TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 14/31 pp186–197
DOI: 10.26599/TST.2023.9010148
Volume 30, Number 1, February 2025

# Heterogeneous Spatio-Temporal Graph Contrastive Learning for Point-of-Interest Recommendation

Jiawei Liu, Haihan Gao, Cheng Yang, Chuan Shi*, Tianchi Yang, Hongtao Cheng, Qianlong Xie, Xingxing Wang, and Dong Wang

**Abstract:** As one of the most crucial topics in the recommendation system field, point-of-interest (POI) recommendation aims to recommending potential interesting POIs to users. Recently, graph neural networks (GNNs) have been successfully used to model interaction and spatio-temporal information in POI recommendations, but the data sparsity of POI recommendations affects the training of GNNs. Although some existing GNN-based POI recommendation approaches try to use social relationships or user attributes to alleviate the data sparsity problem, such auxiliary information is not always available for privacy reasons. Self-supervised learning gives a new idea to alleviate the data sparsity problem, but most existing self-supervised recommendation methods cannot be directly used in the spatio-temporal graph of POI recommendations. In this paper, we propose a novel heterogeneous spatio-temporal graph contrastive learning method, HestGCL, to compensate for existing GNN-based methods' shortcomings. To model spatio-temporal information, we generate spatio-temporally specific views and design view-specific heterogeneous graph neural networks to model spatial and temporal information, respectively. To alleviate data sparsity, we propose a cross-view contrastive strategy to capture differences and correlations among views, providing more supervision signals and boosting the overall performance collaboratively. Extensive experiments on three benchmark datasets demonstrate the effectiveness of HestGCL, which significantly outperforms existing methods.

**Key words:** point-of-interest recommendation; graph neural network; self-supervised learning

## 1 Introduction

In recent years, with the rise of location-based social networks (LBSNs), many users record and share their check-in histories on websites such as Foursquare and Gowalla. Based on users' check-in data, service providers can suggest locations for users that may interest them, i.e., point-of-interest (POI) recommendation. Point-of-interest recommendations can be divided into two categories[1]: One is focused on the sequential information of user behavior modelling, called next-POI recommendation (or successive POI recommendation)[2]. The other focuses on the spatial and temporal patterns of user-POI interactions and is called general POI recommendation[3]. In this paper, we study general POI recommendation task and refer to it as POI recommendation.

Different from other recommendation tasks, the key characteristic of general POI recommendation is to take not only collaborative signals but also the spatial (i.e.,

• Jiawei Liu, Cheng Yang, Chuan Shi, Tianchi Yang, and Hongtao Cheng are with School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: liu_jiawei@bupt.edu.cn; yangcheng@bupt.edu.cn; shichuan@bupt.edu.cn; yangtianchi@bupt.edu.cn; 2019211200@bupt.cn.
• Haihan Gao, Qianlong Xie, Xingxing Wang, and Dong Wang are with Meituan, Beijing 100102, China. E-mail: gaohaihan@meituan.com; xieqianlong@meituan.com; wangxingxing04@meituan.com; wangdong07@meituan.com.
* To whom correspondence should be addressed.
Manuscript received: 2023-08-21; revised: 2023-11-14; accepted: 2023-12-06

geographical) and temporal influence into consideration. Take Fig. 1 as an example, $U2$ visited $P1$ and $P2$ in different spatio-temporal contexts (i.e., $[L2, 9:00]$, $[L3, 18:30]$), and $P1$ was visited by $U1$, $U2$, and $U3$ in different time points. Therefore, many existing works tried to capture geographical and temporal influence on POI recommendation. Earlier approaches[4, 5] design time-enhanced or geographical-aware matrix factorization (MF), which can be regarded as variants of the classical MF method. With the rise of graph embedding (GE) methods, many GE-based POI recommendation[6–8] have been proposed, which regard user and POI as nodes in a bipartite graph and adapt existing graph embeddings methods (e.g., LINE[9], node2vec[10], TransE[11]) to learn their embeddings for recommendation. Recently, with the widespread use of graph neural networks (GNN), some researchers applied GNNs[12, 13] or spatio-temporal graph neural networks (STGNNs)[14] for POI recommendation, and have achieved better results on recommendation performance.

Although current GNN-based methods show their superior performance, we argue that they are flawed in two aspects:

- **Coarse granularity for modelling heterogeneity in POI recommendation data.** Existing approaches tend to construct only a bipartite graph of users and POIs to capture their node heterogeneity and simple interaction between them. Nevertheless, due to the existence of time and space factors, there are more kind of heterogeneities and more complex relationships in POI recommendation, which is often overlooked by existing GNN-based methods[12]. To deal with the wide variety of heterogeneity, some work[14] constitutes a complex multi-layer graph to cover various heterogeneities, but the newest work[13] argues that this may reduce the recommendation performance because it mixes too much information while message passing and introduces additional noise. Therefore, how to reduce the impact of noise on GNN training while using various heterogeneous information is still a key topic that needs to be explored.

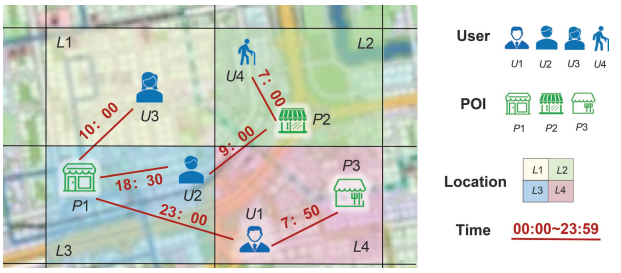



**Fig. 1 Toy example of POI recommendation data, including four types of information (i.e., user, POI, location, and time). Note that the modelling granularity of location and time can be specified, for example, by encoding the latitude and longitude of the area according to the geohash algorithm, or encoding the time by hour.**

- **Insufficient consideration of interaction sparsity issues.** Recommender systems often suffer from data sparsity, making GNN-based models challenging to learn high-quality node representations or susceptible to interaction noise. In other recommendation scenarios, some recent works[15, 16] have focused on reducing the impact of data sparsity problems on GNN models. However, for POI recommendations, this issue still receives little attention.

To address these two limitations, we propose a novel POI recommendation method, named heterogeneous spatio-temporal graph contrastive learning (HestGCL). To model the heterogeneity in POI recommendation scenarios at a finer granularity, we build a heterogeneous spatio-temporal graph including three kinds of nodes (i.e., User, POI, and Location) and three kinds of relations, which helps uncover the influence of heterogeneous information on recommendations. To address the challenges posed by data sparsity and spatio-temporal noises to GNN models, inspired by self-supervised learning[17], we propose a cross-view contrastive learning technique for spatio-temporal heterogeneous graphs. Specifically, we first split the complete heterogeneous graph into a spatial view and a temporal view. Then, we design spatial-aware and temporal-aware graph neural networks for spatial and temporal views, respectively. Finally, we use the node representations obtained from each view for contrastive learning.

Experimental results on three public datasets demonstrate that our HestGCL model achieves consistent and significant improvement over state-of-the-art baseline methods on the POI recommendation task. The relative improvements of Recall@50 are 8.83%, 14.61%, and 6.86% on Foursquare, Gowalla, and Meituan, respectively. Ablation studies and hyper-parameter experiments further demonstrate the effectiveness and robustness of our model.

The contributions of this paper are summarized as follows:

(1) Different from traditional GNN-based POI

recommendation models which are usually modelled as a user-POI bipartite graph or multi-layer graph, we model the POI recommendation task as a dual-view heterogeneous spatio-temporal graph, which allows for better exploitation of data heterogeneity and mitigate data noise.

(2) We propose a novel cross-view heterogeneous spatio-temporal graph contrastive learning model (i.e., HestGCL), which combines spatio-temporal heterogeneous graph neural networks with contrastive learning to provide self-supervision signals and mitigate data sparsity problems. To the best of our knowledge, this is the first attempt to utilize heterogeneous spatio-temporal graph contrastive learning on POI recommendation.

(3) We conduct extensive experiments on three benchmark POI recommendation datasets, and the performances outperform existing state-of-the-art baselines, demonstrating the proposed model's effectiveness.

## 2 Related Work

In this section, we review some related works of this paper, including POI recommendation and GNN-based recommendation.

### 2.1 Point-of-interest recommendation

POI recommendation can be divided into two categories: Next-POI recommendation and general POI recommendation. The former focuses on sequence information modeling, while the latter pays more attention to spatial-temporal information modeling. As for general POI recommendation, earlier approaches are improved based on matrix decomposition algorithms. For example, LRT[4] designs time-enhanced matrix factorization by modeling users' preferences at different times, and IRenMF[5] designs geographical-aware matrix factorization by modeling location-level and region-level influence. Recently, thanks to the rise of graph embedding (GE) technology, more and more GE-based POI recommendation algorithms are proposed. For example, GE[6] builds four bipartite graphs (i.e., POI-POI graph, POI-region graph, POI-Time graph, and POI-word graph), and uses a variant of LINE[9] to get the embedding of the POI node in each graph, and then makes a linear combination of them to get the final embedding. LBSN2Vec[7] builds a hypergraph including four kinds of nodes (i.e., POI, user, tag, time), and uses a variant of node2vec[10] to get users' and POIs' embeddings. Spatiotemporal context-aware and translation-based recommender framework (STA)[8] treats location-time pair as a translation from user to POI, and uses TransE[11] to learn embeddings. More recently, graph neural networks, especially spatio-temporal graph neural networks, have achieved significant performance in various graph-related tasks, and have also been used for POI recommendation. For example, graph neural network-based POI recommendation model (GPR)[12] builds user-POI graph and POI-POI graph, and uses graph convolutional network (GCN)[18] to learn embeddings. Spatial-temporal aware graph convolutional neural network (STGCN)[14] builds a multi-layer graph including three kinds of nodes (i.e., POI, user, region), and uses a method similar to relational graph convolutional networks (RGCN)[19] to consider the various information in the graph without difference. After GNN's aggregation process, MPGRec[13] designs additional memory modules to model the space consistency in POI recommendation. However, existing works mostly model basic user-item interaction as bipartite graphs, not considering data heterogeneity and the potential effect of spatio-temporal information. In this paper, we design a cross-view heterogeneous spatio-temporal graph learning model to address this challenge.

### 2.2 GNN-based recommendation

As an effective tool for modeling interactions, graphs are widely used in recommender systems. Especially with the rise of graph neural networks, GNN-based recommendation algorithms have achieved the best results in many recommendation tasks. For example, neural graph collaborative filtering (NGCF)[20] and LightGCN[21] apply graph neural network to explicitly model the high-level connection between user and item. DiffNet[22] models users' interest preference from their social relationships and historical behaviors, and uses GraphSAGE[23] to simulate how users are affected by social communication. Data sparse and interaction noise bring great challenges to the training of graph neural networks, and therefore self-supervised learning (SSL) is used for GNN-based recommendation[24]. Before that, SSL shows its potential to solve data sparsity problem on computer vision (CV)[25], natural language processing (NLP)[26], and graph-related tasks[27]. To combine SSL with GNN-based recommendation, self-supervised graph learning

(SGL)[16] uses dropout and random walk strategy to generate different views of the initial graph, and maximizes the consistency between the different views of the same node and the view of other nodes. Neighborhood-enriched contrastive learning (NCL)[17] introduces the neighbors of users (or items) from the structure and semantic space, respectively, and uses them as positive (or negative) contrastive pairs. Compared to traditional recommendation tasks, POI recommendation has complex and heterogeneous spatial-temporal information, so it brings more noise. However, existing self-supervised GNN-based recommendation methods do not take into account the spatio-temporal characteristics of the data, and there are no existing works to apply the SSL method to general POI recommendation task. In this paper, we design the first SSL-enhanced spatio-temporal graph learning model for general POI recommendation, to mitigate the impact of data sparsity and interaction noise.

## 3 Preliminary

In this section, we formally define some significant concepts related to our work.

**POI recommendation.** Let $U = \{u_1, u_2, \ldots, u_M\}$ be a set of $M$ users and $P = \{p_1, p_2, \ldots, p_N\}$ be a set of $N$ POIs. The POI recommendation task aims to recommend a list of ranked POIs from $P$ for each given user $u$ in $U$, based on their historical POI visits $P_u$, POIs' locations $L_P$, and their interaction timestamps $T_{\text{UP}}$.

**Heterogeneous spatio-temporal graph (HeSTG).** HeSTG refers to $G = (V, E, L, T)$ consisting of an object set $V$, a link set $E$, a location set $L$, and a timestamp set $T$, where $L$ and $T$ are associated with either $V$ or $E$, and there exists a node type mapping function $\phi : V \to A$ and a link type mapping function $\varphi : E \to R$, satisfying $|A| + |R| > 2$.

## 4 Proposed Model

In this section, we introduce our proposed model, heterogeneous spatio-temporal graph contrastive learning (HestGCL), a heterogeneous spatio-temporal graph contrastive learning method for POI recommendation. The overall architecture is depicted in Fig. 2. More specifically, we first generate the graph topology and features from raw data, and extract the full graph into dual-view subgraphs. Secondly, we propose dual-view heterogeneous spatio-temporal graph neural networks to generate spatial-based and temporal-based embeddings, and then fuse them to obtain spatio-temporal embeddings. Finally, we design contrastive learning modules to provide self-supervised signals for the recommendation.

### 4.1 Graph construction and data preprocessing

As for the POI recommendation task, the

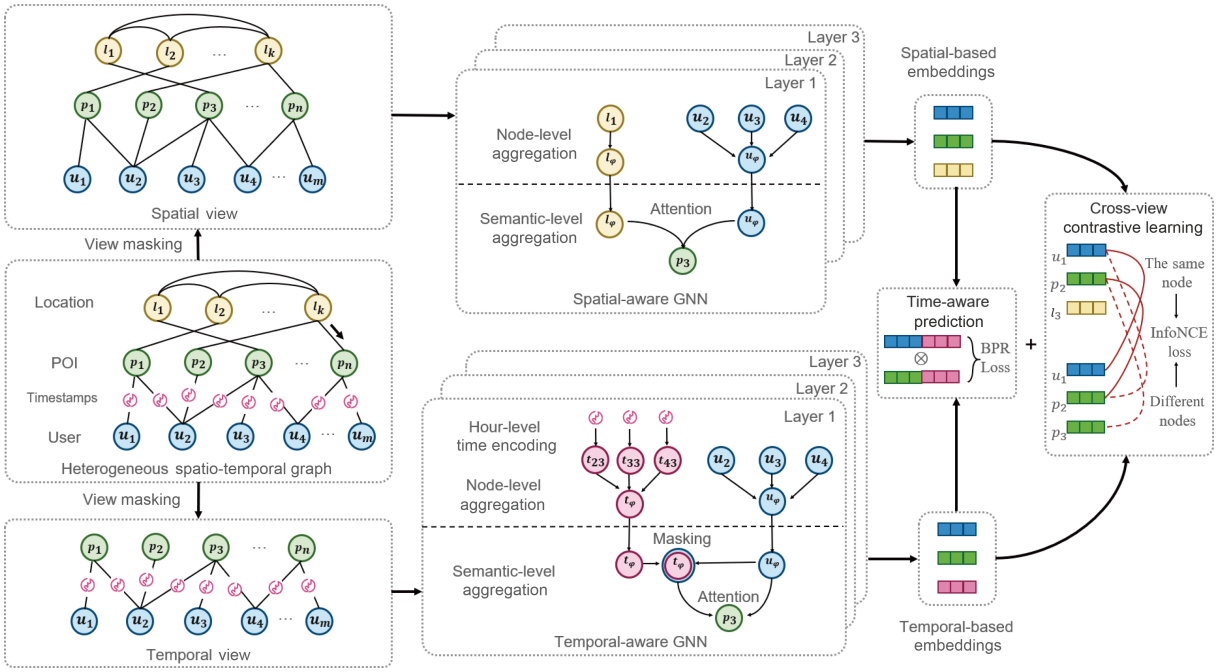



**Fig. 2 Overall architecture of HestGCL.**

corresponding HeSTG can be defined as $G=(V_P, V_U, V_L, E_{\mathrm{UP}}^T, E_{\mathrm{LP}}, E_{\mathrm{LL}})$ consisting of a POI node set $V_P$, a user node set $V_U$, a location node set $V_L$, a user-POI interaction edge set $E_{\mathrm{UP}}^T$ with check-in timestamp set $T$, a location-POI geographical mapping edge set $E_{\mathrm{LP}}$, and a location-location geographical proximity edge set $E_{\mathrm{LL}}$. Note that location nodes $V_L$ refer to the region of POI and have their IDs that can be obtained by the geohash encoding algorithm.

However, such a complex HeSTG mixes information from multiple aspects, and using GNN directly may introduce a lot of noise[13]. Besides, the POI recommendation task usually has just ID information on users (or POIs), but GNN depends on nodes' features as inputs. Therefore, in this section, we introduce our graph extraction and data preprocessing strategies.

#### 4.1.1 Dual-view graph extraction and view masking

Inspired by multi-view representation learning which can learn higher level representation in view-specific modules[28], we extract the full HeSTG $G$ into two view-specific subgraphs $G_S=(V_P, V_U, V_L, E_{\mathrm{UP}}, E_{\mathrm{LP}}, E_{\mathrm{LL}})$ and $G_T=(V_P, V_U, E_{\mathrm{UP}}^T)$, representing spatial view and temporal view separately:

- Spatial view $G_S$: Includes three kinds of nodes $V_P$, $V_U$, $V_L$ and three kinds of edges $E_{\mathrm{UP}}$, $E_{\mathrm{LP}}$, $E_{\mathrm{LL}}$, where $E_{\mathrm{UP}}$ doesn't consider the influence of time.
- Temporal view $G_T$: Includes two kinds of nodes $V_P, V_U$ and one kind of edge $E_{\mathrm{UP}}^T$ that has timestamps as edge features, not considering the influence of geography.

#### 4.1.2 Node feature initialization

Following Refs. [12, 21], we only use ID features as input without considering other information (e.g., category, social connection), and build an embedding lookup table based on the parameterized matrix. Take user nodes $V_U$ as an example, we initialize their features as $X_U=[x_{u_1},\ldots,x_{u_N}]$, where $N$ represents the number of users, and $x_u\in R^d$ is an embedding vector ($d$ denotes the embedding size). Similarly, we obtain POI and location nodes' feature matrices $X_P$ and $X_L$. All of these features are also represented as inputs $X^{(0)}$ of GNN model.

### 4.2 Dual-view heterogeneous spatio-temporal graph neural network

In this section, we introduce the backbone GNN of our method, that is, a dual-view heterogeneous spatio-temporal graph neural network, named HestGNN. In detail, we design view-specific GNNs (i.e., HestGNN-S and HestGNN-T) for spatial and temporal views.

#### 4.2.1 Spatial-aware graph neural network

For the spatial view, we design a spatial-aware graph neural network, named HestGNN-S. The spatial view contains multiple types of nodes (i.e., user, POI, location) and their relations, and each relation has a different impact on the nodes. Therefore, we design message passing functions for each type of nodes. We firstly define node updating functions of the $(k+1)$ layer for them:

$$x_u^{(k+1)}=f_{\mathrm{agg}}\left(x_u^{(k)},\{x_p^{(k)}|p\in N_u\}\right) \tag{1}$$

$$x_p^{(k+1)}=f_{\mathrm{agg}}\left(x_p^{(k)},\{x_u^{(k)}|u\in N_p\},\{x_l^{(k)}|l\in N_p\}\right) \tag{2}$$

$$x_l^{(k+1)}=f_{\mathrm{agg}}\left(x_l^{(k)},\{x_p^{(k)}|p\in N_l\},\{x_l^{(k)}|l\in \mathcal{N}_l\}\right) \tag{3}$$

where $x_u$, $x_p$, and $x_l$ are the aggregated embeddings of User $u$, POI $p$, and Location $l$, and $f_{\mathrm{agg}}(\cdot)$ is the aggregation function of them. $N_u$, $N_p$, and $N_l$ are their corresponding neighbors. Note that for user nodes, they are only connected to the POI nodes which are interacted in check-in history. For POI nodes, they are not only connected to user nodes but also location nodes which they are located in. For location nodes, they are connected to not only POI nodes but also their geographical-neighboring location nodes.

Next we define the aggregation functions $f_{\mathrm{agg}}(\cdot)$ for each type of nodes as follows:

$$f_{\mathrm{agg}}\left(x_u^{(k)},\{x_{p\in N_u}^{(k)}\}\right)=\sum_{p\in N_u}\frac{1}{|N_u|}x_p^{(k)} \tag{4}$$

$$\begin{aligned}f_{\mathrm{agg}}\left(x_p^{(k)},\{x_{u,l\in N_p}^{(k)}\}\right)=b_p\sum_{u\in N_p}\frac{1}{|N_p|}x_u^{(k)}+\\(1-b_p)\sum_{l\in N_p}\frac{1}{|N_p|}x_l^{(k)}\end{aligned} \tag{5}$$

$$\begin{aligned}f_{\mathrm{agg}}\left(x_l^{(k)},\{x_{p,l\in N_l}^{(k)}\}\right)=b_l\sum_{p\in N_l}\frac{1}{|N_l|}x_p^{(k)}+\\(1-b_l)\sum_{l\in N_l}\frac{1}{|N_l|}x_l^{(k)}\end{aligned} \tag{6}$$

where $b_p$ and $b_l$ are node-specific learnable attention coefficients. Note that for user nodes, we use the similar aggregation strategy to LightGCN[21] by averaging their POI neighbors because they have only one type of neighbors. As for POI and Location nodes, we design a dual-level aggregation strategy, that is, node-level and semantic-level aggregation. As for

node-level aggregation, we aggregate neighbors of the same relation by averaging them. As for semantic-level aggregation, we use an attention coefficient to to aggregate information of different relations, and therefore distinguish the impact of different relations and obtain the embeddings on each layer.

#### 4.2.2 Temporal-aware graph neural network

For the temporal view, we design a temporal-aware graph neural network, named HestGNN-T. The temporal view contains check-in timestamps on user-POI edges, and these timestamps help model users' interest preferences. Therefore, we consider time information by time-encoding on edges and fusing edge embeddings during aggregation.

To be more specific, we first use an hour-level time-encoding strategy to transform a timestamp $t$ to an embedding $e_t$. The encoding strategy is to build an embedding lookup table as follows:

$$e_T = [\sigma(e_{t_1}), \ldots, \sigma(e_{t_N})] \tag{7}$$

where $N$ is the number of hours (i.e., 24), $e_{t_1} \in \mathrm{R}^d$ is an embedding vector ($d$ denotes the embedding size), and $\sigma(\cdot)$ is the Sigmoid function.

Then, we use a gating strategy to aggregate time embeddings and neighbor embeddings for each node on the $(k+1)$ layer:

$$x_u^{(k+1)} = c_u(\hat{x}_u^{(k+1)} \odot \tilde{x}_u) + (1 - c_u)\hat{x}_u^{(k+1)} \tag{8}$$

$$x_p^{(k+1)} = c_p(\hat{x}_p^{(k+1)} \odot \tilde{x}_p) + (1 - c_p)\hat{x}_p^{(k+1)} \tag{9}$$

where $\odot$ is the element-wise product, $c_u$ and $c_p$ are learnable node-specific attention coefficients for aggregating time and neighbor information, respectively $\hat{x}_u^{(k+1)}$ and $\hat{x}_p^{(k+1)}$ are the $(k+1)$-th layer nighbor-based embeddings of user $u$ and POI $p$, while $\tilde{x}_u$ and $\tilde{x}_p$ are layer-independent time-based embeddings for user $u$ and POI $p$. The calculation process of $\hat{x}_u^{(k+1)}$, $\hat{x}_p^{(k+1)}$, $\tilde{x}_u$, and $\tilde{x}_p$ are defined as follows.

Take user $u$ as an example, $\hat{x}_u^{(k+1)}$ is calculated by averaging the neighbors $p \in N_u$ embeddings $x_p^{(k)}$, while the calculation process of $\tilde{x}_u$ is to average the timestamp embeddings on node pairs $(u, p)$, which is layer-independent and calculated only once:

$$\hat{x}_u^{(k+1)} = \sum_{p \in N_u} \frac{1}{|N_u|} x_p^{(k)} \tag{10}$$

$$\tilde{x}_u = \sum_{p \in N_u} \frac{1}{|N_u|} e_{t_{up}} \tag{11}$$

where $t_{up}$ is the timestamp of between node $u$ and $p$, and $e_{t_{up}}$ is the embedding of timestamp $t_{up}$. Since the roles of user and POI are reciprocal in the temporal view, we can obtain POI embeddings $\hat{x}_p^{(k+1)}$ and $\tilde{x}_p$ in the same way.

### 4.3 Prediction and optimization

In this section, we introduce our prediction and optimization strategy. Based on the embeddings obtained from HestGNN-S and HestGNN-T, we generate a recommendation list based on a time-aware prediction module, and use an extra contrastive module to enhance self-supervised signals. We name the complete model as HestGCL.

#### 4.3.1 Time-aware prediction module

We average the embeddings of each layer $k \in K$ and obtain the final embeddings $(x_u^S, x_p^S)$ and $(x_u^T, x_p^T)$ for spatial and temporal view. For example,

$$x_u^S = \sum_{k \in [0,K]} \frac{1}{|K+1|} x_p^{(k)} \tag{12}$$

Then, we use the attention mechanism to fuse $x^S$ and $x^T$ to obtain the final embeddings $x^F$.

$$x^F = a_{\text{final}} x^S + (1 - a_{\text{final}}) x^T \tag{13}$$

where $a_{\text{final}}$ is a learnable attention coefficient.

Because the user's decision is closely related to time, we consider the time encoding obtained from Eq. (7) when computing the similarity of users' embeddings $x_u^F$ and POIs' embeddings $x_p^F$, and then sort the similarity scores of POIs to generate a recommendation list.

$$y_{up} = (x_u^F \odot e_{t_{up}})^\top \cdot (x_p^F \odot e_{t_{up}}) \tag{14}$$

where $\cdot$ is the inner product, $\odot$ is the element-wise product, and $e_{t_{up}}$ is the embedding of interaction timestamp $t_{up}$ between user $u$ and POI $p$.

As for optimization, we use Bayesian personalized ranking (BPR) loss to encourage the prediction of observed pairs to be higher than unobserved pairs:

$$L_{\text{BPR}} = -\sum_{u=1}^{|U|} \sum_{i \in N_u} \sum_{j \notin N_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \tag{15}$$

#### 4.3.2 Cross-view graph contrastive module

For each node $i \in \{\mathcal{V}_\mathcal{P}, \mathcal{V}_\mathcal{U}\}$, we have generated its spatial-view embeddings $\boldsymbol{x}_{\boldsymbol{S}(i)}$ and temporal-view embeddings $\boldsymbol{x}_{\boldsymbol{T}(i)}$. To better use the information of two views, we propose a cross-view contrastive strategy, which regards the corresponding embeddings $\boldsymbol{x}_{\boldsymbol{S}(i)}$ and $\boldsymbol{x}_{\boldsymbol{T}(i)}$ as positive pairs, while others as negative pairs.

Formally, we also adopt InfoNCE[29] loss:

$$L_{CL} = \sum_{i \in V} \sum_{j \neq i} -\log(f(x_{S(i)}, x_{T(i)}) - \log(1 - f(x_{(i)}, x_{(j)})) \tag{16}$$

where $f(\cdot)$ measures the cosine similarity between two embeddings, and $x_{(i)}$ can be spatial-view embeddings $x_{S(i)}$ or temporal-view embeddings $x_{T(i)}$.

#### 4.3.3 Overall

We unify the prediction module and cross-view contrastive learning module into a primary&auxiliary learning framework, where the prediction module is a primary task and the contrastive learning module is an auxiliary task. Formally, the overall optimization function is

$$L = L_{\text{BPR}} + \beta * L_{CL} + \gamma * \|\Theta\|_2^2 \tag{17}$$

where $\beta$ and $\gamma$ are trade-off parameters to balance three parts, and $\Theta$ is the set of embeddings obtained from the model.

## 5 Experiment

In this section, we conduct experiments on three real-world datasets, which aims to answer the following research questions (RQs):

- RQ1: How does the proposed HestGCL method perform compared to the start-of-the-art baselines in POI recommendation?
- RQ2: How does each component of HestGCL contribute to the overall performance?
- RQ3: How do different hyper-parameters affect the performance of HestGCL?

### 5.1 Experimental setup

#### 5.1.1 Datasets

To evaluate the performance of the proposed method, we employ the following three real-world LBSN datasets. Foursquare[30] and Gowalla[30] are two benchmark POI recommendation datasets, where we use the geohash algorithm to generate location ID and location neighbors. Meituan† is a recent take-out delivery dataset, consisting of users' order histories obtained from Meituan APP, including orders' location ID and POIs' location ID. In our setting, we ignore orders' region ID to keep information consistent with other datasets. The statistical information of these datasets is summarized in Table 1.

For all datasets, we split them into training/validation/test sets in chronological order, where the oldest 70% check-ins are used as the training set, more recent 20% check-ins are used as the validation set, and the most recent 10% check-ins are used as the test set.

#### 5.1.2 Baselines

To validate the effectiveness of our proposed model, we compare it with five kinds of baselines, as illustrated in Table 2: Non-GNN general recommendation models (i.e., BPRMF[31] and NeuMF[32]); Non-GNN POI recommendation models (i.e., LGLMF[33] and STACP[34]); GNN-based general recommendation models (i.e., NGCF[20] and LightGCN[21]); GNN-based state-of-the-art POI recommendation model (i.e., GPR[12]); GNN+CL-based recommendation models (i.e., SGL[16] and NCL[17]). For all these baselines, we set the optimal hyper-parameter settings reported in their papers.

Note that in our paper, we focus on the spatial-temporal information under data sparsity scenario in POI recommendation, and thus we do not choose those sequence models[35–37] for next-POI recommendation task as baselines.

#### 5.1.3 Implementation detail

We implement our framework based on Recbole[38],

**Table 1 Dataset statistics.**

| Dataset | #check-ins | #POI | #User | Sparsity | Time span |
|---|---|---|---|---|---|
| Foursquare | 1 196 248 | 28 593 | 24 941 | 99.900% | Apr.2012–Sep.2013 |
| Gowalla | 1 278 274 | 32 510 | 18 737 | 99.865% | Feb.2009–Oct.2010 |
| Meituan | 602 331 | 3182 | 38 904 | 99.51% | Mar.1st 2021–Mar.28th 2021 |

**Table 2 Category of baselines and HestGCL.**

| Architecture | General item recommendation | General POI recommendation |
|---|---|---|
| Non-GNN | BPRMF[31], NeuMF[32] | LGLMF[33], STACP[34] |
| GNN | NGCF[20], LightGCN[21] | GPR[12], STGCN[14], MPGRec[13] |
| GNN+SSL | SGL[16], NCL[17] | HestGCL (Ours) |

†https://www.biendata.xyz/competition/smp2021_1/

and employ the Adam[39] optimizer to minimize the overall loss $\mathcal{L}$, where the learning rate is set as 0.05. Same as baselines, we employ an early stopping strategy with a patience of 10, i.e., we will stop training if the Recall@5 metric does not increase for 10 epochs. For the optimization function, we set both $\beta$ and $\gamma$ as 0.000 05. For hyperparameter tuning, we conduct a heuristic search by exploring the number of layer $K \in \{1,2,3,4,5\}$ and embedding size in HestGNN $d_{\text{GNN}} \in \{64,128,256\}$.

#### 5.1.4 Evaluation metrics

To validate the performance of the recommendation models, we use the following two widely-used evaluation metrics: Recall@$N$ and MAP@$N$, where $N$ is the number of POIs in the ranked list. For a more comprehensive comparison of results, we report the results for $N$ = 5, 10, 20, and 50.

### 5.2 Performance comparision (RQ1)

We present the results of top-$N$ recommendation on three datasets in Table 3. Based on the experimental results, we can observe that:

(1) HestGCL outperforms all baseline methods across three datasets in terms of all metrics. More specifically, it achieves relative improvements than the strongest baselines on Recall@50 by 8.83%, 14.61%, and 6.86% on Foursquare, Gowalla, and Meituan datasets, respectively. This demonstrates the validity of our model.

(2) For both Foursquare and Gowalla datasets, GNN-based methods tend to work better than NeuMF. This indicates that the GNN-based methods can propagate higher-order information and thus learn better node representations. For the Meituan dataset, NeuMF can also achieve similar performance as the GNN-based baselines. This may be since the Meituan dataset is the least sparse and has a much fewer number of POIs, which makes it less necessary to aggregate higher-order information for GNNs.

(3) Compared with traditional GNN models (e.g., LightGCN, NGCF), general self-supervised GNN models (e.g., SGL, NCL) can enhance the models in some cases, but they are unstable. For example, the Recall score of SGL (Recall@5 = 0.3373) is worse than LightGCN (Recall@5 = 0.3456) on Meituan Dataset. However, our model can achieve better results, probably because it is designed explicitly for spatio-temporal data and therefore provides a more efficient self-supervised signal.

### 5.3 Ablation study (RQ2)

To investigate the effect of each module on the overall effect of the model, we designed three model variants: HestGNN-S, HestGNN-T, and HestGNN, where HestGNN-S excludes temporal-view GNN module and CL module, HestGNN-T excludes spatial-view GNN module and CL module, and HestGNN excludes only CL module.

From the results of the ablation study in Fig. 3, we obtained the following findings:

(1) The complete HestGCL model consistently achieves the best performance, which shows that each component of HestGCL contributes to performance.

(2) Compared to HestGNN-S and HestGNN-T, HestGNN does not improve performance, while HestGCL does. This indicates that the contrastive learning module mitigates the impact of data sparsity on GNN training, and therefore enhances the model performance collaboratively.

(3) For the Foursquare and Gowalla datasets, HestGNN-T outperforms HestGNN-S; while for the Meituan dataset, HestGNN-S outperforms HestGNN-T. This indicates that temporal modelling is more important for the Foursquare and Gowalla datasets, while spatial modelling is more important for the Meituan dataset.

### 5.4 Influence of hyper-parameters (RQ3)

Our model contains two key hyperparameters (i.e., layer number $L$ and embedding size $d_{\text{GNN}}$). In this section, we will explore the effects of these two hyperparameters on the model.

#### 5.4.1 Effect of layer numbers of HestGNN

To analyze the influence of layer number $L$, we vary it from 1 to 5 and illustrate the performance comparisons on three datasets. As shown in Fig. 4a, we make the following observations:

(1) The performance on Foursquare and Gowalla is not sensitive to the number of layers. In the range of 1 to 5, the model performance is slightly enhanced as the number of layers increases. This indicates that aggregating higher-order information is helpful for both datasets.

(2) The Meituan dataset is relatively sensitive to the number of layers. The model works best when $L = 2$. This indicates that neighbors within 2-hops are most helpful for the Meituan dataset, while higher-order information introduces noise. This is consistent with

**Table 3 Overall performance and the corresponding ranking (in parentheses) of HestGCL and baselines on Foursquare (FS.), Gowalla (GW.), and Meituan (MT.) datasets.**

| Dataset | Model | Recall@5 | Recall@10 | Recall@20 | Recall@50 | MAP@5 | MAP@10 | MAP@20 | MAP@50 | Avg. Rank |
|---|---|---|---|---|---|---|---|---|---|---|
| FS. | BPRMF | 0.0359 (8) | 0.0604 (8) | 0.0976 (8) | 0.1706 (7) | 0.0220 (8) | 0.0226 (8) | 0.0256 (8) | 0.0294 (8) | 7.875 |
| | NeuMF | 0.0368 (7) | 0.0610 (7) | 0.0981 (6) | 0.1727 (6) | 0.0235 (7) | 0.0241 (7) | 0.0271 (7) | 0.0310 (7) | 6.750 |
| | LGLMF | 0.0284 (11) | 0.0459 (11) | 0.0729 (11) | 0.1284 (9) | 0.0192 (9) | 0.0190 (11) | 0.0212 (11) | 0.0242 (10) | 10.375 |
| | STACP | 0.0274 (12) | 0.0450 (12) | 0.0700 (12) | 0.1275 (10) | 0.0187 (10) | 0.0186 (12) | 0.0206 (12) | 0.0235 (11) | 11.375 |
| | NGCF | 0.0390 (6) | 0.0627 (6) | 0.0980 (7) | 0.1688 (8) | 0.0249 (6) | 0.0250 (6) | 0.0278 (6) | 0.0314 (6) | 6.375 |
| | LightGCN | 0.0469 (3) | 0.0721 (4) | 0.1076 (5) | 0.1796 (5) | 0.0317 (3) | 0.0312 (3) | 0.0341 (3) | 0.0378 (3) | 3.625 |
| | GPR | 0.0316 (9) | 0.0502 (9) | 0.0763 (9) | 0.1272 (11) | 0.0183 (11) | 0.0205 (9) | 0.0224 (9) | 0.0243 (9) | 9.500 |
| | STGCN | 0.0309 (10) | 0.0487 (10) | 0.0758 (10) | 0.1225 (12) | 0.0173 (12) | 0.0195 (10) | 0.0215 (10) | 0.0231 (12) | 10.750 |
| | MPGRec | 0.0592 (1) | 0.0848 (1) | 0.1200 (2) | 0.1915 (2) | 0.0366 (1) | 0.0398 (1) | 0.0425 (1) | 0.0452 (1) | 1.250 |
| | SGL | 0.0452 (5) | 0.0707 (5) | 0.1080 (4) | 0.1852 (3) | 0.0300 (5) | 0.0299 (5) | 0.0330 (5) | 0.0371 (5) | 4.625 |
| | NCL | 0.0463 (4) | 0.0723 (3) | 0.1083 (3) | 0.1839 (4) | 0.0313 (4) | 0.0310 (4) | 0.0338 (4) | 0.0378 (3) | 3.625 |
| | **HestGCL** | **0.0571 (2)** | **0.0847 (2)** | **0.1254 (1)** | **0.2084 (1)** | **0.0334 (2)** | **0.0371 (2)** | **0.0399 (2)** | **0.0425 (2)** | **1.750** |
| GW. | BPRMF | 0.0303 (7) | 0.0494 (8) | 0.0806 (8) | 0.1442 (8) | 0.0229 (7) | 0.0210 (8) | 0.0229 (8) | 0.0263 (8) | 7.750 |
| | NeuMF | 0.0302 (8) | 0.0497 (7) | 0.0808 (7) | 0.1478 (6) | 0.0227 (8) | 0.0211 (7) | 0.0231 (7) | 0.0267 (7) | 7.125 |
| | LGLMF | 0.0241 (11) | 0.0398 (11) | 0.0646 (11) | 0.1156 (11) | 0.0209 (9) | 0.0188 (11) | 0.0201 (11) | 0.0230 (11) | 10.750 |
| | STACP | 0.0176 (12) | 0.0302 (12) | 0.0509 (12) | 0.0964 (12) | 0.0142 (12) | 0.0131 (12) | 0.0143 (12) | 0.0168 (12) | 12.000 |
| | NGCF | 0.0308 (6) | 0.0500 (6) | 0.0810 (6) | 0.1458 (7) | 0.0235 (6) | 0.0216 (6) | 0.0234 (6) | 0.0268 (6) | 6.125 |
| | LightGCN | 0.0352 (3) | 0.0564 (3) | 0.0897 (5) | 0.1593 (5) | 0.0271 (3) | 0.0247 (3) | 0.0267 (3) | 0.0305 (3) | 3.500 |
| | GPR | 0.0302 (8) | 0.0483 (9) | 0.0766 (9) | 0.1310 (9) | 0.0183 (11) | 0.0196 (10) | 0.0216 (9) | 0.0238 (9) | 9.250 |
| | STGCN | 0.0294 (10) | 0.0454 (10) | 0.0691 (10) | 0.1181 (10) | 0.0187 (10) | 0.0199 (9) | 0.0215 (10) | 0.0234 (10) | 9.875 |
| | MPGRec | 0.0471 (2) | 0.0706 (2) | 0.1050 (2) | 0.1718 (2) | 0.0308 (1) | 0.0325 (1) | 0.0349 (1) | 0.0377 (1) | 1.500 |
| | SGL | 0.0338 (5) | 0.0557 (5) | 0.0911 (3) | 0.1657 (3) | 0.0259 (5) | 0.0240 (5) | 0.0262 (5) | 0.0305 (3) | 4.250 |
| | NCL | 0.0344 (4) | 0.0561 (4) | 0.0902 (4) | 0.1631 (4) | 0.0267 (4) | 0.0244 (4) | 0.0264 (4) | 0.0305 (3) | 3.875 |
| | **HestGCL** | **0.0496 (1)** | **0.0770 (1)** | **0.1171 (1)** | **0.1969 (1)** | **0.0283 (2)** | **0.0320 (2)** | **0.0348 (2)** | **0.0375 (2)** | **1.500** |
| MT. | BPRMF | 0.3185 (8) | 0.3901 (8) | 0.4468 (9) | 0.5195 (5) | 0.2163 (7) | 0.2310 (7) | 0.2381 (7) | 0.2425 (7) | 7.250 |
| | NeuMF | 0.3540 (3) | 0.4315 (3) | 0.4631 (6) | 0.5063 (9) | 0.2339 (4) | 0.2517 (4) | 0.2562 (4) | 0.2588 (4) | 4.625 |
| | LGLMF | 0.0005 (12) | 0.0010 (12) | 0.0030 (12) | 0.0087 (12) | 0.0002 (12) | 0.0002 (12) | 0.0004 (12) | 0.0005 (12) | 12.000 |
| | STACP | 0.0054 (11) | 0.0094 (11) | 0.0203 (11) | 0.0421 (11) | 0.0022 (11) | 0.0026 (11) | 0.0034 (11) | 0.0041 (11) | 11.000 |
| | NGCF | 0.3198 (7) | 0.3952 (7) | 0.4499 (8) | 0.5177 (7) | 0.2123 (8) | 0.2280 (8) | 0.2349 (8) | 0.2392 (8) | 7.625 |
| | LightGCN | 0.3456 (5) | 0.4221 (5) | 0.4705 (2) | 0.5315 (3) | 0.2250 (5) | 0.2417 (5) | 0.2482 (5) | 0.2520 (5) | 4.375 |
| | GPR | 0.2984 (9) | 0.3758 (9) | 0.4573 (7) | 0.5669 (2) | 0.2066 (9) | 0.2185 (9) | 0.2253 (9) | 0.2297 (9) | 7.875 |
| | STGCN | 0.1932 (10) | 0.2723 (10) | 0.3570 (10) | 0.4668 (10) | 0.0962 (10) | 0.1083 (10) | 0.1155 (10) | 0.1201 (10) | 10.000 |
| | MPGRec | 0.3930 (2) | 0.4316 (2) | 0.4655 (5) | 0.5147 (8) | 0.3170 (2) | 0.3240 (2) | 0.3272 (2) | 0.3294 (2) | 3.125 |
| | SGL | 0.3373 (6) | 0.4123 (6) | 0.4674 (3) | 0.5311 (4) | 0.2432 (3) | 0.2591 (3) | 0.2664 (3) | 0.2705 (3) | 3.875 |
| | NCL | 0.3466 (4) | 0.4236 (4) | 0.4667 (4) | 0.5187 (6) | 0.2222 (6) | 0.2391 (6) | 0.2451 (6) | 0.2484 (6) | 5.250 |
| | **HestGCL** | **0.4514 (1)** | **0.4921 (1)** | **0.5358 (1)** | **0.6058 (1)** | **0.3741 (1)** | **0.3799 (1)** | **0.3831 (1)** | **0.3854 (1)** | **1.000** |

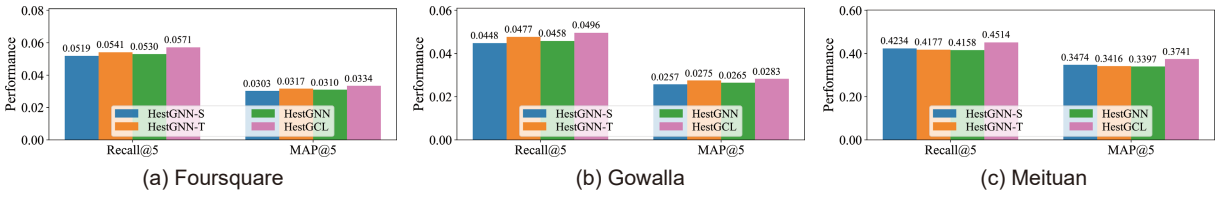



**Fig. 3 Performance (Recall@5 and MAP@5) of ablation study on the proposed HestGCL model.**

the observation of the performance of NeuMF in the main experiment.

### 5.4.2 Effect of embedding size of HestGNN

To explore the impact of embedding size $d_{GNN}$, we

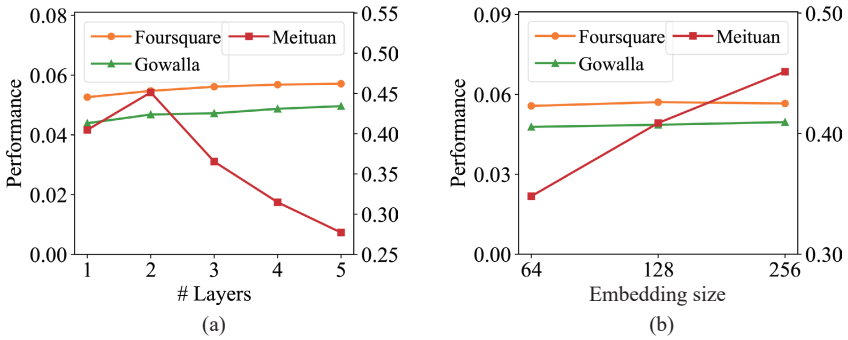



**Fig. 4 Performance (Recall@5) of different hyperparameter settings on HestGCL.**

experiment HestGCL under different embedding sizes (i.e., 64,128, 256) and record experimental results.

As illustrated in Fig. 4b, we find a difference between the Meituan dataset and other datasets. The Meituan dataset performs better with a larger embedding size, while other datasets are not much affected by embedding size. This may be because the sparsity of the Meituan dataset is relatively low, so a larger embedding dimension is needed to store more information.

## 6 Conclusion and Future Work

In this paper, we propose a novel model HestGCL for POI recommendation, which goes beyond the limitations of existing methods and explores a new strategy based on contrastive learning.

We build a heterogeneous spatio-temporal graph and design a cross-view heterogeneous spatio-temporal graph contrastive learning framework to capture heterogeneity information. The proposed heterogeneous spatio-temporal graph neural network can aggregate information from both spatial and temporal perspectives, and the contrastive learning modules can extract the information shared by views, thus alleviating the data sparsity problem. Experimental results on three benchmark datasets show the advantages of our HestGCL model over all baseline models.

For future work, we will investigate other graph augmentation techniques or design more complex contrastive tasks for POI recommendation task. We can also combine different locations and time definitions to make multi-grain modelling of users' behaviours.

## References

[1] S. L. Zhao, I. King, and M. R. Lyu, A survey of point-of-interest recommendation in location-based social networks, arXiv preprint arXiv: 1607.00647, 2016.

[2] C. Cheng, H. Yang, M. R. Lyu, and I. King, Where you like to go next: Successive point-of-interest recommendation, in *Proc. Twenty-Third Int. Joint Conf. Artificial Intelligence*, Beijing, China, 2013, pp. 2605–2611.

[3] M. Ye, P. Yin, and W. C. Lee, Location recommendation for location-based social networks, in *Proc. 18th SIGSPATIAL Int. Conf. Advances in Geographic Information Systems*, San Jose, CA, USA, 2010, pp. 458–461.

[4] H. Gao, J. Tang, X. Hu, and H. Liu, Exploring temporal effects for location recommendation on location-based social networks, in *Proc. 7th ACM Conf. Recommender systems*, Hong Kong, China, 2013, pp. 93–100.

[5] Y. Liu, W. Wei, A. Sun, and C. Miao, Exploiting geographical neighborhood characteristics for location recommendation, in *Proc. 23rd ACM Int. Conf. Conf. Information and Knowledge Management*, Shanghai, China, 2014, pp. 739–748.

[6] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, Learning graph-based POI embedding for location-based recommendation, in *Proc. 25th ACM Int. on Conf. Information and Knowledge Management*, Indianapolis, IN, USA, 2016, pp. 15–24.

[7] D. Yang, B. Qu, J. Yang, and P. Cudre-Mauroux, Revisiting user mobility and social relationships in LBSNs: A hypergraph embedding approach, in *Proc. World Wide Web Conf.*, San Francisco, CA, USA, 2019.

[8] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin, Spatiotemporal representation learning for translation-based POI recommendation, *ACM Trans. Inf. Syst.*, vol. 37, no. 2, p. 18.

[9] J. Tang, M. Qu, M. Z. Wang, M. Zhang, J. Yan, and Q. Z. Mei, Line: Large-scale information network embedding, in *Proc. of WWW*, Florence, Italy, 2015.

[10] A. Grover and J. Leskovec, node2vec: Scalable feature learning for networks, *KDD*, vol. 2016, pp. 855–864, 2016.

[11] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, Translating embeddings for modeling multi-relational data, in *Proc. 26th Int. Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, 2013.

[12] B. Chang, G. Jang, S. Kim, and J. Kang, Learning graph-

based geographical latent representation for point-of-interest recommendation, in *Proc. 29th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2020.

[13] T. Yang, H. Gao, C. Yang, C. Shi, Q. Xie, X. Wang, and D. Wang, Memory-enhanced period-aware graph neural network for general POI recommendation, in *Proc. Int. Conf. on Database Systems for Advanced Applications*, Kyoto, Japan 2023, pp. 462–472.

[14] H. Han, M. Zhang, M. Hou, F. Zhang, Z. Wang, E. Chen, H. Wang, J. Ma, and Q. Liu, STGCN: A spatial-temporal aware graph learning method for POI recommendation, in *Proc. IEEE Int. Conf. Data Mining* (*ICDM*), Sorrento, Italy, 2020.

[15] X. Xia, H. Z. Yin, J. L. Yu, Q. Y. Wang, L. Z. Cui, and X. L. Zhang, Self-supervised hypergraph convolutional networks for session-based recommendation, https://doi.org/10.1609/aaai.v35i5.16578.

[16] J. C. Wu, X. Wang, F. L. Feng, X. N. He, L. Chen, J. X. Lian, and X. Xie, Self-supervised graph learning for recommendation, in *Proc. 44th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, New York, NY, USA, pp. 726–735, 2021.

[17] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, Improving graph collaborative filtering with neighborhood-enriched contrastive learning, in *Proc. ACM Web Conf. 2022*, Lyon, France, 2022, pp. 2320–2329.

[18] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, *Neural Processing Letters*, vol. 54, pp. 2645–2656, 2022.

[19] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, Modeling relational data with graph convolutional networks, in *Lecture Notes in Computer Science*, A. Gangemi, R. Navigli, M. E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, M. Alam, eds. Cham, Switzerland: Springer, 2018, pp. 593–607.

[20] X. Wang, X. N. He, M. Wang, F. L. Feng, and T. S. Chua, Neural graph collaborative filtering, in *Proc. 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Paris, France, 2019.

[21] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, LightGCN: Simplifying and powering graph convolution network for recommendation, in *Proc. 43rd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Virtual Event, 2020.

[22] L. Wu, P. J. Sun, Y. J. Fu, R. C. Hong, X. T. Wang, and M. Wang, A neural influence diffusion model for social recommendation, in *Proc. 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Paris, France, 2019.

[23] W. Hamilton, Z. T. Ying, and J. Leskovec, Inductive representation learning on large graphs, in *Proc. Thirty-first Annual Conf. on Neural Information Processing Systems* (*NIPS*), Long Beach, CA, USA, 2017.

[24] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, Self-supervised learning for recommender systems: A survey, *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 335–355, 2024.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, Momentum contrast for unsupervised visual representation learning, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Seattle, WA, USA, 2020.

[26] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805, 2018.

[27] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, Deep Graph Infomax, arXiv preprint arXiv: 1809.10341, 2019.

[28] Y. Li, M. Yang, and Z. Zhang, A survey of multi-view representation learning, *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1863–1883, 2019.

[29] A. van den Oord, Y. Li, and O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv: 1807.03748, 2018.

[30] Y. Liu, T. A. N. Pham, G. Cong, and Q. Yuan, An experimental evaluation of point-of-interest recommendation in location-based social networks, *Proc. VLDB Endow.*, vol. 10, no. 10, pp. 1010–1021, 2017.

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

[32] X. N. He, L. Z. Liao, H. W. Zhang, L. Q. Nie, X. Hu, and T. S. Chua, Neural collaborative filtering, in *Proc. 26th Int. Conf. on World Wide Web*, pp. 173–182, 2017.

[33] H. A. Rahmani, M. Aliannejadi, S. Ahmadian, M. Baratchi, M. Afsharchi, and F. Crestani, LGLMF: Local geographical based logistic matrix factorization model for POI recommendation, arXiv preprint arXiv:1909.06667, 2019.

[34] H. A. Rahmani, M. Aliannejadi, M. Baratchi, and F. Crestani, Joint geographical and temporal modeling based on matrix factorization for point-of-interest recommendation, in *European Conference on Information Retrieval*, J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, F. Martins, eds. Cham, Switzerland: Springer, 2020, pp. 205–219.

[35] N. Lim, B. Hooi, S. K. Ng, X. Wang, Y. L. Goh, R. Weng, and J. Varadarajan, STP-UDGAT: Spatial-temporal-preference user dimensional graph attention network for next POI recommendation, in *Proc. 29th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2020.

[36] Y. Li, T. Chen, Y. Luo, H. Yin, and Z. Huang, Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation, in *Proc. Thirtieth Int. Joint Conf. Artificial Intelligence*, Montreal, Canada, 2021.

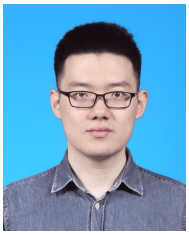
[37] X. Rao, L. Chen, Y. Liu, S. Shang, B. Yao, and P. Han, Graph-flashback network for next location recommendation, in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 1463–1471.

[38] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian et al., RecBole: Towards a unified, comprehensive and efficient framework for

recommendation algorithms, in *Proc. 30th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2021.

[39] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, in *Proc. of 3rd Int. Conf. on Learning Representations*, San Diego, CA, USA, 2015.

**Jiawei Liu** received the BS degree in computer science and technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2020. He is currently pursuing the PhD degree in computer science and technology from Beijing University of Posts and Telecommunications, Beijing, China. His research interests include graph data mining and machine learning.



**Cheng Yang** received the PhD degree in computer science and technology from Tsinghua University in 2019. He is currently an associate professor at Beijing University of Posts and Telecommunications. His research interests include social computing and network representation learning.

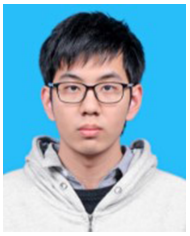
**Tianchi Yang** received the PhD degree in computer science and technology from Beijing University of Posts and Telecommunications, China in 2022. He is currently an engineer at Windows and Web Experiences (WWE) team from Microsoft Corp. His research interests include natural language processing and recommendation system.

**Qianlong Xie** received the MS degree from Beijing University of Posts and Telecommunications in Beijing, China in 2013. He is currently employed as a machine learning scientist at Meituan. He has published several works at prestigious conferences such as ICML, WWW, and DASFAA. His primary research interests include image generation, machine learning, and data mining.

**Dong Wang** received the PhD degree from Tsinghua University, Beijing, China, in 2008. Currently, he works at Meituan as a scientist. His main research interests include video recommender systems, machine learning, and data mining.

**Haihan Gao** received the MS degree from Carneige Mellon University, Pittsburgh, PA, USA, in 2016. He is currently employed as a machine learning scientist at Meituan. His primary research interests include recommender systems, machine learning, and data mining.

**Chuan Shi** received the PhD degree in computer software and theory from Chinese Academy of Sciences in 2007. He is currently a professor at Beijing University of Posts and Telecommunications. His research interests include data mining, machine learning, artificial intelligence, and big data analysis.



**Hongtao Cheng** received the BS degree in computer science and technology from Beijing University of Posts and Telecommunications in 2023. He is currently pursuing the MS degree in computer science and technology at Beijing University of Posts and Telecommunications. His research interests include machine learning and data mining.

**Xingxing Wang** earned the MS degree from Chinese Academy of Sciences in Beijing, China in 2013. He is currently employed as a machine learning scientist at Meituan. He has published several works at prestigious conferences such as ICML, WWW, AAAI, CIKM, and DASFAA. His primary research interests include recommender systems, machine learning, and data mining.