawn based on the work (Cai et al., 2022).

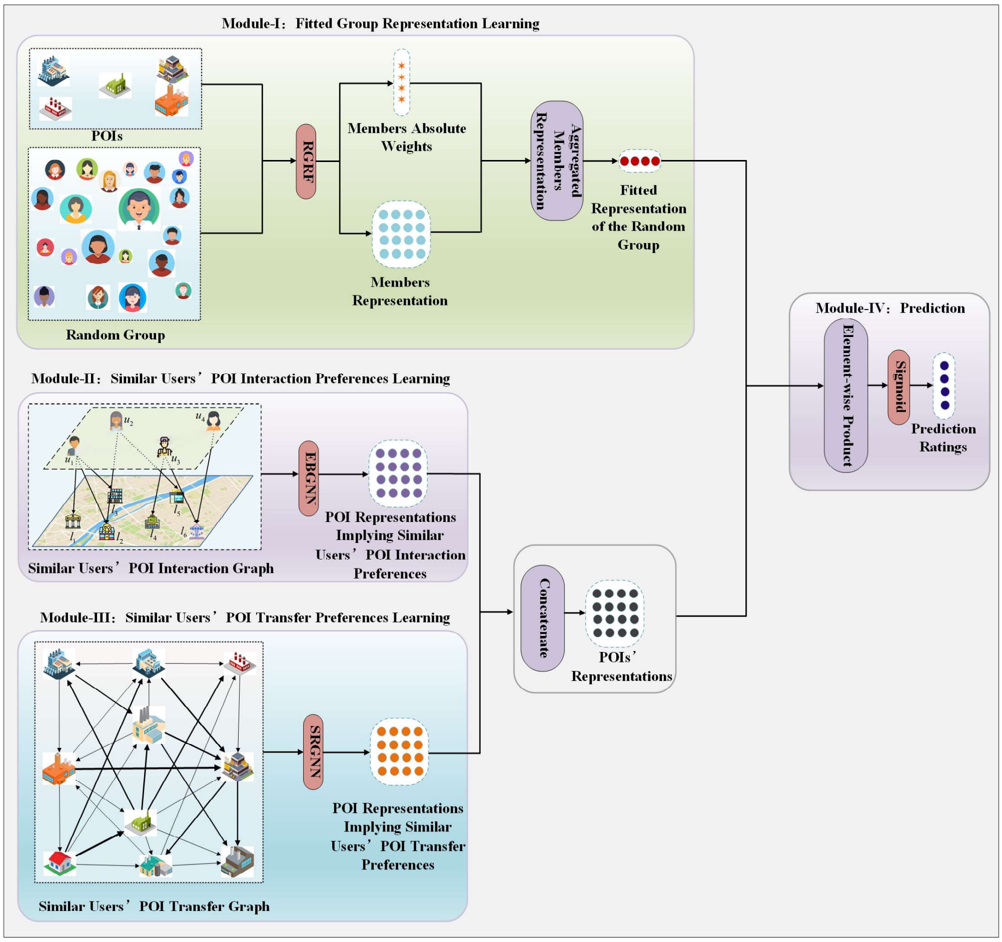

Figure 2: The structure of the CGNN-PRRG model [2].

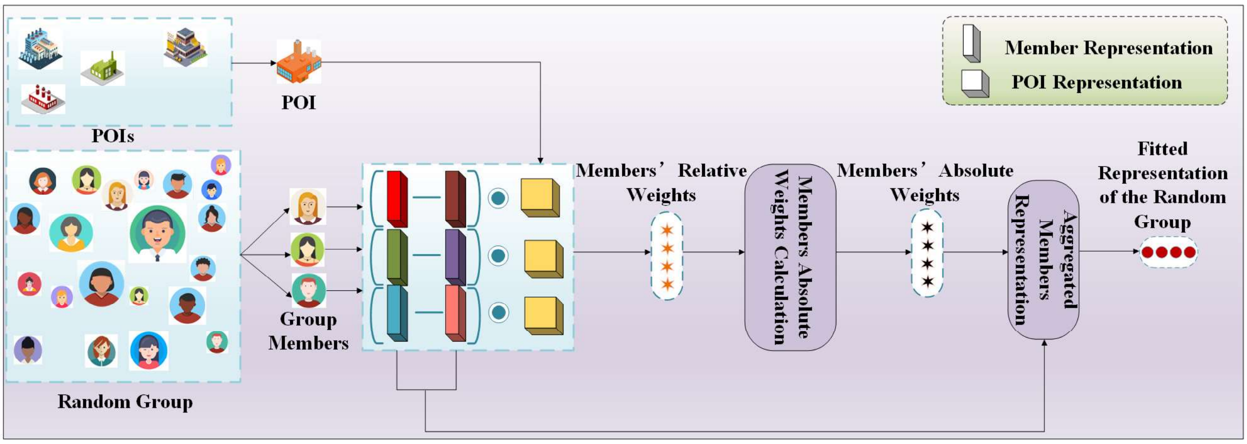

Figure 3: Calculation process of fitted representation generation for random groups.

(1) *Calculating Members' Absolute Weights.*

To obtain members' absolute weights, FRGRG first calculates members' relative weights. That is, for any two group members u_m, u_n and a POI p_i, FRGRG calculates the comparison vector between u_m and u_n on POI p_i, which is described as Eq. (1).

$$\vec{\mathbf{C}}_{mn}^{i} = \left(\vec{\mathbf{u}}_m - \vec{\mathbf{u}}_n\right) \odot \vec{\mathbf{p}}_i \tag{1}$$

where $\vec{\mathbf{C}}_{mn}^{i}$ denotes the comparison vector, $\vec{\mathbf{u}}_m$ and $\vec{\mathbf{u}}_n$ denote the representations of u_m and u_n, respectively. $\vec{\mathbf{p}}_i$ is the representation of POI p_i. The main idea of Eq (1) is that (Bordes et al., 2013), the semantic relationship between two objects can be regarded as a transformation of the difference between them. The semantic relationships is useful for exploring the relationship between different group members, and modeling the decision-making process of the group. For group recommendation, such comparisons can help to identify which member is more familiar with the target POI and has more important influential in the group.

Then, FRGRG fuses other group member's comparison vectors with u_m against POI p_i, which can be conducted as Eq. (2).

$$\vec{\mathbf{C}}_{m}^{i} = \sum_{n=1, n\neq m}^{|U_g|} \vec{\mathbf{C}}_{mn}^{i} = \sum_{n=1, n\neq m}^{|U_g|} \left(\vec{\mathbf{u}}_m - \vec{\mathbf{u}}_n\right) \odot \vec{\mathbf{p}}_i \tag{2}$$

where $\vec{\mathbf{C}}_{m}^{i}$ denotes the fused comparison vector of u_m on POI p_i, which is the sum of the comparison vectors of other group members with u_m against POI p_i.

In the third step, work (Zan et al., 2021) first fuses comparison vectors of different members for the i_{th} item, and then feed them into the MLP model to obtain the weights of group members against the i_{th} item. Different from work (Zan et al., 2021), we do not compute members' weights against a specific item i, but compute weights of members against all the items. The computing process can be described as Eq. (3).

$$\begin{aligned} \vec{\mathbf{h}}_1 &= \left[\vec{\mathbf{C}}_{m}^{1}, \vec{\mathbf{C}}_{m}^{2}, \ldots, \vec{\mathbf{C}}_{m}^{k}\right] \\ \vec{\mathbf{h}}_2 &= \mathrm{Relu}\left(\mathbf{W}_2 \cdot \vec{\mathbf{h}}_1 + \mathbf{b}_2\right) \\ w_m &= \mathrm{sigmoid}\left(\mathbf{W}_3 \cdot \vec{\mathbf{h}}_2 + \mathbf{b}_3\right) \end{aligned} \tag{3}$$

where w_m represents the relative weight of member u_m, $[\cdot]$ concatenates u_m's all fused comparison vectors against all POIs, $Relu(\cdot)$ is the activation function used in the hidden layer, $sigmoid(\cdot)$ function normalizes the relative weights into the range of (0,1). In Eq. (3), we first conduct a concatenation operation for the fused comparison vectors $\vec{\mathbf{C}}_{m}^{k}$, then we input the concatenation result $\vec{\mathbf{h}}_1$ into an MLP model to get the relative weight w_m.

Actually, the relative weight w_m is calculated by performing comparison between different group members against POIs. However, when calculating w_m according to Eq. (3), there is a lack of weight comparison with a fixed representative member, which makes w_m unable to reflect the absolute importance of a member in the group. For this problem, FRGRG first randomly selects a representative member u_k from the group, then compares other members' relative weights with that of the representative member u_k, and finally obtains the members' absolute weights. The calculating process can be described as Eq. (4).

$$w'_m = abs\left(w_m - w_k\right), m \neq k \tag{4}$$

where w_m and w_k mean the relative weight of u_m and u_k. w'_m denotes the absolute weight of u_m, which indicates the absolute influence of u_m in the random group. $abs(\cdot)$ means the absolute value operation. Since w'_m is calculated through weights comparison with a fixed representative member u_k, thus the weights can reflect the fair and absolute importance of the member in the group. Moreover, our experimental results (which is presented in Section 4.9) prove that generating group's fitted representation based on members' absolute weights can further enhance the performance of our proposed model CGNN-PRRG.

(2) *Generating the Fitted Representation.*

After obtaining each member's absolute weight, FRGRG generates the fitted representation of the random group based on members' absolute weights and representations. The formula for fitted representation generation is defined as Eq. (5).

$$\vec{\mathbf{g}} = \sum_{i=0}^{|RG|} w_i' \cdot \vec{\mathbf{u}}_i \tag{5}$$

where $\vec{\mathbf{g}}$ denotes the fitted representation of the random group, which is calculated through weighted sum of members' representations and absolute weights. Different from existing methods, FRGRG not only assigns different weight to each member, but also calculates members' absolute weights through weight comparison. Thus, the fitted representations of random groups produced by FRGRG will be more accurate.

(3) ***Loss Function.***

To train the fitted representation learning module, we take the interaction prediction between group and POIs as the driving task to activate the training process. Firstly, the prediction score is calculated with a three-layer MLP model, which is described as Eq. (6). The input of the MLP model are the original fitted representation of the random group $\vec{\mathbf{g}}$ and the POI representation $\vec{\mathbf{p}}_i$, the input layer of MLP applies the element-wise product $\vec{\mathbf{g}} \odot \vec{\mathbf{p}}_i$ to capture interactions of the second-order feature (Cao et al., 2019).

$$\begin{aligned} \vec{\mathbf{h}}_1' &= \left[\vec{\mathbf{g}} \odot \vec{\mathbf{p}}_i, \vec{\mathbf{g}}, \vec{\mathbf{p}}_i\right] \\ \vec{\mathbf{h}}_2' &= \mathrm{Relu}\left(\mathbf{W}_2' \cdot \vec{\mathbf{h}}_1' + \mathbf{b}_2'\right) \\ S_{gi} &= \mathrm{sigmoid}\left(\mathbf{W}_3' \cdot \vec{\mathbf{h}}_2' + \mathbf{b}_3'\right) \end{aligned} \tag{6}$$

where S_{gi} indicates the predicted score for POI p_i, $sigmoid(\cdot)$ is used to normalize the predicted score into the range of (0,1), $\mathbf{W}_2'$,$\mathbf{W}_3'$,$\mathbf{b}_2'$ and $\mathbf{b}_3'$ represents trainable parameters.

Then, we use members' historical check-in records as the positive POI interaction samples of the random group, and select a set of POI samples that the random group had not interacted with through a random negative sampling method, and apply the regression-based pairwise loss (Cao et al., 2018) to maximize the gap between the positive and negative samples, which is defined as Eq. (7).

$$\mathcal{L} = \frac{1}{|\,\mathrm{Train}\,|} \sum_{(g,i,j)\in \mathrm{Train}} \left(S_{gi} - S_{gj} - 1\right)^2 \tag{7}$$

where $Train$ means the training set, i and j denote the positive and negative POI samples of the random group. After obtaining the loss $\mathcal{L}$, we train the FRGRG according to the back propagation mechanism.

3.3.2. Overcoming the Cold Start Problem

Actually, random groups are formed stochastically and unrepeatable. Therefore, for a new random group, there is none POI interaction record belong to the new random group. Thus, we face the cold start problem when recommending POIs for random groups (Khazaei and Alimohammadi, 2018). To overcome this issue, we first generate the fitted representation of a random group, and take the fitted representation as the representation of a virtual user (Ding et al., 2020), this is the mainstream in treating the groups in group recommendation research field. Then, we transform the POI recommendation for random groups into a personalized POI recommendation for the virtual user. Finally, we find the users that have the similar representations with that of the random group, and take similar users' POI interaction data as the learning data set of model CGNN-PRRG. We argue that similar users' preferences can be used to approximate the random groups' preferences (Jiang et al., 2019; Chen et al., 2020; Xue et al., 2019; Singh et al., 2022). Therefore, it is reasonable for taking similar users' POI interaction data to overcome the cold start problem.

To find the most suitable method for calculating the similarity between users' presentations and the fitted representation of the random group, we test the performance of CGNN-PRRG with three different similarity functions, which are Cosine Similarity, Euclidean Distance and Pearson correlation coefficient. Based on the testing results, we adopt the cosine similarity as the similarity calculation function, which is described as Eq. (8).

$$\mathrm{similarity}(\theta) = \frac{\vec{\mathbf{u}}\cdot\vec{\mathbf{g}}}{\|\vec{\mathbf{u}}\|\|\vec{\mathbf{g}}\|} = \frac{\sum_{i=1}^{K} \vec{\mathbf{u}}_i \cdot \vec{\mathbf{g}}_i}{\sqrt{\sum_{i=1}^{K} \vec{\mathbf{u}}_i^2} \cdot \sqrt{\sum_{i=1}^{K} \vec{\mathbf{g}}_i^2}} \tag{8}$$

where K represents the dimension of the representation, $similarity(\theta)$ denotes the similarity coefficient between users and the random group, $\vec{\mathbf{g}}$ and $\vec{\mathbf{u}}_i$ denotes the representations of the random group and user u_i, respectively.

According to the similarity between each user and the random group, we select a set of similar users for the random group, and take the similar users' POI interaction data as the learning data set for CGNN-PRRG. Experimental results (Introduced in Section 4.8) prove that this method can effectively overcome the cold start problem.

3.3.3. POI Interaction Preferences Learning

For POI recommendation, users' POI interaction preferences have great influences on the effect of recommendation. In this work, to learn similar users' comprehensive POI interaction preferences, we first construct the interaction bipartite graph about users and POIs (Maier and Simovici, 2022; Lang et al., 2022) based on similar users' POI interaction data (shown as Fig. 4(a)). Moreover, considering that different users have different opinions about POIs (including both positive and negative attitudes), we introduce these two kinds of attitudes into the interaction bipartite graph and construct the signed bipartite graphs. Compared with unsigned bipartite graph (Facchetti et al., 2011), the links (edges) of the signed bipartite graph contain more information of users' POI preferences. Thus, we plan to learn users' comprehensive POI interaction preferences from the signed bipartite graph. Recently, work (Huang et al., 2021a) proposed an signed bipartite graph neural networks (SBGNN) to learn the signed bipartite graph. However, SBGNN only focuses on the node-level learning on the graph, and ignores the edge-level learning, which makes SBGNN unable to learn users' comprehensive POI interaction preferences.

To solve the above problem, we first construct the weighted signed interactive bipartite graph based on similar users' POI interaction data (shown as Fig. 4(b)), and then propose an Edge-learning enhanced Bipartite Graph Neural Networks (named as EBGNN) to comprehensively learn similar users' POI interaction preferences from the weighted signed interactive bipartite graph. Different from SBGNN, EBGNN not only learns from the nodes, but also learns from the edges, which enables EBGNN to have a stronger preference learning ability. Next, we will introduce the construction of the weighted signed interactive bipartite graph and the learning process of EBGNN.

(1) *Constructing the Weighted Signed Interactive Bipartite Graph.*

Assume $RG = \left\{u_i, u_j, \ldots, u_r\right\}$ is a random group, $P_G = \left\{p_1, p_2, \ldots, p_k\right\}$ denotes the set of POIs that similar users has visited and $P_G \subseteq P$. $I =< u_i, p_j > (u_i \in RG, p_j \in P_G)$ represent the POI interaction records of similar users. The weighted signed interactive bipartite graph is constructed as follows: If u_i has visited p_j, a positive edge is established between u_i and p_j. Otherwise, a negative edge is established between them. If u_i and u_m have links with same sign on p_j, a positive edge is constructed between u_i and u_m. Otherwise, a negative edge is constructed between these two similar users. Similarly, when u_i has links with same sign on p_k and p_f, a positive edge is constructed between p_k and p_f. Otherwise, a negative edge is constructed between these two POIs.

The check-in frequencies of similar users to POIs are treated as the edge information and placed on the positive edges between similar users and POIs. The weighted signed interactive bipartite graph is illustrated as Fig. 4(b). Let $G_I = (RG, P_G, E_G)$ denote the weighted signed interactive bipartite graph, where RG and P_G indicate the sets of similar users and POIs, E_G refers to the set of connecting edges between nodes in G_I, $E_G = \varepsilon^+ + \varepsilon^-$ and $\varepsilon^+ \cap \varepsilon^- = \varnothing$, where ε^+ and ε^- denotes positive and negative edges, respectively.

(2) *The Learning Process of EBGNN.*

Based on the weighted signed interactive bipartite graph G_I, we propose an Edge-learning enhanced Bipartite Graph Neural Networks (named as EBGNN) to learn POIs' representations, which imply users' comprehensive POI interaction preferences. Specifically, taking G_I as the input of EBGNN, the POIs' representations will be learned by aggregating information from positive and negative neighbors through message propagation, message aggregation and updating mechanisms of EBGNN. The learning details of EBGNN are introduced as follows.

Message Propagation. Let Set_1 denote the set of nodes including both similar users and POIs, and Set_2 denote the set of nodes with same type (such as only users set or POIs set). For the l_{th} layer of EBGNN, $W_p^l \rightarrow +u$ and $W_p^l \rightarrow -u$ are used to propagate the messages from p_j to u_i based on the positive and negative edges. The message propagation process can be illustrated as Eq. (9).

$$
\begin{aligned}
m^l_{p\rightarrow^+u}\left(p_j, u_i\right) &= \mathrm{MSG}\left(\vec{\mathbf{h}}^l_{u_i}, \vec{\mathbf{h}}^l_{p_j}\right) = W^l_{p\rightarrow^+u} \cdot \vec{\mathbf{h}}^l_{p_j} \\
m^l_{p\rightarrow^-u}\left(p_j, u_i\right) &= \mathrm{MSG}\left(\vec{\mathbf{h}}^l_{u_i}, \vec{\mathbf{h}}^l_{p_j}\right) = W^l_{p\rightarrow^-u} \cdot \vec{\mathbf{h}}^l_{p_j}
\end{aligned} \tag{9}
$$

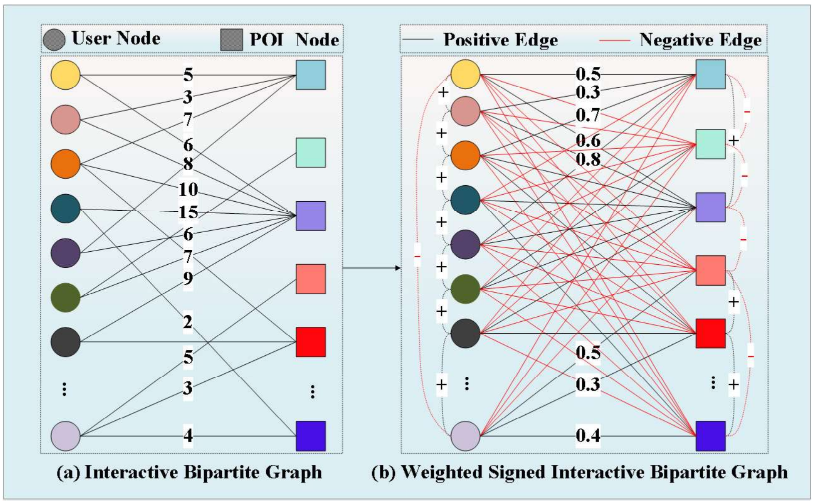

Figure 4: Weight signed interactive bipartite graph construction.

where $N_{p\rightarrow+u}(u_i)$ and $N_{p\rightarrow-u}(u_i)$ denotes the neighbors based on the positive and negative edges to u_i. $p_j \in N_{p\rightarrow+u}(u_i)$ and $p_j \in N_{p\rightarrow-u}(u_i)$. Similarly, we use $W_u^l \rightarrow +p$ and $W_u^l \rightarrow -p$ to propagate the message from u_j to p_i, and the message propagation process is described as Eq.(10).

$$\begin{aligned} m_{u\rightarrow^{+}p}^{l}(u_j, p_i) &= \mathrm{MSG}\left(\vec{\mathbf{h}}_{p_i}^l, \vec{\mathbf{h}}_{u_j}^l\right) = W_{u\rightarrow^{+}p}^{l} \cdot \vec{\mathbf{h}}_{u_j}^l \\ m_{u\rightarrow^{-}p}^{l}(u_j, p_i) &= \mathrm{MSG}\left(\vec{\mathbf{h}}_{p_i}^l, \vec{\mathbf{h}}_{u_j}^l\right) = W_{u\rightarrow^{-}p}^{l} \cdot \vec{\mathbf{h}}_{u_j}^l \end{aligned} \tag{10}$$

where $N_{u\rightarrow^{+}p}(p_i)$ and $N_{u\rightarrow^{-}p}(p_i)$ represent the neighbors with positive and negative edges to p_i. $u_j \in N_{u\rightarrow^{+}p}(p_i)$ and $u_j \in N_{u\rightarrow^{-}p}(p_i)$.

Since here only the representations of POIs are used, so Set_2 refers to the set of POIs. Therefore, we only propagate messages for the positive and negative edges from p_j to p_i with $W_p^l \rightarrow +p$ and $W_p^l \rightarrow -p$, respectively. The message propagation formula for Set_2 is defined as Eq. (11).

$$\begin{aligned} m_{p\rightarrow^{+}p}^{l}(p_j, p_i) &= MSG\left(\vec{\mathbf{h}}_{p_j}^l, \vec{\mathbf{h}}_{p_i}^l\right) = W_{p\rightarrow^{+}p}^{l} \cdot \vec{\mathbf{h}}_{p_j}^l \\ m_{p\rightarrow^{-}p}^{l}(p_j, p_i) &= MSG\left(\vec{\mathbf{h}}_{p_j}^l, \vec{\mathbf{h}}_{p_i}^l\right) = W_{p\rightarrow^{-}p}^{l} \cdot \vec{\mathbf{h}}_{p_j}^l, \end{aligned} \tag{11}$$

where $N_{p\rightarrow^{+}p}(p_i)$ and $N_{p\rightarrow^{-}p}(p_i)$ represents the positive and negative neighbors for p_i, respectively. $p_j \in N_{p\rightarrow^{+}p}(p_i)$ and $p_j \in N_{p\rightarrow^{-}p}(p_i)$.

Message aggregation. The message aggregation operation is performed after message propagation. Here we bring the weights on edges into the learning process. Firstly, EBGNN converts the weights on edges (users' check-in frequencies) into normalized weights and fuses it into the nodes' representations. The process is described as Eq. (12).

$$\begin{aligned} &w_{ij} = \mathrm{NOR}\left(W_{i,j}\right), w_{ij} \in (0,1) \\ &\vec{\mathbf{h}}_i = w_{ij} \cdot \vec{\mathbf{h}}_i' \end{aligned} \tag{12}$$

where $W_{i,j}$ denotes the weights on edges, NOR$(\cdot)$ is the normalization function. Through normalization operation, the weights on edges are transformed into the normalized weights in the range of $(0, 1)$. $\vec{\mathbf{h}}_i$ and $\vec{\mathbf{h}}_i'$ are nodes' representations, $\otimes$ denotes the vector multiplication operation.

Then, EBGNN adopts a graph attention mechanism to aggregate information from the neighbor nodes, here the weight coefficient between two nodes is calculated, which reflects the correlation between the two nodes. The details are described as Eq. (13).

$$\alpha_{ij} = \frac{\exp\left(\text{Leaky ReLU}\left(\vec{\mathbf{a}}^{T}\left[W\vec{\mathbf{h}}_i \| W\vec{\mathbf{h}}_j\right]\right)\right)}{\sum_{k\in N_i}\exp\left(\text{Leaky ReLU}\left(\vec{\mathbf{a}}^{T}\left[W\vec{\mathbf{h}}_i \| W\vec{\mathbf{h}}_k\right]\right)\right)} \tag{13}$$

where $\vec{\mathbf{h}}_i$ is the representation of node i, $\|$ denotes the concatenation operation, W means the weight matrix, $\vec{\mathbf{a}}$ indicates the learnable parameter vector. N_i refers to the neighbors set of node i. $LeakyReLU(\cdot)$ is the activation function. $\exp(\bullet)$ represents the $softmax(\cdot)$ function, which is used to normalize the weight coefficients. Then, the messages from neighbors are aggregated based on the weight coefficient α_{ij}, which is described as Eq. (14).

$$\begin{aligned} m^l\left(u_i\right) &= \sum_{j\in N_i} \alpha^{ij} m^l\left(u_j\right) \\ m^l\left(p_i\right) &= \sum_{j\in N_i} \alpha^{ij} m^l\left(p_j\right) \end{aligned} \tag{14}$$

where N_i represents neighbors set of node i. We can convert the attention aggregation into a function with learnable weighted average mechanism.

Updating representations. After message propagation and aggregation, each u_i obtains four groups of message sets from neighbors, namely $m_p^l \rightarrow +u$, $m_p^l \rightarrow -u$, $m_u^l \rightarrow +u$ and $m_u^l \rightarrow -u$. Each p_i also has four sets of messages from neighbors, that is $m_u^l \rightarrow +p$, $m_u^l \rightarrow -p$, $m_p^l \rightarrow +p$ and $m_p^l \rightarrow -p$. Then, EBGNN aggregates the information of the four groups of neighbor nodes into the node i and obtains the final representation of node i through an MLP. The above process can be detailed as Eq. (15).

$$\begin{aligned} h_u^{l+1} &= MLP\left(\vec{\mathbf{h}}_u^l \left\| m_{p\rightarrow^+u}^l \right\| m_{p\rightarrow^-u}^l \left\| m_{u\rightarrow^+u}^l \right\| m_{u\rightarrow^-u}^l\right) \\ h_p^{l+1} &= MLP\left(\vec{\mathbf{h}}_p^l \left\| m_{u\rightarrow^+p}^l \right\| m_{u\rightarrow^-p}^l \left\| m_{p\rightarrow^+p}^l \right\| m_{p\rightarrow^-p}^l\right) \end{aligned} \tag{15}$$

where $\vec{\mathbf{h}}_u^l$ denote the representation of similar user, $\vec{\mathbf{h}}_p^l$ denotes the representation of POIs, $\|$ means the concatenation operation, MLP is constructed by two fully connected neural layers with a dropout ratio and activation function. The MLP is defined as Eq. (16).

$$\text{MLP}(x) = W_2\left(\sigma\left(\text{dropout}\left(W_1 x + b_1\right)\right)\right) + b_2 \tag{16}$$

where W_1, W_2 and b_1, b_2 denotes the learnable parameters of the MLP, $\sigma(\cdot)$ is an activation function, $dropout$ is used to prevent over-fitting.

The output sequences of EBGNN are denoted as $Z = \left[\vec{\mathbf{z}}_1^p, \vec{\mathbf{z}}_2^p, \ldots, \vec{\mathbf{z}}_k^p\right]$, where $\vec{\mathbf{z}}_i^p$ represents the representation of the POI p_i, which imply users' comprehensive POI interaction preferences for p_i.

(3) *Loss Function.*

To train the comprehensive POI interaction preferences learning module, we select the edge link prediction of graph G_I as the driving task to activate the training of EBGNN. Specifically, we first predict the probability that indicates whether there exists an edge between user u_i and POI p_j based on their representations. Then, we calculate the loss of EBGNN based on the predicted score and the ground truth, and finally train the EBGNN with the back propagation algorithm. The details are presented as follows.

Firstly, we calculate the prediction value for $u_i \rightarrow p_j$ according to an MLP mode, which is described as Eq. (17).

$$y_{\text{pred}} = \text{sigmoid}\left(\text{MLP}\left(\vec{\mathbf{z}}_j^p \| \vec{\mathbf{z}}_i^u\right)\right) \tag{17}$$

where $\vec{\mathbf{z}}_j^p$ denotes the representation of POI, $\vec{\mathbf{z}}_i^u$ is the similar user's representation, y_{pred} is a probability value in the range of $(0, 1)$, which indicates the possibility of having a connecting edge between similar user u_i and POI p_j.

Then, the binary cross entropy is adopted to compute the loss, which is described as Eq. (18).

$$\mathcal{L} = -w\left[y_{\text{true}} \cdot \log y_{\text{pred}} + \left(1 - y_{\text{true}}\right) \cdot \log\left(1 - y_{\text{pred}}\right)\right] \tag{18}$$

where w is a parameter used to overcome the problem of unbalanced number of positive and negative samples, which can make the training process of EBGNN more stable, y_{true} denotes the ground truth which is mapped $\{-1, 1\}$ to $\{0, 1\}$.

3.3.4. POI Transfer Preferences Learning

In social activities, users always transfer from one POI to the next POI, and users' POI transfer preferences hidden in the users' POI check-in records(Zhang et al., 2022), which have significant impact on POI recommendation (Jiang et al., 2023). However, currently few studies consider users' POI transfer preferences in POI group recommendation. To tackle this issue, we plan to conduct similar users' POI transition preferences learning for POI recommendation for random groups. Specifically, we first establish the POI transfer directed graphs according to the similar users' POI check-in sequences. Then, we apply SRGNN (Wu et al., 2019) to learn POIs' representations from the POI transfer directed graphs, which imply similar users' POI transfer preferences. In the following sections, we first introduce the construction of the POI transfer directed graphs, and then we elaborate the learning process of SRGNN.

(1) ***Constructing the POI Transfer Directed Graphs.***

Assume that $S = \{S_i, S_j, ..., S_r\}$ denotes the set of similar users' POI check-in sequences, where $S_i = \{s_1, s_2, \ldots, s_i\}$ represents the set of check-in sequences of the similar u_i, and $s_i = \{p_1, \ldots, p_k\}$ $(k \in [0, n])$ indicates the i_{th} POI check-in sequence of similar user u_i. Based on the check-in sequence s_i, we can construct the POI transfer directed graph $G_T^i = (P_i, E_i)$, where P_i denotes the set of POIs, E_i refers to the set of edges in graph G_T^i, edge $e = (p_k, p_j)$ $(e \in E_i)$ indicates that user u_i visited POI p_j after visiting p_k.

Assume that $A_i \in \mathbb{R}^{n \times 2k}$ denotes the adjacency matrix of graph G_T^i. A_i is defined as the concatenation of two adjacency matrices $A_i^{(in)}$ and $A_i^{(out)}$, which display the weighted connections between the edges of G_T^i. In the graph G_T^i, there exists incoming edge e_{ji}^{in} and outgoing edge e_{ij}^{out} for each node. Considering several POIs may show in a sequence more than once, so we need to allocate normalized weights to these edges (Zhang et al., 2020). Let A_{ji}^{in} denote the weight of incoming edge e_{ji}^{in}, which can be calculated with Eq. (19), and A_{ij}^{out} denote the weight of outgoing edge e_{ij}^{out}, which can be calculated with Eq. (20).

$$A_{ji}^{in} = \frac{\text{count}\left(v_j, v_i\right)}{\sum_{k \in N_{in}(i)} \text{count}\left(v_k, v_i\right)} \tag{19}$$

$$A_{ij}^{out} = \frac{\text{count}\left(v_i, v_j\right)}{\sum_{k \in N_{out}(i)} \text{count}\left(v_i, v_k\right)} \tag{20}$$

where $count(x, y) \in \{0, 1\}$ represent whether there exists an edge between node v_i and v_j, N_{in}^i denotes the predecessor nodes set of node v_i, N_{out}^i is the successor nodes of v_i.

Take a POI check-in sequence $s_i = \{p_1, p_2, p_3, p_2, p_4\}$ for example, the POI transfer directed graph and the corresponding adjacency matrix A_i are described as Fig. 5. According to the directed graph construction method, we can construct many POI transfer directed graphs according to all the POI check-in sequences, and take these POI transfer directed graphs as the input of the SRGNN model.

(2) ***The Learning Process of SRGNN.***

Based on the constructed POI transfer directed graphs, we apply SRGNN (Wu et al., 2019) to learn POIs' representations implying similar users' POI transfer preferences from the constructed POI transfer graphs. SRGNN (Wu et al., 2019) is a session recommendation model developed based on GNNs. It learns users' transfer preferences from their history interaction sequences, and then predicts the next possible interaction item for the user. This feature of SRGNN is suitable for us to adopt it for the similar users' POI transfer preferences learning. Here, we take the graph G_T^i as an example to illustrate the learning process of SRGNN (Wu et al., 2019). Specifically, assuming that $Q_i = \{\vec{\mathbf{q}}_1, \vec{\mathbf{q}}_2, \ldots, \vec{\mathbf{q}}_k\}$ $(\vec{\mathbf{q}}_k \in \mathbb{R}^F)$ represents the set of POIs' representations, where F indicates the dimension, and A_i is the corresponding matrix of the G_T^i. We input the Q_i and A_i into SRGNN, and the representations of POIs will be learned by the gating mechanism of SRGNN. For graph G_T^i, the representations of nodes are aggregated based on the adjacency matrix A_i. The calculation is described as Eq.(21).

$$\vec{\mathbf{a}}_i^t = A_i \cdot \left[\vec{\mathbf{q}}_1^{t-1}, \vec{\mathbf{q}}_2^{t-1}, \ldots, \vec{\mathbf{q}}_n^{t-1}\right]^T H + \vec{b} \tag{21}$$

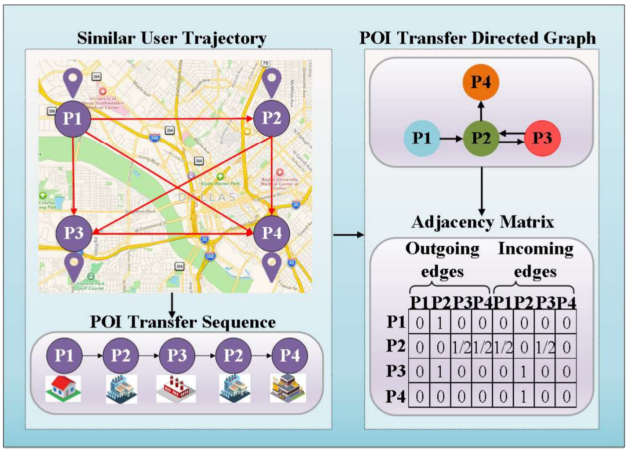

Figure 5: POI transfer directed graph construction.

where A_i denotes the adjacency matrix, which can be used to present the weighted connections of the graph G_T^i, including outgoing and incoming edges, $\vec{\mathbf{q}}_k$ denotes the representation of POI p_k, H refers to the weight matrix, $\vec{b}$ denotes the bias vector.

Secondly, we calculate the values of the update and the reset gate according to Eq. (22). The update gate controls what information should be kept, and the reset gate controls what information should be ignored.

$$\begin{aligned} z_i^t &= \sigma\left(W_z \cdot \vec{\mathbf{a}}_i^t + U_z \cdot \vec{\mathbf{q}}_i^{t-1}\right) \\ r_i^t &= \sigma\left(W_r \cdot \vec{\mathbf{a}}_i^t + U_r \cdot \vec{\mathbf{q}}_i^{t-1}\right) \end{aligned} \tag{22}$$

where z_i^t represents the update gate, r_i^t indicates the reset gate, W and U refer to the model parameters and $\sigma(\bullet)$ denotes the activation function.

Further, the candidate state $\vec{\mathbf{q}}_i^t$ of node p_i is calculated according to its previous state $\vec{\mathbf{q}}_i^{t-1}$, current state $\vec{\mathbf{a}}_i^t$ and reset gate state r_i^t, which is defined as Eq. (23).

$$\vec{\mathbf{q}}_i^t = \tanh\left(W_o \cdot \vec{\mathbf{a}}_i^t + U_o\left(r_i^t \odot \vec{\mathbf{q}}_i^{t-1}\right)\right) \tag{23}$$

Finally, the final state of node p_i is calculated based on the previous hidden state $\vec{\mathbf{q}}_i^{t-1}$ and candidate state $\vec{\mathbf{q}}_i^t$, under the control of the update gate state z_i^t. The calculation process is detailed as Eq. (24).

$$\vec{\mathbf{q}}_i = \left(1 - z_i^t\right) \odot \vec{\mathbf{q}}_{i-1}^t + z_i^t \odot \vec{\mathbf{q}}_i^t \tag{24}$$

The output of SRGNN is denoted as $S = \left[\vec{\mathbf{q}}_1^p, \vec{\mathbf{q}}_2^p \dots, \vec{\mathbf{q}}_k^p\right]$, where $\vec{\mathbf{q}}_i^p$ indicates the representation of POI p_i, which imply similar users' POI transfer preferences for POI p_i.

(3) *Loss Function.*

To train the POI transfer preferences learning model, we take the sequence prediction as the driving task to activate the training of the SRGNN model. Specifically, we predict each candidate POI's probability which indicates the possibility of a POI as the last interacted POI in a sequence. That is, if a POI has a higher predicted probability, it is more likely to be the final interacting POI in a POI sequence. Then, we calculate the loss of SRGNN based on the predicted score and the ground truth value, and finally train the SRGNN with the back propagation algorithm. The process is described as follows.

Firstly, assume that $\vec{\mathbf{s}}_l$ denotes the local embedding of POI transfer sequence s_i, which means the embedding of the last POI ($\vec{\mathbf{q}}_k^p$) of the check-in sequence, thus $\vec{\mathbf{s}}_l = \vec{\mathbf{q}}_k^p$. Then, the global embedding $\vec{\mathbf{s}}_g$ of the POI transfer sequence s_i can be obtained by aggregating all POIs' representations. Since the POIs' representations may have different degrees of priority, we adopt the soft-attention mechanism to depict the global embedding $\vec{\mathbf{s}}_g$, which is illustrated as Eq.(25).

$$\begin{aligned}\alpha_i &= \vec{\mathbf{x}}^{\top} \cdot \sigma\left(W_1 \cdot \vec{\mathbf{q}}_k^p + W_2 \cdot \vec{\mathbf{q}}_i^p + c\right) \\ \vec{s}_g &= \sum_{i=1}^{k} \alpha_i \cdot \vec{\mathbf{q}}_i^p\end{aligned} \tag{25}$$

where parameter vector $\vec{\mathbf{x}}$, W_1 and W_2 control the weights of POIs' representations, c is the trainable parameter.

Thirdly, we compute the hybrid embedding $\vec{\mathbf{s}}_h$ by applying the linear transformation over the concatenation of the local and global embedding representations, which is described as Eq.(26).

$$\vec{\mathbf{s}}_{\mathrm{h}} = W_3 \cdot \left[\vec{\mathbf{s}}_{\mathrm{l}} \| \vec{\mathbf{s}}_{\mathrm{g}}\right] \tag{26}$$

where W_3 is responsible for compressing $\vec{\mathbf{s}}_l$ and $\vec{\mathbf{s}}_g$ into the latent space.

Then, we compute $\hat{y}_i$ for each candidate POI p_i by multiplying its representation $\vec{\mathbf{q}}_i$ with hybrid embedding $\vec{\mathbf{s}}_h$, which can be defined as Eq. (27).

$$\hat{z}_i = \vec{\mathbf{s}}_h^{\mathrm{T}} \cdot \vec{\mathbf{q}}_i^p \tag{27}$$

Finally, we construct a prediction vector $\hat{\vec{\mathbf{z}}}$ over the concatenation of all the prediction scores $\hat{z}_i$, and then use the softmax function to produce the output vector $\hat{\vec{\mathbf{y}}}$, which is illustrated as Eq. (28).

$$\begin{aligned}\hat{\vec{\mathbf{z}}} &= \left[\hat{z}_1, \hat{z}_2, ..., \hat{z}_n\right] \\ \hat{\vec{\mathbf{y}}} &= \sigma\left(\hat{\vec{\mathbf{z}}}\right)\end{aligned} \tag{28}$$

where $\sigma(\cdot)$ is the softmax function, $\hat{\vec{\mathbf{z}}}$ indicates the prediction score for all candidate POIs, $\hat{\vec{\mathbf{y}}}$ denotes the possibility of POIs showing to the last interaction in POI transfer sequence s_i.

For each POI transfer sequence, the loss function is defined based on the prediction and the ground truth according to the cross entropy function, which is defined as Eq. (29).

$$\mathcal{L}\left(\hat{\vec{\mathbf{y}}}\right) = -\sum_{i=1}^{n} \vec{y}_i \log\left(\hat{y}_i\right) + \left(1 - \vec{y}_i\right) \log\left(1 - \hat{y}_i\right), \hat{y}_i \in \hat{\vec{y}} \tag{29}$$

where $\vec{\mathbf{y}}$ is the one-hot encoding vector of the ground truth POI. After obtaining the loss $\mathcal{L}\left(\hat{\vec{\mathbf{y}}}\right)$, SRGNN is trained using the back propagation mechanism according to the calculated $\mathcal{L}\left(\hat{\vec{\mathbf{y}}}\right)$. It should be noted that the length of the most POI transfer sequences is relatively short. Therefore, it is suggested to choose a relatively small number of training epochs to prevent over-fitting.

3.3.5. POI Recommendation

Based on the above learned results (the fitted representation of the random group $\vec{\mathbf{g}}$, two sets of POI representations Z and S), we calculate the rating of the random group on each POI, and then recommend the TOP-K POIs. The details are presented as follows. Firstly, the final representations of POIs ($\vec{\mathbf{h}}_{\mathrm{p}}$) are generated via summation on the two sets of POIs' representations (Z and S), which is defined as Eq.(30).

$$\vec{\mathbf{h}}_{p_i} = \vec{\mathbf{z}}_i^p + \vec{\mathbf{q}}_i^p, \vec{\mathbf{z}}_i^p \in Z, \vec{\mathbf{q}}_i^p \in S \tag{30}$$

Then, the rating of the random group on each POI (denoted as $\hat{y}_i$) is calculated via vector product based on the random group's fitted representation ($\vec{\mathbf{g}}$) and the final POI representation ($\vec{\mathbf{h}}_{p_i}$). For comparison purposes, the rating $\hat{y}_i$ is compressed into the range (0,1) with the *sigmoid* function. The calculation process is illustrated as Eq. (31). Finally, the ratings of the *RG* on all the POIs are ranked in descending order, and the TOP-K POIs are recommended to the random group.

$$\hat{y}_i = \mathrm{sigmoid}\left(\vec{\mathbf{g}}^T \cdot \vec{\mathbf{h}}_{p_i}\right), \hat{y}_i \in (0, 1) \tag{31}$$

3.4. The Training Strategy of the CGNN-PRRG

In our proposed model CGNN-PRRG, Module I learns the fitted representation of the random group, Module II learns POIs' representations from the weighted signed interactive bipartite graph with our proposed EBGNN, and Module III learns POIs presentations from the POI transfer directed graphs with SRGNN. Obviously, these three modules conduct different learning tasks from three different kinds of graphs, the input of the three modules are different, and the training process of the three modules quite different. Therefore, these three modules cannot be trained jointly.

Algorithm 1 : POI Recommendation for Random Group Based on CGNN-PRRG.

Input: U: Users set, P: POIs set, RG: Random group, S: Similar users' POI check-in set.
Output: POI recommendation results for the random group RG.

for each member u_i in RG **do**
    Compute u_i's comparison vector for p_i with Eq. (1)
    Fuse u_i's comparison vector with Eq. (2)
    Calculate u_i's relative influence weight with Eq. (3)
    Calculate u_i's absolute influence weight with Eq. (4)
end for
Generate RG's fitted representation $\vec{\mathbf{g}}$ with Eq. (5)
Construct weighted signed interactive bipartite graph G_I
for each POI p_i in G_I **do**
    Get positive and negative neighbors
    Collect messages from positive and negative neighbors with Eq. (9) and Eq. (10)
    Fuse check-in frequencies into node's representations with Eq. (12)
    Aggregate messages from positive and negative neighbors with Eq. (14)
    Update p_i's representation $\vec{\mathbf{z}}_i^p$ with Eq. (15) and Eq. (16)
end for
Construct POI transfer directed graph G_T^i
for each p_i in G_T^i **do**
    Collect messages from neighbors with Eq. (21)
    Compute values of update and reset gates with Eq. (22)
    Compute candidate state with Eq. (23)
    Compute output state $\vec{\mathbf{q}}_i^p$ with Eq. (24)
end for
Generate final POI representation with Eq. (30)
Compute prediction rating $\hat{y}_i$ with Eq. (31)
Sort prediction ratings in descending order
Select TOP-K POIs as the recommendation result for the RG
end

Moreover, the performance of model CGNN-PRRG depends on the performance of each sub-module. We argue that the performance of CGNN-PRRG would be optimal when each module reaches its optimal performance. To ensure the whole model to get the best performance, we propose to train each module independently. That is, we train the three modules separately based on three different kinds of graphs, then we retain the trained parameters of each module for the POI recommendation for the random group. The parameters for the first three modules are presented in Table 1. The process of POI recommendation for random group based on CGNN-PRRG is described as Algorithm 1.

4. Experiments

To evaluate our proposed model CGNN-PRRG, extensive experiments based on three public benchmark datasets are conducted. In the following sections, we first introduce the three public benchmark datasets and the experimental platform, then we introduce the ten baseline models and present the experimental results.

4.1. Datasets and Experimental Platform

Currently, there are no public POI interaction datasets of random groups. Moreover, in this work, POI recommendation for random groups is converted into personalized POI recommendation by transforming the random group

Table 1
Parameters of the first three modules.

Parameters	FRGRG	EBGNN	SRGNN
num_layers	4	2	2
epochs	32	64	128
batch_size	1,024	500	100
hidden_dim	9	9	9
dropout	0.3	0.5	0.5
learning_rate	0.01	0.005	0.001
optimizer	RMSprop	Adam	Adam
weight_decay	-	1e-5	1e-5

into a virtual user. Therefore, we adopt the three public benchmark datasets (Foursquare[3], Gowalla[4] and Yelp[5]) about personalized POI recommendation to verify the performance of model CGNN-PRRG. There are massive check-in records in these three datasets, and each check-in record includes a unique user ID and POI ID. In our experiment, we preprocess the three datasets by filtering out the inactive users and unpopular POIs. Specifically, we filter out those users who have fewer than 10 check-in POIs and those POIs which are visited by less than 10 users. Then, 60%, 20% and 20% of each processed dataset is selected as the training, validation and testing datasets. The statistics of three datasets are described in Table 2. Unfortunately, we cannot obtain the source codes of baseline models, and we also failed to reproduce these baseline models perfectly as the original papers. Therefore, we did not perform data preprocessing operations on the baseline models.

Table 2
Statistics of the three datasets.

Dataset	Number of users	Number of POIs	Number of check-ins
Foursquare	2,551	13,474	124,933
Gowalla	5,628	31,803	620,683
Yelp	30,887	18,995	860,888

In our experiments, the random groups are generated through randomly selecting some users from the whole users set of the three datasets. It should be noted that the comprehensive POI interaction preferences learning module and POI transfer preferences learning module are trained based on the similar users' POI interaction data, while the fitted representation learning module is trained based on group members' POI interaction data. The experimental platform is as follows. Operating system: Windows 10 Professional 64-bit. CPU:12th Gen Intel(R) Core(TM) i9-12900K 3.19 GHz. GPU: NVIDIA GeForce RTX 3090. RAM: 32GB. Model platform: Pycharm (Community Edition)[6]. Programming language: Python 3.8[7]. Deep learning framework: Pytorch[8]. Moreover, our implementation of the CGNN-PRRG model is available on github[9].

4.2. Evaluation Metrics

In our experiment, we apply two widely-used evaluation metrics to evaluate model CGNN-PRRG, which are Precision ($P@K$) (Wu et al., 2019) and NDCG($NDCG@K$) (Zan et al., 2021). $P@K$ is used to measure how many POIs in the recommended POI list are included in the testing set. $NDCG@K$ is used to measure the accuracy of the recommendation results ranking. The two metrics can be calculated with Eq. (32) and Eq(33).

$$\text{Precision@K}=\frac{|R(g)\cap T(g)|}{|R(g)|} \tag{32}$$

[3]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[4]https://snap.stanford.edu/data/loc-gowalla.html
[5]https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset
[6]https://www.jetbrains.com/pycharm/download/
[7]https://www.python.org/downloads/release/python-380/
[8]https://pytorch.org/
[9]https://github.com/IamMichaelMeng/CGNN_PRRG

$$DCG@K = \sum_{k=1}^{K} \frac{rel_i}{\log_2(i+1)}$$
$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{33}$$

where $R(g)$ represents the recommended POI list for the random group, and $T(g)$ denotes the actual check-in list of the random group, rel_i is the binary value indicating whether the POI at position i in the recommended list appears in the actual check-in list. $IDCG@K$ is the ideal (maximum) $DCG@K$ value among all possible recommendation lists with length K. A higher $NDCG@K$ means better recommendation performance. In our experiments, we use K to denote the length of the POI recommendation list, and K is set in the range of [2, 5, 10, 15, 20].

4.3. Baseline Models

To the best of our knowledge, there is no research has been conducted on POI recommendation for random groups, so there is none model can be used for performance comparison with our proposed model CGNN-PRRG. Moreover, in this work, we transform the problem of POI recommendation for random groups into the problem of personalized POI recommendation by converting a random group into a virtual users. Therefore, we select ten representative personalized POI recommendation models as baseline models to verify the performance of CGNN-PRRG. The ten baseline models are introduced as follows.

TransMKR (Hu et al., 2022): which is proposed based on knowledge graph translation. TransMKR uses POI attributes (such as the access times and locations of POIs) to construct knowledge graph. Based on different attribute values, the representations of POIs can be described in a more detailed and accurate manner. The dimensions of user and POI representation are 8, the number of low layers and the high layers are both 1, the epoch is set as 10, the batch size is set as 32, the weight of L_2 regularization is $1e^{-6}$.

NPGR (Yu et al., 2022): Based on users' POI interaction data, NPGR constructs a heterogeneous graph including users, POI categories and check-in time windows. NPGR applies the Node2Vec method to extract node representations, and considers user check-in frequency, locations and popularity of POIs, and other factors for POI recommendations. In NPGR, γ means the relative importance of geographical influences and is set as 0.1, α and β denote the importance of popularity and category, which are set as 0.4 and 0.2. The best threshold δ is set as 1, the optimal $MinPts$ is set to 7, other parameters $d, n\omega, \omega l, p$ are set as $d = 16, n\omega = 14, \omega l = 25, p = 1$, respectively.

GSTN (Wang et al., 2022): which is proposed based on a graph-enhanced spatio-temporal network. GSTN can effectively capture the relationship between time and space of POI with the GNN technology. The grid search strategy is leveraged to perform hyper-parameter tuning for the GSTN, the learning rate is set as in the range of $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$, the coefficients λ of L_2 regularization is searched in the range of $\{10^{-3}, 10^{-4}, 10^{-5}\}$, the dropout is examined in the range of $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, the length of sequence is set to 20, the threshold of distance is set as 1 km, the dimension of latent representation is 64, K means the negative samples and is set as 5, the cell and hidden state are both initialized as 0, the batch size is 128. GSTN adopts the Adam algorithm as the optimizer, sets learning rate to 0.001 and sets the coefficients λ as 10^{-4}.

STORE (Liu et al., 2022): Compared with the traditional methods of studying time and space separately, STORE checks the factors of time and space simultaneously, and studies impact of spatio-temporal factors (for the recommendation of POIs) on user check-in behaviors. The parameters of STORE are initialized randomly with a gaussian distribution, the value of learning rate is 0.001, Adam is applied as optimizer, the value of batch size is 256, the regularization parameter is set to 0.0001, the embedding dimension of user and POI is set to128, the dimension of spatio-temporal effects and meta-path-based context is 32, the encoding length of Geohash is set as 5. STORE selects 10% of the training data to optimize the ItemKNN's parameters.

MBR (Lang et al., 2022): which is designed based on multivariate GNNs. MBR reduces the computational costs of traditional bipartite graph models through clustering operations on the graph model. To improve the performance, MBR deeply analyzes the influence of the locations of POIs, and the user check-in time on the users' check-in behaviors. We cannot present the parameters of MBR in detail, because the authors did not display the parameters related to the MBR in the paper (Lang et al., 2022).

FG-CF (Cai et al., 2022): which is proposed by combining collaborative filtering with graph convolutional network and can overcome the data sparsity problem. In addition, the user's social information is added to the user-POI bipartite graph, so that the user's representations can be described more accurately. The batch size is 1024. The dimension of user and POI representation is set to 64, the number of connectivity layers is set to 3, the dimension of each layer is set as 64, the learning rate is selected from $\{0.0001, 0.001, 0.01, 0.05, 0.1\}$, the coefficient of L_2 normalization is in the range of $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, the dropout ratio is set in $\{0.1, ..., 0.9\}$. FG-CF initialize model's parameters with Xavier initializer and adopts Adam to optimize the model.

FedPOIRec (Perifanis et al., 2023): FedPOIRec focuses on the privacy protection when perform POI recommendation. It combines federated learning techniques with preferences from users social friends to generate recommendation results. The presentation dimension is set to 128, learning rate L_2 is set to 10^{-6}. The Adam optimizer is used to optimize the model. As for the federated setting, FedPOIRec use the SGD technique with learning rate $\eta = 10^{-1}$.

DSMR (Wang et al., 2023c): DSMR is a deep semantic POI recommendation model, which adopts a prompt engineering to conduct semantic modeling and uses a pre-trained language model to learning deep semantic information. The presentation dimension d is set to 128, dropout rate is 0.3, Adam optimizer is used to train the model, learning rate is 10^{-4} with batch size 40 with 0 epochs for the first training process, and for the second training process, the batch size is set to 20 with 200 epochs.

TGSTAN (Cao et al., 2023): TGSTAN is an improved spatial–temporal attention network, which includes a dynamic graph convolution network module to capture the local spatial correlations and update the graph weights in real time and learn user's personalized local spatial relationships. The attention head is 2, the dropout is 1.0 and 0.7 for two different datasets, learning rate is set to 10^{-3}, weight decay is $5e-4$, the training epochs is 35, and Adam optimizer is selected to train the model.

STaTRL (Wang et al., 2023b): StaTRL is an unvisited POI recommendation method by learning spatial-temporal and text representation. It adopts multi-task learning and is trained to simultaneously recommend top-k POIs and predict user preferences. The learning rate of STaTRL and that of the ABSA encoder were searched in $[1e-5, 1e-4]$, and $[1e-6, 5e-6, 1e-5, 5e-5, 1e-4]$, respectively. The number of the Geo-Transformer layers N was searched in $\{1, 2, 3, 4, 5\}$, and the number of heads in multi-attention was searched in $\{4, 8, 12, 16\}$.

4.4. Influence of Group Sizes on the Performance of CGNN-PRRG

In our experiment, random groups are randomly generated with different sizes, and the sizes of random groups have important impact on the performance of CGNN-PRRG. In this experiment, we aim to investigate the effects of group sizes on the performance of CGNN-PRRG, and then determine the group sizes on different datasets when CGNN-PRRG performs well. Specifically, we set the value of group sizes from 10 to 100, respectively, and adopt Precision as the metric. Then, we assess CGNN-PRRG's performance with Precision under different group sizes on the three datasets. Experimental results are presented in Fig. 6.

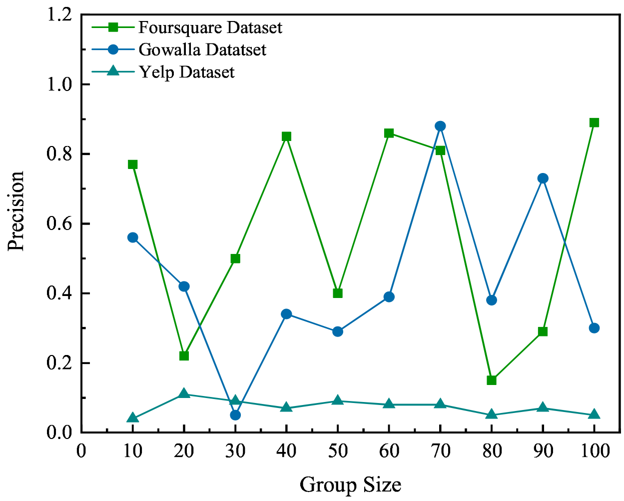

Figure 6: Influence of different group sizes on CGNN-PRRG.

From Fig. 6 we can find that, the Precision of CGNN-PRRG fluctuates with the increase of the group sizes on the three datasets. Specifically, on dataset Foursquare, the Precision reaches the best when the group size is equal to 100.

On dataset Gowalla, CGNN-PRRG gets the highest Precision when the group size is set as 70. For dataset Yelp, the Precision of CGNN-PRRG gets the maximum when the group size is equal to 20. Based on the above experimental results we can conclude that, when the group sizes are small, the training data for CGNN-PRRG is insufficient, which influences CGNN-PRRG's performance. Otherwise, when the group size is too large, the differences of all the group members will also impact the performance of CGNN-PRRG. Therefore, group size is a significant factor that influences the effect of POI recommendation for random groups, and selecting an appropriate group size is crucial for obtaining the best performance of the recommendation model. In the following experiments, we set the group sizes as 100, 70, and 20 on data sets Foursquare, Gowalla and Yelp.

Table 3
Performance comparison results on the Foursquare dataset.

Models	Precision@K					NDCG@K				
	2	5	10	15	20	2	5	10	15	20
STORE*	-	-	-	-	-	-	0.13	0.16	-	-
GSTN*	-	-	-	-	-	0.16	0.2	0.22	-	-
NPGR*	-	0.05	0.04	0.04	0.04	-	-	-	-	-
TransMKR*	-	-	0.40	0.35	-	-	-	-	-	-
FedPOIRec*	-	0.21	0.15	-	-	-	-	-	-	-
DSMR*	-	0.51	0.59	-	-	-	-	-	-	-
TGSTAN*	-	0.47	0.57	-	-	-	0.34	0.41	-	-
CGNN-PRRG*	**0.98**	**0.89**	**0.75**	**0.58**	**0.45**	**0.98**	**0.91**	**0.81**	**0.68**	**0.58**

4.5. Performance Verification

This experiment aims to verify the performance of our proposed model CGNN-PRRG. In this experiment, we compare CGNN-PRRG with other ten baseline models in terms of Precision@K and NDCG@K. Performances comparison on the three data sets are presented in Table 3, Table 4 and Table 5, where the best results in each table are marked with boldface. The performances of model CGNN-PRRG are obtained by running it on our experiment platform. Unfortunately, we didn't find the source codes of baseline models, thus we cannot get the experimental results of the baseline models through running them on our experiment platform. Therefore, we take the experimental results presented in the published works of baseline models for performance comparison. Moreover, the values of K and metrics used by baseline models are inconsistent with model CGNN-PRRG, to make performance comparison as fair as possible, we select metrics that commonly used by most baseline models for performance comparison (such as Precision@5, etc.), and this is why there are some missing values in Table 3, Table 4 and Table 5 (the missing values are marked with symbol "-"). In the three tables, "*" indicates that the model has been trained with the preprocessed datasets. In the following sections, we will analyze the comparison results of each benchmark dataset.

On dataset Foursquare, models STORE, GSTN, NPGR, TransMKR, FedPOIRec, DSMR and TGSTSN are selected as the baseline models, experimental results are presented in Table 3. From Table 3 we can find that, for metric Precision@K, it gets the highest value of 0.98 when K is set as 2. After that, Precision@K decreases to 0.45 when K is set as 20. As for NDCG@K, it first reaches the maximum value of 0.98 when K is equal to 2, and then gradually decreases. When K is equal to 20, NDCG@K reaches the minimum value of 0.58. Furthermore, based on Table 3 we can observe that, CGNN-PRRG model improves Precision@K, and NDCG@K by 30% and 60%, respectively.

On dataset Gowalla, we adopt FG-CF, MBR, GSTN, TransMKR and TGSTAN as the baseline models, experimental results are presented in Table 4. From Table 4 we can find that, with K increases from 2 to 20, the values of Precision@K shows a fluctuating tendency. Specifically, it first reaches the peak value 0.88 when K increases from 2 to 5. After that, it gradually decreases to 0.45 when K increases from 5 to 20. For metric NDCG@K, it keeps an downward trend from 0.88 to 0.57 when K increases from 2 to 20. Although NDCG@K displays a downward trend, CGNN-PRRG is still superior to GSTN. Moreover, based on Table 4 we can observe that, CGNN-PRRG model improves Precision@K and NDCG@K by 44% and 65%, respectively.

For dataset Yelp, since this dataset is rarely used in performance verification of POI recommendation models, among the baseline models, only FG-CF and STaTRL has been tested on this dataset. Therefore, we select FG-CF and STaTRL as the baseline model for performance comparison with the CGNN-PRRG model. Experimental results are presented in Table 5. From Table 5 we can find that, metrics Precision@K shows a decrease tendency from 0.13 to

Table 4
Performance comparison results on the Gowalla dataset.

Models	Precision@K					NDCG@K				
	2	5	10	15	20	2	5	10	15	20
FG-CF	-	0.09	0.07	0.06	0.06	-	-	-	-	-
MBR*	-	0.03	0.02	0.02	0.02	-	-	-	-	-
GSTN*	-	-	-	-	-	0.12	0.15	0.17	-	-
TransMKR*	-	0.43	0.42	-	-	-	-	-	-	-
TGSTAN*	-	0.30	0.40	-	-	-	0.23	0.26	-	-
CGNN-PRRG*	**0.87**	**0.88**	**0.76**	**0.58**	**0.45**	**0.88**	**0.88**	**0.81**	**0.67**	**0.57**

Table 5
Performance comparison results on the Yelp dataset.

Models	Precision@K					NDCG@K				
	2	5	10	15	20	2	5	10	15	20
FG-CF	-	**0.28**	0.02	0.02	0.02	-	-	-	-	-
STaTRL*	-	0.04	0.035	-	0.03	-	-	-	-	-
CGNN-PRRG*	**0.13**	0.11	**0.1**	**0.09**	**0.07**	**0.13**	**0.11**	**0.1**	**0.09**	**0.08**

0.07 as K increases from 2 to 20. As for NDCG@K, it also keeps decreasing from 0.13 to 0.08 when K increases from 2 to 20. It should be noted that the FG-CG did not use NDCG@K as the evaluation metric, so we cannot conduct the performance comparison between CGNN-PRRG, FG-CF and STaTRL based on NDCG@K. Moreover, based on Table 5 we can observe that CGNN-PRRG model improves Precision@K by 0.75%.

Based on the above experimental results, we come to the conclusions that: (1) On the three benchmark datasets, the overall performance of CGNN-PRRG are always better than that of the other baseline models, which can prove that model CGNN-PRRG is effective; (2) The performance improvement of CGNN-PRRG in POI recommendation is very significant. This is because that CGNN-PRRG incorporates more information, such as group members' absolute influence information, similar users' POI interaction information and POI transition information, while the baseline models did not take the above information into consideration (Hu et al., 2022; Yu et al., 2022; Lv et al., 2022). Moreover, CGNN-PRRG converts POI interaction information into graph-structured data and applies GNN-based models for information learning, which enables CGNN-PRRG can learn deeper and complex information, which is beneficial for improving the performance of CGNN-PRRG.

4.6. Ablation Experiments

Our proposed model CGNN-PRRG consists of four modules. The first module is used for learning the fitted representation of a random group, the second module is used for learning users' POI interaction preferences, the third module is applied to learn users' POI transfer preferences, and the fourth module performs POI recommendation. The first module and the fourth module can be removed from CGNN-PRRG, because without these two modules, CGNN-PRRG cannot work. Therefore, in this ablation experiment, we validate the effectiveness of POI interaction preferences learning module and POI transfer preferences learning module in CGNN-PRRG. In this experiment, we compare CGNN-PRRG with its two variants, namely EBGNN-PRRG and SRGNN-PRRG, which are obtained by removing the POI interaction preferences learning module or POI transfer preferences learning module from CGNN-PRRG. The two variants are defined as follows:

- **EBGNN-PRRG:** which is obtained by removing the POI transfer preferences learning module from the CGNN-PRRG model. It only contains the group fitted representation learning module, POI interaction preferences learning module and POI prediction module, while ignoring users' POI transfer information.
- **SRGNN-PRRG:** which is obtained by removing the POI comprehensive interaction preferences learning module from the CGNN-PRRG model. It only consists of the fitted representation learning module, POI transfer preferences learning module, while ignoring POI interaction information.

Table 6
Results of the ablation experiment on the Foursquare dataset.

Models	Precision@K					NDCG@K				
	2	5	10	15	20	2	5	10	15	20
EBGNN-PRRG	0.933	**0.887**	0.743	0.542	0.413	0.941	0.903	0.801	0.653	0.548
SRGNN-PRRG	0.033	0.020	0.023	0.020	0.017	0.033	0.025	0.025	0.022	0.020
CGNN-PRRG	**0.983**	0.886	**0.747**	**0.578**	**0.453**	**0.980**	**0.913**	**0.809**	**0.682**	**0.581**

We use Precision@K and NDCG@K as the performance evaluation metrics, and compare the performance of CGNN-PRRG, EBGNN-PRRG and SRGNN-PRRG on the Foursquare dataset. Experimental results are presented in Table 6. From Table 6 we can find that, SRGNN-PRRG has the lowest Precision@K and NDCG@K value on the Foursquare dataset. Besides, the performance of EBGNN-PRRG is superior to SRGNN-PRRG, but weaker than that of the CGNN-PRRG, which demonstrates that POI transfer preferences learning module has weaker influence on the performance of CGNN-PRRG than that of the POI interaction preferences learning module. Moreover, except Precision@5, the experimental results prove that CGNN-PRRG performs better than EBGNN-PRRG and SRGNN-PRRG in all cases, and we can calculate from the Table 6 that CGNN-PRRG model improves Precision@K and NDCG@K by 2.58% and 2.38%, respectively. Therefore, we can conclude that exploiting users' POI interaction preferences and POI transfer preferences are useful in enhancing the whole performance of CGNN-PRRG.

4.7. Validation of Edge Learning in EBGNN

In this work, we propose an Edge-learning enhanced Bipartite Graph Neural Networks (EBGNN) to comprehensively learn users' POI interaction preferences from the weighted signed interactive bipartite graph. Different from SBGNN, in EBGNN, we add edge learning operation to learn information from edges in the graph. To validate the effectiveness of edge-learning operation in EBGNN, we first generate a variant of the CGNN-PRRG model (namely CGNN-SBGNN), and then compare the performances of CGNN-PRRG and CGNN-SBGNN. CGNN-SBGNN is generated by removing the edge-learning mechanism in the similar users' POI interaction preferences learning module, the other modules of CGNN-SBGNN are the same with CGNN-PRRG.

Table 7
Validation experimental results of the edge-learning enhanced operation on the Gowalla dataset.

Models	Precision@K					NDCG@K				
	2	5	10	15	20	2	5	10	15	20
CGNN-SBGNN	0.833	0.789	0.578	0.411	0.317	0.833	0.804	0.655	0.525	0.443
CGNN-PRRG	**0.866**	**0.880**	**0.763**	**0.582**	**0.448**	**0.881**	**0.882**	**0.805**	**0.674**	**0.569**

We use Precision@K and NDCG@K as the performance evaluation metrics, and compare the performance of CGNN-PRRG and CGNN-SBGNN on the Gowalla dataset. Experiment results are presented in Table 7. From Table. 7 we can find that, CGNN-PRRG is always better than that of CGNN-SBGNN in all cases, and we can calculate that CGNN-PRRG model improves Precision@K and NDCG@K by 12.22% and 11.02%, respectively. These experimental results can prove that our proposed edge-learning mechanism is effective, and is useful for comprehensively learning similar users' POI interaction preferences.

4.8. Validation of the Idea for Overcoming the Cold Start Problem

To overcome the cold start problem in POI recommendation for random groups, we take the similar users' POI interaction data as the CGNN-PRRG's learning data. The motivation behind this method is to approximate the preferences of the random group by learning the preference of similar users. This idea is a typical collaborative filtering strategy widely used in the field of recommendation system research, which approximates the preferences of target users through similar ones when their preferences are unknown. Therefore, we argue that our idea for overcoming the cold start problem in POI group recommendation for random groups is reasonable.

To verify the effectiveness of this idea, we first train the CGNN-PRRG model with two different POI interaction datasets, which are similar users' POI interaction data(named as CGNN-PRRG-Similar) and ordinary users' POI

Table 8
Validation experimental results of overcoming the cold start problem on the Yelp dataset.

Models	Precision@K					NDCG@K				
	2	5	10	15	20	2	5	10	15	20
CGNN-PRRG-Ordinary	0.033	0.073	0.073	0.076	0.070	0.033	0.063	0.067	0.070	0.067
CGNN-PRRG-Similar	**0.133**	**0.113**	**0.096**	**0.087**	**0.073**	**0.125**	**0.114**	**0.102**	**0.094**	**0.084**

interaction data (named as CGNN-PRRG-Ordinary). Then, we compare the performances of the two trained CGNN-PRRG models in terms of Precision@K and NDCG@K on the Yelp dataset. Experimental results are presented in Table 8. From Table 8 we can find that, the performance of CGNN-PRRG trained with similar users' POI interaction data are always better than that of CGNN-PRRG trained with ordinary users' POI interaction data in all cases, and we can compute the model trained with similar users' POI interaction data improves Precision@K and NDCG@K by 5.52% and 4.38%, respectively. Thus, we can conclude that taking similar users' POI interaction data as the CGNN-PRRG's learning dataset is an effective to overcome the cold start problem in POI recommendation for random groups.

4.9. Validation of Fitted Representation Generation Method

To validate the fitted representation generation method proposed for random groups, we first generate a variant of the CGNN-PRRG model, namely CGNN-RELA, which is obtained by removing the absolute influence calculation in the fitted representation learning module, the other modules of CGNN-RELA are consistent with CGNN-PRRG model. Then, we compare the performance of CGNN-PRRG and CGNN-RELA in terms of Precision@K and NDCG@K on the Foursquare dataset. Experimental results are presented in Table. 9. From Table. 9 we can find that, CGNN-PRRG outperforms CGNN-RELA in all cases, and we can compute that the CGNN-PRRG improves Precision@K and NDCG@K by 15.02% and 12.44%, respectively. Therefore, we can conclude that calculating members' absolute influence is effective when generating the random group fitted representation, and is useful for to improving the performance of model CGNN-PRRG.

Table 9
Validation experimental results of the fitted representation generation method on the Foursquare dataset.

Models	Precision@K					NDCG@K				
	2	5	10	15	20	2	5	10	15	20
CGNN-RELA	0.950	0.760	0.513	0.376	0.293	0.954	0.820	0.627	0.511	0.431
CGNN-PRRG	**0.983**	**0.886**	**0.747**	**0.578**	**0.453**	**0.980**	**0.913**	**0.809**	**0.682**	**0.581**

5. Conclusion

5.1. Theoretical and Practical Implications

In this work, we propose a POI recommendation model for random groups based on Cooperative Graph Neural Networks (named as CGNN-PRRG). Specifically, CGNN-PRRG first generates the fitted representation of the random group with the proposed fitted representation learning method (named as FRGRG). Secondly, to overcome the cold start problem, CGNN-PRRG selects users with the similar representations with that of the random group, and then takes the similar users' POI interaction data as the learning data. Thirdly, with the proposed Edge-learning enhanced Bipartite Graph Neural Networks (named as EBGNN), CGNN-PRRG learns POIs' representations from the POI interaction bipartite graph by using the proposed EBGNN, which contains users' POI interaction preferences. Meanwhile, with the Session-based Graph Neural Networks (SRGNN), CGNN-PRRG learns POIs' representations from the POIs transfer directed graph, which imply users' POI transfer preferences. Finally, CGNN-PRRG conducts POI recommendation task based on the fitted representation of the random group and the representations of POIs.

The performance comparison between CGNN-PRRG and ten baseline models on three popular benchmark datasets proves that CGNN-PRRG has a more powerful capability to perform POI recommendation for random groups. Moreover, based on the ablation experimental results, we find that users' POI transfer preferences have a weaker influence on the performance of CGNN-PRRG than that of the POI interaction preferences. Therefore, it is meaningful to put more attention on the users' POI interaction preferences learning in the future work.

CGNN-PRRG also has good practical implications. For example, CGNN-PRRG effectively fills the gap that there is no method for POI recommendation for random groups. In addition, the proposed Edge-learning enhanced Bipartite Graph Neural Networks can be applied for other graph learning in personalized recommendation, service recommendation, and so on. Besides, the idea of adopting similar users' historical data to alleviate the cold start problem can also be used for other problems with few historical data.

5.2. Limitations and Future Work

Although the proposed model CGNN-PRRG has a powerful capability to perform POI recommendation for random groups, it failed to take full use of contextual factors (such as time, location, weather, transportation, and so on) in the group recommendation process. Moreover, users' preferences are easy to be affected by the specific contextual environment factors. In addition, POI group recommendation model should consider both the current state and context information to select the most suitable POIs for the group. Therefore, in our future work, we plan to integrate contextual information into POI recommendation and conduct research on context-aware POI recommendation for random groups.

CRediT Authorship Contribution Statement

Zhizhong Liu: Conceptualization, Supervision, Investigation, Methodology, Writing-Original draft preparation. **Lingqiang Meng**: Experiment Programming, Writing-Reviewing and Editing. **Quan Z. Sheng**: Experiment Programming, Validation, Writing-Reviewing and Editing. **Dianhui Chu**: Formal Analysis, Validation, Visualization. **Jian Yu**: Writing-Reviewing and Editing. **Xiaoyu Song**: Writing-Reviewing and Editing.

Data Availability

Data will be made available on request.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant nos. 62273290, 61872126, 61832004, 62272405), Key projects of Shandong Natural Science Foundation (Grant no. ZR2020KF019), Special Funding Program of Shandong Taishan Scholars Project, The Natural Science Foundation of Shandong Province under Grant ZR2022MF238, Yantai Science and Technology Innovation Development Plan Project under Grant 2022XDRH023, Australian Research Council (ARC) Future Fellowship FT140101247 and Discovery Project DP200102298.

References

Amer-Yahia, S., Roy, S. B., Chawlat, A., Das, G., and Yu, C. (2009). Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment*, 2(1):754–765.

Bahari Sojahrood, Z. and Taleai, M. (2022). Behavior-based poi recommendation for small groups in location-based social networks. *Transactions in GIS*, 26(1):259–277.

Berkovsky, S. and Freyne, J. (2010). Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 111–118.

Boratto, L. and Carta, S. (2011). State-of-the-art in group recommendation and new approaches for automatic identification of groups. In *Information retrieval and mining in distributed environments*, pages 1–20. Springer.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Cai, Z., Yuan, G., Qiao, S., Qu, S., Zhang, Y., and Bing, R. (2022). Fg-cf: Friends-aware graph collaborative filtering for poi recommendation. *Neurocomputing*, 488:107–119.

Cao, D., He, X., Miao, L., An, Y., Yang, C., and Hong, R. (2018). Attentive group recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 645–654.

Cao, D., He, X., Miao, L., Xiao, G., Chen, H., and Xu, J. (2019). Social-enhanced attentive group recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 33(3):1195–1209.

Cao, G., Cui, S., and Joe, I. (2023). Improving the spatial–temporal aware attention network with dynamic trajectory graph learning for next point-of-interest recommendation. *Information Processing & Management*, 60(3):103335.

Chen, J., Zhao, C., Uliji, and Chen, L. (2020). Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering. *Complex & Intelligent Systems*, 6:147–156.

Ding, Z., Qin, Z., Wang, Q.-X., and Qin, Z.-G. (2020). Random group recommendation model based on fuzzy clustering. *Journal of Electronic Science and Technology*, 18(2):100054–100061.

Dokoupil, P. and Peska, L. (2023). The effect of similarity metric and group size on outlier selection & satisfaction in group recommender systems. In *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, pages 296–301.

El Yebdri, Z., Benslimane, S. M., Lahfa, F., Barhamgi, M., and Benslimane, D. (2021). Context-aware recommender system using trust network. *Computing*, 103(9):1919–1937.

Facchetti, G., Iacono, G., and Altafini, C. (2011). Computing global structural balance in large-scale signed social networks. *Proceedings of the National Academy of Sciences*, 108(52):20953–20958.

Gan, M. and Ma, Y. (2023). Mapping user interest into hyper-spherical space: A novel poi recommendation method. *Information Processing & Management*, 60(2):103169–103186.

Ghazarian, S. and Nematbakhsh, M. A. (2015). Enhancing memory-based collaborative filtering for group recommender systems. *Expert Systems with Applications*, 42(7):3801–3812.

Guo, L., Yin, H., Chen, T., Zhang, X., and Zheng, K. (2021). Hierarchical hyperedge embedding-based representation learning for group recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–27.

Guo, Z., Zeng, W., Wang, H., and Shen, Y. (2019). An enhanced group recommender system by exploiting preference relation. *IEEE Access*, 7:24852–24864.

He, L., Wang, X., Wang, D., Zou, H., Yin, H., and Xu, G. (2023). Simplifying graph-based collaborative filtering for recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 60–68.

Hu, B., Ye, Y., Zhong, Y., Pan, J., and Hu, M. (2022). Transmkr: Translation-based knowledge graph enhanced multi-task point-ofinterest recommendation. *Neurocomputing*, 474:107–114.

Huang, J., Shen, H., Cao, Q., Tao, S., and Cheng, X. (2021a). Signed bipartite graph neural networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 740–749.

Huang, Z., Liu, Y., Zhan, C., Lin, C., Cai, W., and Chen, Y. (2021b). A novel group recommendation model with two-stage deep learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(9):5853–5864.

Huang, Z., Xu, X., Zhu, H., and Zhou, M. (2020). An efficient group recommendation model with multiattention-based neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4461–4474.

Ji, P. and Ma, X. (2023). A fuzzy intelligent group recommender method in sparse-data environments based on multi-agent negotiation. *Expert Systems with Applications*, 213:119294–119311.

Jiang, L., Cheng, Y., Yang, L., Li, J., Yan, H., and Wang, X. (2019). A trust-based collaborative filtering algorithm for e-commerce recommendation system. *Journal of ambient intelligence and humanized computing*, 10:3023–3034.

Jiang, S., He, W., Cui, L., Xu, Y., and Liu, L. (2023). Modeling long-and short-term user preferences via self-supervised learning for next poi recommendation. *ACM Transactions on Knowledge Discovery from Data*.

Kajan, E. and Maamar, Z. (2021). Knowledge engineering in the social era. In *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 2150–2155. IEEE.

Khazaei, E. and Alimohammadi, A. (2018). An automatic user grouping model for a group recommender system in locationbased social networks. *ISPRS International Journal of Geo-information*, 7(2):67–84.

Khazaei, E. and Alimohammadi, A. (2019). Context-aware group-oriented location recommendation in location-based social networks. *ISPRS International Journal of Geo-Information*, 8(9):406.

Krouska, A., Troussas, C., and Sgouropoulou, C. (2023). A novel group recommender system for domain-independent decision support customizing a grouping genetic algorithm. *User Modeling and User-Adapted Interaction*, pages 1–28.

Lang, C., Wang, Z., He, K., and Sun, S. (2022). Poi recommendation based on a multiple bipartite graph network model. *The Journal of Supercomputing*, 78(7):9782–9816.

Lehmann, L. and Feldman, M. W. (2008). The co-evolution of culturally inherited altruistic helping and cultural transmission under random group formation. *Theoretical population biology*, 73(4):506–516.

Li, G., Chen, Q., Zheng, B., Yin, H., Nguyen, Q. V. H., and Zhou, X. (2020). Group-based recurrent neural networks for poi recommendation. *ACM Transactions on Data Science*, 1(1):1–18.

Liu, Y., Yang, Z., Li, T., and Wu, D. (2022). A novel poi recommendation model based on joint spatiotemporal effects and fourway interaction. *Applied Intelligence*, 52(5):5310–5324.

Liu, Y., Yin, M., and Zhou, X. (2021). A collaborative filtering algorithm with intragroup divergence for poi group recommendation. *Applied Sciences*, 11(12):5416.

Liu, Z., Xu, W., Zhang, W., and Jiang, Q. (2023). An emotion-based personalized music recommendation framework for emotion improvement. *Information Processing & Management*, 60(3):103256–103267.

Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., and Li, X. (2019). Efficient user profiling based intelligent travel recommender system for individual and group of users. *Mobile Networks and Applications*, 24:1018–1033.

Lv, Y., Sang, Y., Tai, C., Cheng, W., Shang, J. S., Qu, J., Chu, X., and Zhang, R. (2022). Online meta-learning for poi recommendation. *GeoInformatica*, 27:61–76.

Maier, C. and Simovici, D. (2022). Bipartite graphs and recommendation systems. *Journal of Advances in Information Technology-in print*.

Meng, L., Liu, Z., Chu, D., Sheng, Q. Z., Yu, J., and Song, X. (2024). Poi recommendation for occasional groups based on hybrid graph neural networks. *Expert Systems with Applications*, 237:121583.

Perifanis, V., Drosatos, G., Stamatelatos, G., and Efraimidis, P. S. (2023). Fedpoirec: Privacy-preserving federated poi recommendation with social influence. *Information Sciences*, 623:767–790.

Pujahari, A. and Sisodia, D. S. (2020). Aggregation of preference relations to enhance the ranking quality of collaborative filtering based group recommender system. *Expert Systems with Applications*, 156:113476–113487.

Qi, L., Wang, R., Hu, C., Li, S., He, Q., and Xu, X. (2019). Time-aware distributed service recommendation with privacy-preservation. *Information Sciences*, 480:354–364.

Schiaffino, S., Godoy, D., Pace, J. A. D., and Demazeau, Y. (2020). A mas-based approach for poi group recommendation in lbsn. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection: 18th International Conference, PAAMS 2020, L'Aquila, Italy, October 7–9, 2020, Proceedings 18*, pages 238–250. Springer.

Seo, Y.-D., Kim, Y.-G., Lee, E., and Kim, H. (2021). Group recommender system based on genre preference focusing on reducing the clustering cost. *Expert Systems with Applications*, 183:115396–115412.

Singh, P. K., Sinha, S., and Choudhury, P. (2022). An improved item-based collaborative filtering using a modified bhattacharyya coefficient and user–user similarity as weight. *Knowledge and Information Systems*, 64(3):665–701.

Skaggs, G. (2005). Accuracy of random groups equating with very small samples. *Journal of Educational Measurement*, 42(4):309–330.

Sojahrood, Z. B. and Taleai, M. (2021). A poi group recommendation method in location-based social networks based on user influence. *Expert Systems with Applications*, 171:114593–114605.

Sun, L., Ma, J., Zhang, Y., Dong, H., and Hussain, F. K. (2016). Cloud-fuser: Fuzzy ontology and mcdm based cloud service selection. *Future Generation Computer Systems*, 57:42–55.

Thaipisutikul, T. and Chen, Y.-N. (2023). An improved deep sequential model for context-aware poi recommendation. *Multimedia Tools and Applications*, pages 1–26.

Tu, L., Zhang, F., Wang, F., and Wang, X. (2009). A random group mobility model for mobile networks. In *2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pages 551–556. IEEE.

Wang, J., Xie, H., Wang, F. L., Lee, L.-K., and Wei, M. (2023a). Jointly modeling intra-and inter-session dependencies with graph neural networks for session-based recommendations. *Information Processing & Management*, 60(2):103209.

Wang, W., Zhang, G., and Lu, J. (2016). Member contribution-based group recommender system. *Decision Support Systems*, 87:80–93.

Wang, X., Fukumoto, F., Li, J., Yu, D., and Sun, X. (2023b). Statrl: Spatial-temporal and text representation learning for poi recommendation. *Applied Intelligence*, 53(7):8286–8301.

Wang, Z., Zeng, J., Wen, J., Gao, M., and Zhou, W. (2023c). Point-of-interest recommendation using deep semantic model. *Expert Systems with Applications*, page 120727.

Wang, Z., Zhu, Y., Zhang, Q., Liu, H., Wang, C., and Liu, T. (2022). Graph-enhanced spatial-temporal network for next poi recommendation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–21.

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., and Tan, T. (2019). Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 346–353.

Wu, W.-c., Lin, S.-J., et al. (2002). Housing demand with random group effects. *International Real Estate Review*, 5(1):133–45.

Xu, Z., Zheng, X., Zhang, H., and Luo, Y. (2021). A novel travel group recommendation model based on user trust and social influence. *Mobile Information Systems*, 2021:1–10.

Xue, F., He, X., Wang, X., Xu, J., Liu, K., and Hong, R. (2019). Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems (TOIS)*, 37(3):1–25.

Ye, H., Song, Y., Li, M., and Cao, F. (2023). A new deep graph attention approach with influence and preference relationship reconstruction for rate prediction recommendation. *Information Processing & Management*, 60(5):103439–103455.

Yin, H., Wang, Q., Zheng, K., Li, Z., Yang, J., and Zhou, X. (2019). Social influence-based group representation learning for group recommendation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 566–577. IEEE.

Yu, D., Yu, T., Wang, D., and Shen, Y. (2022). Ngpr: A comprehensive personalized point-of-interest recommendation method based on heterogeneous graphs. *Multimedia Tools and Applications*, 81:39207–39228.

Yu, Z., Zhou, X., Hao, Y., and Gu, J. (2006). Tv program recommendation for multiple viewers based on user profile merging. *User Modeling and User-adapted Interaction*, 16(1):63–82.

Zan, S., Zhang, Y., Meng, X., Lv, P., and Du, Y. (2021). Uda: A user-difference attention for group recommendation. *Information Sciences*, 571:401–417.

Zhan, Z. and Xu, B. (2023). Analyzing review sentiments and product images by parallel deep nets for personalized recommendation. *Information Processing & Management*, 60(1):103166.

Zhang, J., Gao, M., Yu, J., Guo, L., Li, J., and Yin, H. (2021). Double-scale self-supervised hypergraph learning for group recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2557–2567.

Zhang, M., Wu, S., Gao, M., Jiang, X., Xu, K., and Wang, L. (2020). Personalized graph neural networks with attention mechanism for session-aware recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3946–3957.

Zhang, Z., Zhu, J., and Yue, C. (2022). Session-based graph attention poi recommendation network. *Wireless Communications and Mobile Computing*, 2022.

Zhao, X., Zhang, Z., Bi, X., and Sun, Y. (2020). A new point-of-interest group recommendation method in location-based social networks. *Neural Computing and Applications*, pages 1–12.

Zhu, Q., Wang, S., Cheng, B., Sun, Q., Yang, F., and Chang, R. N. (2018). Context-aware group recommendation for point-of-interests. *IEEE Access*, 6:12129–12144.

Zou, G., Jiang, M., Niu, S., Wu, H., Pang, S., and Gan, Y. (2018). Qos-aware web service recommendation with reinforced collaborative filtering. In *Service-Oriented Computing: 16th International Conference, ICSOC 2018, Hangzhou, China, November 12-15, 2018, Proceedings 16*, pages 430–445. Springer.