Computing and Informatics, Vol. 43, 2024, 1516–1538, doi: 10.31577/cai_2024_6_1516

# TRAVEL INTEREST POINT RECOMMENDATION ALGORITHM BASED ON COLLABORATIVE FILTERING AND GRAPH CONVOLUTIONAL NEURAL NETWORKS

Lan Pan, Jiayin Wei*, Youjun Lu, Fujian Feng

*College of Data Science and Information Engineering*
*Guizhou Minzu University*
*Guiyang 550025, China*
*e-mail:* panlan526@163.com, weijiayin05@sina.com, yjlu111@126.com, fujian_feng@gzmu.edu.cn

**Abstract.** Tourist attraction recommendation algorithms have been developed to meet demand related to tourism, spiritual and cultural pursuits. While many studies have been conducted on such algorithms, problems remain regarding tourist interest point recommendation such as ignoring social information, underutilizing context information, and not capturing node relationships which have limited the recommendation performance and representation capability. This paper proposes an algorithm based on graph convolutional neural networks and collaborative filtering (GCNs-CF) for travel interest point recommendation, using an image denoising encoder (IDE) instead of domain aggregation, to better capture the relationships and features between users and adjacent nodes of travel interest point nodes. An adaptive adjustment of the negative sample gradient size is used to solve the problem of slow convergence of graph convolutional neural network. The experimental results show that the proposed method has a higher recommendation effect than other algorithms.

**Keywords:** Graph convolutional neural network, image denoising encoder, collaborative filtering, domain aggregation, recommendation algorithm

* Corresponding author

## 1 INTRODUCTION

A travel interest point recommendation algorithm is based on calculation and analysis of users interest and historical behavior data to recommend tourist attractions or activities that meet the users' interest and needs. The development of tourism and the increasing demand for travel has made it important to provide users with personalized and accurate travel recommendation services. Travel point-of-interest recommendation algorithms can help users quickly find tourist attractions or activities that meet their needs, improve their travel experience and satisfaction, and promote the development of tourism. However, there are some problems with the current tourism point-of-interest recommendation algorithm. The first is the quality and quantity of data. The algorithm requires a large amount of data about users' historical behavior and tourist attractions or activities, but the quality and quantity of these data can affect the accuracy and effectiveness of the algorithm. Secondly, the algorithm needs to take into account the personalized needs and preferences of the users, but also ensure the diversity of the recommendation results to avoid overly single and repetitive recommendations. Finally, the algorithm must be able to respond to users' needs and changes in a timely manner and also be able to explain the reasons and rationale for the recommendation results, to enhance the users' trust in the algorithm and the experience of using it.

Some research has been carried out on the issue of POI recommendations. Rahmani et al. [1] proposed a recommendation algorithm that integrates user preferences through logical matrix decomposition considering geographical information about users and locations. Zhao et al. [2] proposed a novel unified neural network framework, named NeuNext, which leverages POI context prediction to assist next POI recommendation by joint learning. Hongfei et al. [3] mitigated the problem of sparse travel data by incorporating social, geographical and temporal information into a matrix decomposition (MF) approach. Xing et al. [4] proposed a content-aware interest point recommendation algorithm based on convolutional neural networks (CNNs). Zhai and Li [5] proposed a POI recommendation based graph convolutional neural network (PBGCN) model, which used check-in information, popularity characteristics of interest points, and users' social behaviors to recommend interest points through graph convolutional neural networks (GCNs). Chen et al. [6] proposed the formulation of user feature-level preferences via neural network hypergraphs and well-designed information propagation paths for more effective diffusion collaboration. Cai et al. [7] proposed a friend-aware graph collaborative filtering (FG-CF) model from check-in data and social links in which the association matrix of users with points of interest is estimated.

In recent years, with the introduction of GCNs and their widespread use in various fields, researchers have proposed some collaborative filtering methods based on GCNs to improve the performance of recommendation systems. The core idea of GCNs-based collaborative filtering is information transfer, in which each node aggregates neighborhood embeddings to update its own embedding; by overlaying multiple convolutional layers, a higher-order connectivity representation of users and items can be extracted. Although collaborative filtering methods based on GCNs can effectively learn representations from graph structures, some problems remain:

1. The importance of social information in POI recommendation is ignored, resulting in insufficient feature extraction for users and POIs, which limits the representation capability;
2. Failure to make full use of the contextual information embedded in the user and POI, which limits the recommendation performance;
3. Failure to accurately capture the relationships and features between nodes and neighboring nodes.

User cold-start, data sparsity and accuracy in the recall phase are the problems that need to be addressed in personalized recommendation algorithms for travel interest points. To address these problems, this paper proposes a travel interest point recommendation algorithm based on a GCN collaborative filtering method, which considers the dynamic representation of users and travel interest points, uses domain aggregators instead of traditional GCNs-CF domain aggregation to extract important graph features of user-travel interest points, and selectively embeds higher-order connectivity into the node representation. At the same time, it adopts an adaptive adjustment of the problem of slow convergence of GCNs. In collaborative filtering personalized travel recommendation algorithms are solved by using the method of adaptive adjustment of negative sample gradient size. Finally, by conducting corresponding comparison experiments, the experimental results show that the proposed method in this paper has its specific feasibility and effectiveness.

Our main contributions can be summarized as follows:

1. A travel recommendation algorithm based on graph neural networks and collaborative filtering is proposed, using an image denoising encoder instead of GCNs-CF domain aggregation to better capture the relationships and features between users and neighbouring nodes of the travel point-of-interest nodes.
2. This paper uses adaptive adjustment of negative sample gradient to solve the problem of slow convergence of the personalised travel recommendation algorithm.
3. Comparison experiments are conducted on two publicly available datasets and self-acquired datasets, and the experimental results are significantly better than the baseline model algorithm.

## 2 RELATED WORK

### 2.1 Collaborative Filtering of Travel POI Recommendations

Collaborative filtering (CF) is a common technique used in recommender systems to estimate user preferences for points of interest. In the area of travel interest point recommendation, there are two methods commonly used memory-based collaborative filtering and model-based collaborative filtering.

**Memory-based collaborative filtering:** Based on users' historical behavior data, users are matched with other similar users to recommend travel interest points that they may be interested in. Zhang et al. [8] incorporated both memory-based preferences and the influence of interest point stickiness into a user-based collaborative filtering framework to improve the performance of interest point recommendations. Febre et al. [9] presented an enhanced user-based collaborative filtering approach with demographic information for recommending touristic sites, which provides precise recommendations. Li and Gong [10] proposed a novel deep neural network, named ST-TransRec, for crossing-city POI recommendations. Memory-based collaborative filtering algorithms can be used in scenarios requiring high real-time performance, but they cannot handle massive amounts of data and usually require sampling or dimensionality reduction of the data.

**Model-based collaborative filtering:** By building models of user and item features, these models are used to predict the user's interest level or rating of the item. Chen et al. [11] proposed a personalized travel planning method based on matrix decomposition and sequential pattern mining, which represents the users' historical behavior data as a user-interest point matrix and learns the embedding vectors of users and interest points through matrix decomposition. Cepeda-Pacheco and Domingo [12] proposed a deep-learning-based tourist attraction recommendation system. Kamble and Kounte [13] used DeepHCF, a deep-learning-based model to solve the data sparsity problem. Madani et al. [14] proposed a method for recommending tourist attractions based on a combination of deep learning and collaborative filtering, which learns the feature vectors of the attractions through convolutional neural networks and the embedding vectors of users and attractions through matrix decomposition, while using attention mechanisms and double-gated recurrent neural networks to improve the recommendation effect.

### 2.2 Collaborative Filtering of Travel POI Recommendations with Graph Neural Networks

The GCNs-CF method can be used to recommend travel points of interest based on the users' historical behavior and preferences, using the similarity and association between travel points of interest to better meet the users' needs. The main idea of the GCNs-CF method is to learn the embedding vectors of users and points of interest through graph convolutional neural networks in order to capture the relationship and similarity between them.

Wang et al. [15] proposed an attentive sequential model based on graph neural network (ASGNN), which uses users' long-term and short-term preferences from their behavioral sequences to make the next POI recommendation. Nan et al. [16]

proposed a collaborative mining and filtering process (CMFP) to reduce the data processing overhead and increase the recommendation ratio. Xin et al. [17] proposed a POI recommendation algorithm using a social-temporal contextual GNN model in location-based social networks. Jiang et al. [18] proposed a unified attention framework for next POI recommendation by modeling users' Long-term and Short-term Preferences via self-supervised learning (LSPSL). Wang et al. [19] proposed a novel graph self-supervised behavior pattern learning model (GSBPL) for the next POI recommendation. Cao et al. [20] effectively integrated the time information and geographic information of users' check-in in the LBSN and proposed a POI recommendation algorithm that comprehensively considers edge devices and the Cloud. Zhang et al. [21] proposed a generic point-of-interest recommendation framework GNN-POI that utilizes the powerful modeling capabilities of GNNs to generate personalized recommendations from node information and topology. This paper presents several graph neural network-based interest point recommendation algorithms, including ASGNN, CMFP, a neural network-based collaborative filtering model, a social-temporal contextual graph neural network model, and a GNN-POI framework. These algorithms use the modeling capabilities of graph neural networks to learn node representations from node information and topology to improve POI recommendations. Some of these algorithms also use information such as user behavior sequences, image tags and social networks to obtain long-term and short-term preferences of users to improve the accuracy and personalization of recommendations.

### 2.3 Summary

Conventional approaches use the modelling capabilities of graph neural networks to learn node representations from node information and topology. Few researchers in the field of personalised travel recommendation algorithms have considered collaborative signals in graph convolutional collaborative filtering, unlike the existing research methods, this algorithm considers the embedding of collaborative signals in collaborative filtering between user-points-of-interest and considers graph convolutional collaborative filtering domain aggregation.

## 3 PROBLEM DEFINITION

In this section, we introduce some concepts and definitions in travel the interest point recommendation.

**Definition 1.** Let $U = \{u_1, u_2, u_3, \ldots, u_m\}$ be the set of users, $P = \{p_1, p_2, p_3, \ldots, p_n\}$ be the set of POIs, and the user-POI interaction matrix $\mathbb{D} \in \mathbb{R}^{m \times n}$ be $m$ users and $n$ POIs.

**Definition 2.** Check-in matrix $R \in \mathbb{R}^{|U| \times |P|}$, the matrix $R$ is a binary matrix that records the check-in records of a user. If user $u_i$ has check-in activity at POI $p_j$, $R_{u_i,p_j} = 1$, otherwise $R_{u_i,p_j} = 0$.

**Definition 3.** The graph $G = (V, \zeta, E)$ is a Bipartite graph, where $V = U \cup P$, $\zeta$ is the set of edges representing the interaction between users and POIs, and $E$ is the embedding matrix.

**Problem (Travel POI recommendation).** The purpose of travel recommendation algorithms is to recommend new locations to users and help them discover cities and attractions that may be of interest to them. The POI recommendation method analyses the users' history to build a model of user preferences for unvisited attractions and then recommends the highest-scoring unvisited attractions to the user.

## 4 PROPOSED FRAMEWORK

### 4.1 Model Construction

Most existing CF models based on GCNs construct a bipartite graph $G = (V, \zeta, E)$ for cooperative signal propagation and updating of user and POI embeddings, where $V = U \cup P$, $\zeta$ is the set of edges representing the interaction between users and POIs. Each user and point of interest is a node of $G$ and is characterized as a learnable embedding vector $e_u \in \mathbb{R}^d$ ($e_p \in \mathbb{R}^d$); by stacking them together, the initial embedding matrix $E$ can be obtained. The GCN model can be represented by the function $f$ in the form shown in Equation (1):

$$f(R^-|G, R^+, \theta) : V_u \times V_p \to \mathbb{R}^+, \tag{1}$$

where $\theta$ is a parameter of the model. The matrix and node form of the update rule is usually expressed as follows:

$$E^{(l)} = \sigma\left(\hat{A}E^{(l-1)}W^{(l)}\right), \tag{2}$$

$$e_u^{(l)} = \sigma\left(\sum_{p \in N_u} \frac{1}{\sqrt{d_u + 1}\sqrt{d_p + 1}} e_p^{(l-1)} W^{(l)}\right), \tag{3}$$

where, $\sigma(\cdot)$ is a nonlinear activation function, $W^{(l)}$ is a weight matrix, $d_u$ and $d_p$ represent the degrees of users and POIs nodes, respectively, and $N_u$ is a POI node directly connected to the user $u$.

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \cdot \tilde{A} \cdot \tilde{D}^{-\frac{1}{2}}, \tag{4}$$

where $\tilde{A} = A + I$, $\tilde{D} = A + I$. $A$, $D$ and $I$ are the adjacency matrix, diagonal degree matrix and identity matrix, respectively.

Node embedding is generally updated from the current embedding of the adjacent aggregated neighbors starting from the initial state $e_u^{(0)} = e_u$. Finally, the pool function is used for generating node embedding. The pool function is usually used to reduce the size of the feature map, thereby reducing the computational complexity and the number of model parameters.

$$o_u = Pooling\left(e_u^{(0)}, \ldots, e_u^{(l)}\right). \tag{5}$$

The interaction between users and POIs is estimated as follows:

$$\hat{r}_{up} = o_u^T o_p. \tag{6}$$

The overall process of CF based on GCNs is to update the node embedding layer using the embedding of the previous layer [7]:

$$E^{(l)} = f\left(E^{(l-1)}, G\right), \tag{7}$$

where $E^{(l)}$ and $E^{(l-1)}$ denote the node embeddings in the $l^{\text{th}}$ and $(l-1)^{\text{th}}$ layers, respectively. $f(\cdot)$ is the function used to update the nodes. The core of GCNs-based CF is to update self-embedding through domain aggregation, and inspired by [22] GCN in user-POI collaborative filtering shown in Figure 1.

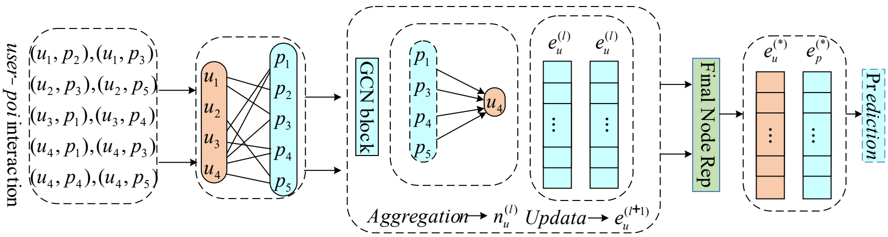


Figure 1. The overall framework of GCN in user-POI collaborative filtering

To explain the process of neighborhood aggregation, we demonstrate the embedded update using user $u$ as the target node:

$$e_u^l = f_{com}\left(e_u^{(l-1)}, f_{aggregation}\left(\left\{e_i^{(l-1)} \mid i \in N_u\right\}\right)\right), \tag{8}$$

where $e_u^l$ and $e_u^{(l-1)}$ denote the embedding of $u$ in the $l^{\text{th}}$ layer and the $(l-1)^{\text{th}}$ layer, respectively; $N_{\text{u}}$ is the domain set of user $u$, that is, the POI visited by user $u$; $f_{aggregation}(\cdot)$ is the embedding aggregation function used to aggregate the neighbors of the $(l-1)^{\text{th}}$ layer; and $f_{com}(\cdot)$ is the function that combines the aggregation embedding of neighbors and the self-embedding of the $(l-1)^{\text{th}}$ layer to update.

In this paper, the high-order embedding of user $u$ is extracted by superimposing the interaction layer of $l^{\text{th}}$ layer. After propagating in $l$ layers, $l$ embedding vectors can be obtained from each layer, and the embedding vectors of each layer can be aggregated by an ensemble function to generate the final embedding vector:

$$e_u^* = f_{\text{int}}\left(\left\{e_u^{(z)} \mid z \in [0, 1, \ldots, l]\right\}\right). \tag{9}$$

After obtaining the final embedding of user $u$ and travel interest point $p$, the users' preference score for the travel interest point is predicted by inner product:

$$\hat{y}_{u,p} = e_u^* e_p^{*^T}. \tag{10}$$

## 4.2 Graph Signal Processing

The image denoising encoder is used to replace the domain aggregation of graphical neural network collaborative filtering to extract the graph signal processing process of important graph features.

Given a graph model containing node and edge information, each node has a feature vector representing the attribute characteristics of the node. These feature vectors are combined into a matrix, which is the graph signal.

For any signal that is on a graph $G$, the graph signal difference is defined as follows [17]:

$$\left\| x - \hat{A}x \right\|. \tag{11}$$

The degree of variation $z(i)$ of the signal $x$ at node $i$ is usually defined as follows:

$$z(i) = \sum_{j \in N_i} w_{ij}(x(i) - x(j)). \tag{12}$$

The metric of change for the entire graph is expressed as [23, 24]:

$$\text{variance} = \sum_{i=1}^{N} z(i)x(i) = \sum_{j \in N_i} w_{ij}(x(i) - x(j))x(i), \tag{13}$$

where $N$ is the total number of nodes, $w_{ij}$ is the weight value of the connection between node $i$ and node $j$, and $A(i,j) = w_{ij}$. Let the eigenvector of the normalized adjacency matrix be $v_t$, and the eigenvalue be $\lambda_t$. Then we have:

$$\left\| v_t - \hat{A}v_t \right\| = 1 - \lambda_t. \tag{14}$$

The amount of variation of a signal on a graph measures the variability of the signal between each node and its neighboring nodes [23, 24]. There is a very direct linear relationship between the total variation of the graph signal and the eigenvalues of the graph; the total variation is a linear combination of all eigenvalues of the graph, with larger eigenvalues being smoother (less variation) and smaller eigenvalues being rougher (more variation). Signals with large variation indicate the variability between each node and its domain, while signals with small variation emphasize the smoothness between them [25].

## 4.3 Proposed GCNs-CF Recommendation Algorithm Based on Travel POI

In this paper, we use the image denoising encoder IDE to replace the domain aggregation process in the collaborative filtering recommendation algorithm based on graph neural networks to fully extract and reduce the dimensionality of node features, so as to achieve the aggregation and representation of node features. The model flow is shown in Figure 2. Wang et al. [26] developed a new recommendation framework - neural graph collaborative filtering (NGCF), which exploits the user item graph structure by propagating embeddings on it. This leads to the expressive modeling of high-order connectivity in the user item graph, effectively injecting the collaborative signal into the embedding process in an explicit manner.

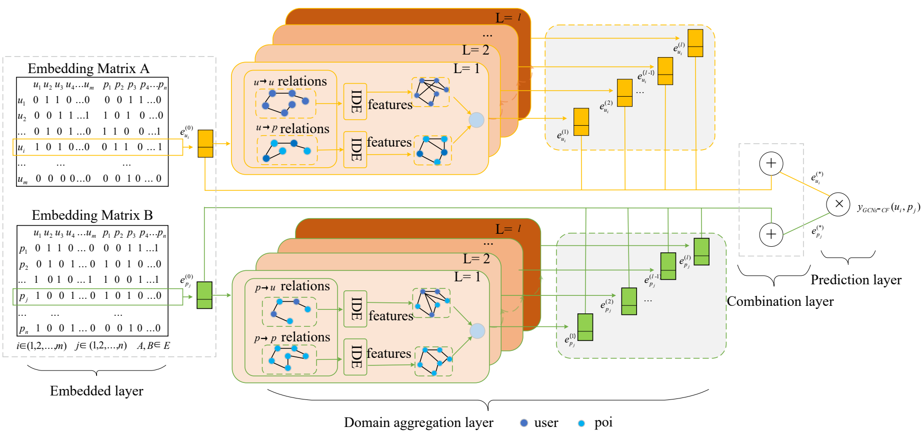


Figure 2. The framework of our proposed GCNs-CF

In Figure 2, an illustration of GCNs-CF model architecture can be seen (the arrowed lines present the flow of information). The representations of user $u_i$ (left) and POI $p_j$ (right) are refined with multiple embedding propagation layers, whose outputs are concatenated to make the final prediction. The model is divided into an embedding layer, a domain aggregation layer and a combination/prediction layer. The layers are described below.

### 4.3.1 Embedded Layer

In order to obtain low-density feature vectors, the feature embeddings of users and POIs need to be initialized at the embedding layer. In the tourism dataset, processed user feature vectors and POI feature vectors can be used to obtain initial embedding vectors of the same dimensionality by means of a single-layer perceptron with Tanh as the activation function:

$$E_u = \tanh(V_u W^u), \tag{15}$$

$$E_p = \tanh(V_p W^p), \tag{16}$$

where $W^u \in R^{d_{V_u} \times d}$, $W^p \in R^{d_{V_P} \times d}$ and Equations (15)and (16) are spliced together to obtain the initial matrix of the propagation:

$$E = [E_u, E_p] = [e_{u_1}, e_{u_2}, e_{u_3}, \ldots, e_{u_n}, ||e_{p_1}, e_{p_2}, e_{p_3}, \ldots, e_{p_m}]. \tag{17}$$

### 4.3.2 Domain Aggregation Layer

The domain aggregator of GCNs-CF is used to aggregate the features of users and interest points into the same domain for better recommendations. Compared to the traditional aggregation layer, the IDE is used instead of the domain aggregator of GCNs-CF to better capture the relationships and features between users and tourism interest point nodes and neighboring nodes. The $l^{\text{th}}$ layer is broadly defined as:

$$e_u^{(l)} = \sum_{p \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_p|}} e_p^{(l-1)}, \tag{18}$$

$$e_p^{(l)} = \sum_{u \in N_p} \frac{1}{\sqrt{|N_p|}\sqrt{|N_u|}} e_u^{(l-1)}, \tag{19}$$

where $e_u^{(0)}$ and $e_p^{(0)}$ are the embedding results of the first layer. Equation (20) can be used to predict whether a user interacts with the POI:

$$R_{up} = \begin{cases} 1, & \text{interactive,} \\ 0, & \text{without.} \end{cases} \tag{20}$$

The embedding vector matrix of users and tourist interest points after the $l^{\text{th}}$ layer is:

$$E^{(l)} = \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}\right) E^{(l-1)}. \tag{21}$$

This gives the final embedding matrix used for prediction as

$$\begin{aligned} E^{(l)} &= \alpha_0 E^{(0)} + \alpha_1 E^{(1)} + \alpha_2 E^{(2)} + \cdots + \alpha_l E^{(l)} \\ &= \alpha_0 E^{(0)} + \alpha_1 \hat{A} E^{(0)} + \cdots + \alpha_l \hat{A}^l E^{(0)}, \end{aligned} \tag{22}$$

here $\alpha_l$ is the importance of the predefined $l^{\text{th}}$ layer representation, which is generally set to $\alpha_l = \frac{1}{1+l}$.

To enhance the informativeness and effectiveness of graph representation, we adopt hypergraphs to represent user-item interactions, which can be seen as a form of dimensionality reduction.

$$A_U = D_u^{-\frac{1}{2}} R D_p^{-1} R^T D_u^{-\frac{1}{2}}, \tag{23}$$

$$A_P = D_p^{-\frac{1}{2}} R D_u^{-1} R^T D_p^{-\frac{1}{2}}, \tag{24}$$

where $D_u$, $D_p$ are the diagonal degree matrices of the user and POIs. In considering the user (POI) relationship, we consider POIs (users) as hyper-edges.

The embeddings generated on hypergraphs are formulated as follows [25]:

$$E_U = \left(O \odot \gamma(U, \pi) O^T\right) E_U, \tag{25}$$

$$E_P = \left(Q \odot \gamma(P, \sigma) Q^T\right) E_P, \tag{26}$$

where $\{O \in \mathbb{R}^{M\times m}, \pi \in \mathbb{R}^m\}$, $\{Q \in \mathbb{R}^{N\times n}, \sigma \in \mathbb{R}^n\}$ are user and POI relations ($A_U$ and $A_P$), $\gamma(\cdot)$ outputs the importance of distinct features to users/POIs, $\odot$ stands for the element-wise multiplication. $E_U$ and $E_P$ are embedding matrices for users and POIs, respectively. Therefore, a paradigm for collaborative filtering based on graph convolution can be expressed as:

$$E^{(l)} = \bar{A}_U E^{(l-1)}. \tag{27}$$

### 4.3.3 Combination Forecasting Layer

After passing through L layers of propagation, we obtain multiple representations for each user $u$. These representations highlight different connections and messages, and therefore have varying degrees of importance in reflecting user preferences. To capture the full range of user preferences, we concatenate these representations to form the final embedding for the user. We perform the same operation on POIs, concatenating the POI representations learned by different layers to obtain the final POI embedding. This process effectively reduces the weight of each individual representation and combines them into a more comprehensive and accurate representation. Combining the different layers of information provides a more comprehensive picture, with the final representation of user $e_u$ and tourism interest point $e_p$ shown in the following equations:

$$e_u^* = \sum_{l=0}^{L} \alpha_l e_u^{(l)}, \tag{28}$$

$$e_p^* = \sum_{l=0}^{L} \alpha_l e_p^{(l)}, \tag{29}$$

$$y_{GCNs-CF}(u,p) = e_u^{*T} e_p^*, \quad e_u^* = e^{(0)}||\ldots||e^{(l)}, \quad e_p^* = e^{(0)}||\ldots||e^{(l)}, \tag{30}$$

where $\alpha_l$ is the importance of the predefined $l^{\text{th}}$ layer representation, which is generally set to $\alpha_l = \frac{1}{1+l}$ . Equation (30) indicates that the user-POI is spliced and then the inner product is predicted, and || indicates the series operation.

### 4.4 Optimization Function

Adaptive adjustment of the negative sample gradient size in GCNs-CF can make better use of the graph structure information between users and interest points to improve the performance of the model and the recommendation effect. Adam (Adaptive Moment Estimation) is an adaptive learning rate optimization algorithm that can automatically adjust the size of the learning rate during the training process. The calculation formula is as follows:

$$\begin{cases} g_t = \nabla \hat{L}(\theta_t), \\ m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \\ v_t = \beta_2 v_{t-1} + (1-\beta_2) {g_t}^2, \\ \hat{m}_t = \frac{m_t}{1-{\beta_1}^t}, \\ \hat{v}_t = \frac{v_t}{1-{\beta_2}^t}, \\ \theta_{t-1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \hat{m}_t, \end{cases} \tag{31}$$

where $t$ denotes the number of iterations; $\beta_1$ and $\beta_2$ are two hyperparameters, usually taking values of 0.9 and 0.999, respectively; $g_t$ is the gradient of the current iteration; $m_t$ and $v_t$ are the first-order and second-order moment estimates of the gradient and gradient squared, respectively; $\hat{m}_t$ and $\hat{v}_t$ are bias corrections for the first- and second-order moment estimates; $\eta$ is the learning rate; and $\varepsilon$ is a very small constant, which usually takes a value of $10^{-8}$.

Adam is a commonly used optimization algorithm [27, 28], which has the advantages of being efficient and adaptive. However, due to the presence of negative samples in the training process, if left untreated, it will lead to the size of the gradient being pulled down, which affects the training effect of the model. In order to solve this problem, the following two methods are considered in this paper:

**Calculating negative sample weights:** There are usually more negative samples than positive samples, so you can consider weighting the negative samples to balance the number of positive and negative samples. Specifically, a weighting factor can be set for the negative samples, such that the gradient calculation offsets the negative sample weights. The general approach is to set a weight for each negative sample, which can be calculated as the ratio of the number of negative samples to the number of positive samples, for example, as the number of positive samples/negative samples.

**Gradient cropping:** Since negative samples may cause the size of the gradient to be pulled down, consider using gradient cropping to limit the size of the gradient. Specifically, a threshold can be set and when the size of the gradient exceeds this threshold, the size of the gradient is cropped to this threshold. This prevents negative samples from having too much influence on the gradient and thus better training the model.

In this paper the Adam method of adaptively adjusting the size of the negative sample gradient is optimized using the method of calculating negative sample weights in conjunction with the characteristics of the dataset, and the adjusted formula is as follows:

$$\begin{cases} g_t = \nabla \hat{L}(\theta_t), \\ m_t = \beta_1 m_{t-1} + (1-\beta_1) w_i g_t, \\ v_t = \beta_2 v_{t-1} + (1-\beta_2) w_i {g_t}^2, \\ \hat{m}_t = \frac{m_t}{1-{\beta_1}^t}, \\ \hat{v}_t = \frac{v_t}{1-{\beta_2}^t}, \\ \theta_{t-1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t+\varepsilon}} \hat{m}_t. \end{cases} \tag{32}$$

The negative sample weight is set as the ratio of the number of positive samples to the number of negative samples by using the equiproportional scaling method in the calculation of negative sample weights, which is calculated as follows:

$$w_i = \begin{cases} \frac{1}{\rho}, & y_i = 1, \\ \frac{1}{1-\rho}, & y_i = 0, \end{cases} \tag{33}$$

where $w_i$ is the weight of the $i^{\text{th}}$ sample, $y_i$ is the label of the $i^{\text{th}}$ sample and $\rho$ is the proportion of positive samples. Adam's algorithm combines the advantages of momentum and RMSprop, which can effectively adapt to different gradient scales and make the gradient descent smoother.

Bayesian Personalized Ranking (BPR) [23, 29] is the most common type of loss function used in recommendation algorithms:

$$L_{BPR} = -In\left(\sigma(\hat{y}_{u,p^+} - \hat{y}_{u,p^-})\right), \tag{34}$$

where $\sigma(\cdot)$ is the activation function, $p^+$ is the POI observed by the user, and $p^-$ is the POI not observed by the user $u$. Since the BPR approach ignores user and item contextual information, it does not sufficiently extract features from users and items, which means its representational power is limited, it performs poorly for sparse datasets, and is susceptible to the cold-start problem. The model can be better evaluated using a loss function incorporating a regularization condition [11]:

$$Loss = -\sum_{(u,p^+,p^-)\in D} In\, \sigma(\hat{y}_{u,p^+} - \hat{y}_{u,p^-}) + \lambda \left\|\Theta\right\|_2^2, \tag{35}$$

where $D = \{(u, p^+, p^-)|(u, p^+) \in R^+, (u, p^-) \in R^-\}$ represents the entire training dataset, and $R^+$ and $R^-$ represent positive and negative samples.

## 5 EXPERIMENTS

### 5.1 Dataset Description

Two public POI travel datasets (Gowalla and Foursquare) and a travel dataset from a related website (Tourism), obtained using crawling techniques were used to evaluate the recommendation performance of the algorithm. The datasets contain information such as user, POI, timestamp, longitude and latitude. The detailed information of these datasets is shown in Table 1.

| Data | Users | POIs | Interactions | Data Density [%] |
|---|---|---|---|---|
| Gowalla | 29 858 | 40 981 | 341 606 | 0.028 |
| Foursquare | 7 642 | 28 483 | 250 271 | 0.115 |
| Tourism | 678 | 1 256 | 58 649 | 6.839 |

Table 1. Statistics of datasets

In data pre-processing, POIs with fewer than 10 signed-in users are removed first, then the data set is randomly divided into a training set and a test set. We use only 20 percent of the user-POI pairs as the test set and the rest are the training set. The training set is used for model training, and the test set data is used to predict the performance of the model.

### 5.2 Evaluation Metrics

In order to evaluate the generalization ability of the model, we use the recall rate (Recall@K) to measure the proportion of accurately recommended travel interest points and the normalized cumulative loss gain ($nDCG@K$) method to measure the ranking performance, and the two metrics are used to further tune the proposed model by stepwise optimization. The recall rate (Recall@K) is calculated as follows:

$$\text{Recall@K} = \frac{S_K^u}{K}, \tag{36}$$

where $S_K^u$ is the number of POIs that users are interested in, and Equation (36) is used to measure the proportion of target tourism interest points in the top-K list.

The normalized discounted cumulative gain takes into account the actual relevance and ranking order of each tourism point of interest and is calculated as follows:

$$\text{IDCG@K} = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log(i+1)}, \tag{37}$$

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log(i+1)}, \tag{38}$$

where $rel_i$ denotes the relevance of the recommendation results for location $i$. $rel_i$ is generally set to a value of 1 for tourism points of interest for which the user gives positive feedback and 0 for the rest of the tourism points of interest. IDCG indicates the list of the best recommendations returned by the recommendation system to a given user, that is, the most relevant results (target tourism points of interest) are placed at the top:

$$\text{nDCG@K} = \frac{DCG@K}{IDCG@K}. \tag{39}$$

The $nDCG$ is calculated in this way, which is equivalent to standardizing it between users so that the $nDCG$ values are comparable between users.

### 5.3 Baseline Model

This paper compares the GCNs-CF approach proposed in this paper with the following recommendation algorithms for POIs of tourism interest.

**NCF [30]:** A collaborative filtering approach based on neural networks, which uses a multilayer perceptron to learn user-item interaction functions proposing a neural network structure to model the latent characteristics of users and items.

**BPRMF [31]:** Based on the assumption that users prefer consumed items over unconsumed items, this algorithm aims to maximize the difference in predicted ratings for these items per user and recommends POIs using implicit feedback data.

**NGCF [26]:** Uses the user item graph structure to emphasize the critical importance of exploiting synergistic signals by propagating embeddings over it.

**LGLMF [1]:** A logistics matrix decomposition POI recommendation model based on local geography.

**LightGCN [32]:** This approach simplifies the design of GCN by proposing a new model that includes only domain aggregation and uses this method for collaborative filtering.

**FG-CF [7]:** A friend-aware graph collaborative filtering approach (FG-CF), which integrates social information into the users' POI graph.

**GDE [25]:** This model investigates how domain aggregation in collaborative filtering of graph convolutional networks can contribute to recommendation algorithms.

**GCNs-CF:** A graph neural network-based recommendation algorithm for the travel points of interest proposed in this paper.

### 5.4 Performance Comparison

In the experiments, we first compare and analyze the recommendation results of the model under Top@10 and Top@20 on three datasets and evaluate the performance of the model using four common metrics of recommendation algorithms; namely Recall@10, Recall@20, nDCG@10, and nDCG@20. Our GCNs-CF model and the baseline model were experimented on three datasets, and the experimental results are shown in Table 2.

| | | NCF | LGLMF | FG-CF | NGCF | BPRMF | LightGCN | GDE | GCNs-CF |
|---|---|---|---|---|---|---|---|---|---|
| Gowalla | nDCG@10 | 0.0361 | 0.0368 | 0.1324 | 0.1401 | 0.1657 | 0.1662 | 0.1672 | **0.1822** |
| | nDCG@20 | 0.0332 | 0.0331 | 0.1206 | 0.1298 | 0.1427 | 0.1556 | 0.1561 | **0.1599** |
| | Recall@10 | 0.0345 | 0.0574 | 0.1276 | 0.1530 | 0.1359 | 0.1521 | 0.1550 | **0.1604** |
| | Recall@20 | 0.0317 | 0.0341 | 0.1031 | 0.1317 | 0.1368 | 0.1373 | 0.1380 | **0.1400** |
| Foursquare | nDCG@10 | 0.0476 | 0.0566 | 0.0823 | 0.0833 | 0.1275 | 0.1276 | 0.1299 | **0.1543** |
| | nDCG@20 | 0.0411 | 0.0418 | 0.0799 | 0.0824 | 0.1081 | 0.1098 | 0.1102 | **0.1288** |
| | Recall@10 | 0.0366 | 0.0665 | 0.0815 | 0.0801 | 0.1132 | 0.1135 | 0.1143 | **0.1343** |
| | Recall@20 | 0.0356 | 0.0429 | 0.0801 | 0.0740 | 0.0975 | 0.1011 | 0.1056 | **0.1097** |
| | nDCG@10 | 0.0391 | 0.0487 | 0.0664 | 0.0893 | 0.0973 | 0.1009 | 0.1109 | **0.1277** |
| Tourism | nDCG@20 | 0.0367 | 0.0405 | 0.0620 | 0.0752 | 0.0840 | 0.0892 | 0.0903 | **0.1136** |
| | Recall@10 | 0.0288 | 0.0386 | 0.0701 | 0.0840 | 0.1045 | 0.1071 | 0.1189 | **0.1294** |
| | Recall@20 | 0.0267 | 0.0359 | 0.0695 | 0.0736 | 0.0911 | 0.0923 | 0.0982 | **0.1092** |

Table 2. Model recommendation accuracy comparison table

As can be seen from Table 2, the model proposed in this paper outperforms other baseline models, both when run on publicly available datasets and when experimented on datasets obtained by ourselves. On the Gowalla dataset, the NCF model has the lowest recall Recall@10, the GDE model is second only to the GCNs-CF model, compared with which the GCNs-CF model improves its accuracy by 3.48 %, and the normalized discounted cumulative gain nDCG@10 and nDCG@20 improve their accuracy by 8.97 % and 2.43 %, respectively, over the optimal baseline model GDE. On the Foursquare dataset, the nDCG@10, nDCG@20, Recall@10 and *Recall*@20 model accuracies improved by 18.78 %, 16.88 %, 17.50 % and 3.88 %, respectively, over the optimal baseline model GDE. On the Tourism dataset, which was obtained by ourselves in this paper, the accuracy of the nDCG@10, nDCG@20, Recall@10, and Recall@20 models improved by 15.14 %, 25.80 %, 8.83 % and 11.20 %, respectively, over the optimal baseline model.

In order to analyze the effect of different recommendation quantities on the recommended effect of the model, $K = 1$, 5 and 10 were chosen as the recommendation quantities. $K = 1$ indicates the unique choice of the user, while $K = 5$ and 10 indicate that the user has different choices, and the recommendation results of different models are shown in Figure 3 and Figure 4.

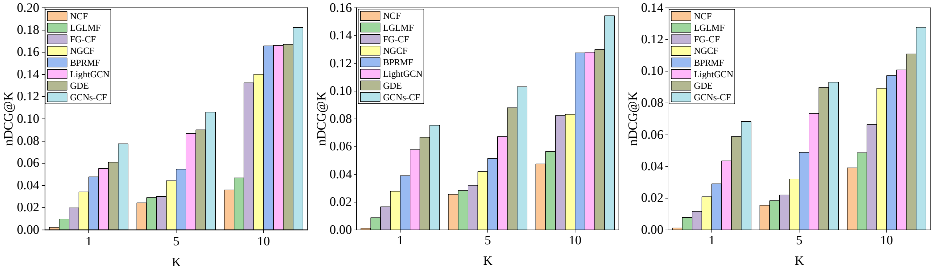


Figure 3. The results of nDCG on Gowalla, Foursquare and Tourism

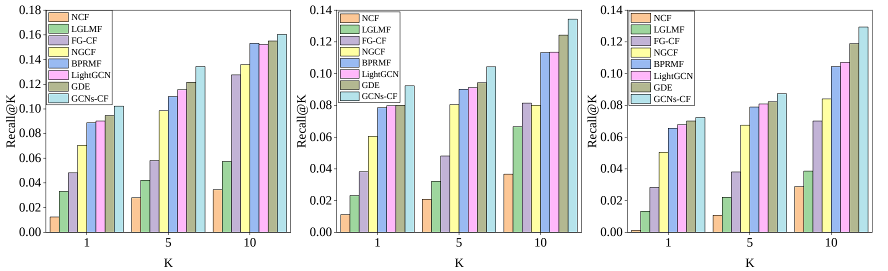


Figure 4. The results of Recall on Gowalla, Foursquare and Tourism

It can be seen from Figure 3 and Figure 4 that the recommendation accuracy of each model on the Gowalla and Foursquare datasets is higher than that on the self-acquired dataset Tourism, which may be due to the unclean data cleaning and small data volume of the self-acquired dataset. The experiments demonstrate that the recommendation performance of the proposed model outperforms the optimal baseline model.

To better evaluate the recommendation performance of the proposed model GCNs-CF, $K = 1$, 5 and 10 were selected as the number of recommendations to compare the recommendation effect of GCNs-CF on the three datasets, and the experimental results are shown in Figure 5.

The choice of the number of convolutional layers is a critical issue when using IDE instead of GCNs-CF for domain aggregation. Deeper networks may capture more complex features, but may also lead to problems such as overfitting and gradient disappearance. In order to select the appropriate number of convolution layers,

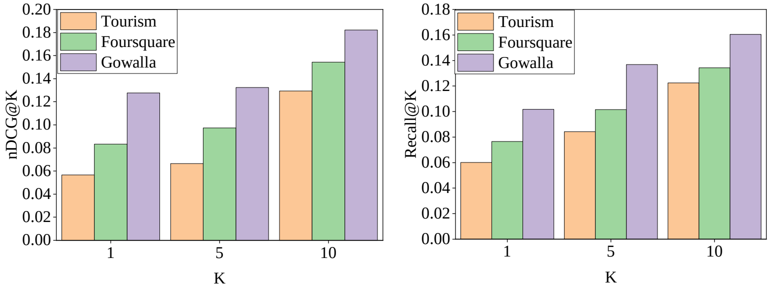


Figure 5. Comparative analysis of recommendation results for different $K$ values on three datasets

the results of comparing the model with different convolution layers under three data sets Recall@10 and nDCG@10 were experimented with, and the results are shown in Figure 6.

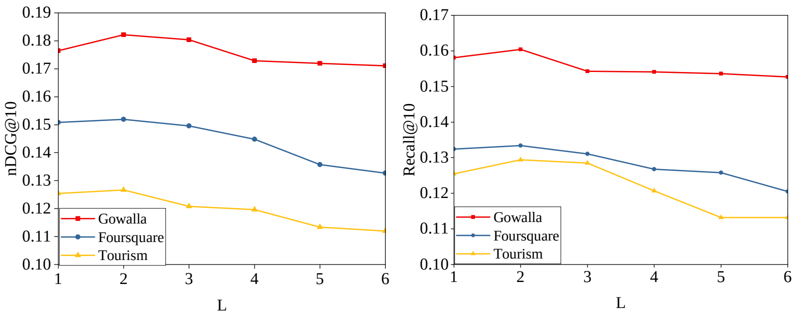


Figure 6. Comparative analysis of recommended results for different number of convolution layers

It can be seen from Figure 6 that the Recall@10 and nDCG@10 values of the GCNs-CF model show an increasing trend from the first to the second layer, and a significant decreasing trend after the second layer. In other words, the best results are obtained when the number of convolution layers is 2, so 2 can be taken as the number of convolution layers.

In the experimental setup involving two convolutional layers, this paper examines the impact of domain aggregation as a key factor for comparison. As evident from the training curve depicted in Figure 7, the proposed method significantly outperforms the traditional domain aggregation approach. Domain aggregation is the process of enhancing the recommendation performance in a recommender system by considering the associations between different users and POIs. This is achieved by representing them as a graph structure and utilizing graph convolutional neural networks to learn representations. These representations enable the extraction of

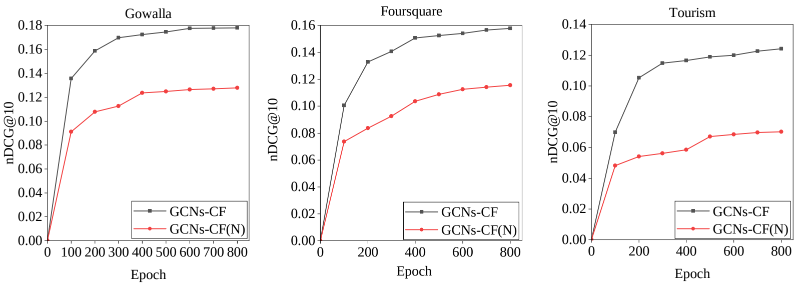


Figure 7. Training curves (where GCNs-CF(N) denotes unchanged domain aggregation)

relationships and features between nodes and their neighboring nodes more effectively.

## 6 CONCLUSIONS

With the development of society, people's living standards are improving and more and more people like to travel; however, the uneven travel recommendation information on the Internet means people are caught in the flood of information and it is difficult to choose the most suitable for their own tourist attractions; therefore, how to achieve accurate personalized travel recommendations is particularly important.

This paper proposes a graph convolutional neural network collaborative filtering method for travel point-of-interest recommendation, which considers the dynamic representation of users and POIs, uses IDE instead of the traditional GCNs-CF domain aggregation method, extracts important user-travel point-of-interest graph features, and selectively embeds higher-order connectivity into the node representation. We also use an adaptive adjustment of the negative sample gradient size to address the problem of slow convergence of graph convolutional neural networks in collaborative filtering personalized travel recommendation algorithms. The experiments were conducted on two publicly available datasets, Gowalla and Foursquare, as well as on our own collected dataset, Tourism, with eight widely used baseline models. The experimental results show that the method proposed in this paper significantly outperforms the baseline model, which means that the method proposed in this paper is feasible in the travel recommendation algorithm.

### Acknowledgments

This work is partially supported by the Guizhou Provincial Department of Education Natural Science Research Project (No. QJJ[2022] 015); the Guizhou Provincial Science and Technology Plan Project Funding (Nos. QKHJC[2018] 1082, QKHJC[2019] 1159); the Science and Technology Projects of Guizhou Province (Nos. QKHJCZK-

2022YB195, QKHJCZK2023YB143, QKHPTRCZCKJ2021007); the Youth Science and Technology Talent Growth Project of Guizhou Province (No. QJHKY2021104); the Natural Science Research Project of Education Department of Guizhou Province (Nos. QJJ2023061, QJJ2023012); the Natural Science Research Project of Department of Education of Guizhou Province (Grant No. QJJ2022015) and (Grant No. QJJ2022047); Guizhou Province University Machine Intelligent Product Research and Development Innovation Team; Provincial Key Laboratory of Pattern Recognition and Intelligent Systems.

## REFERENCES

[1] Rahmani, H. A.—Aliannejadi, M.—Ahmadian, S.—Baratchi, M.—Afsharchi, M.—Crestani, F.: LGLMF: Local Geographical Based Logistic Matrix Factorization Model for POI Recommendation. Information Retrieval Technology (AIRS 2019), Springer, Cham, Lecture Notes in Computer Science, Vol. 12004, 2020, pp. 66–78, doi: 10.1007/978-3-030-42835-8_7.

[2] Zhao, P.—Luo, A.—Liu, Y.—Xu, J.—Li, Z.—Zhuang, F.—Sheng, V. S.—Zhou, X.: Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation. IEEE Transactions on Knowledge and Data Engineering, Vol. 34, 2022, No. 5, pp. 2512–2524, doi: 10.1109/TKDE.2020.3007194.

[3] Xu, H.—Wu, J.—Zhao, L.: Interest Point Recommendation Based on Multi Feature Representation and Attention Mechanism. 2020 IEEE 2nd International Conference on Power Data Science (ICPDS), 2020, pp. 14–17, doi: 10.1109/ICPDS51559.2020.9332495.

[4] Xing, S.—Liu, F.—Wang, Q.—Zhao, X.—Li, T.: Content-Aware Point-of-Interest Recommendation Based on Convolutional Neural Network. Applied Intelligence, Vol. 49, 2019, No. 3, pp. 858–871, doi: 10.1007/s10489-018-1276-1.

[5] Zhai, F.—Li, B.: Point of Interest Recommendation Based on Graph Convolutional Neural Network. Journal of Physics: Conference Series, Vol. 1883, 2021, doi: 10.1088/1742-6596/1883/1/012132.

[6] Chen, X.—Xiong, K.—Zhang, Y.—Xia, L.—Yin, D.—Huang, J. X.: Neural Feature-Aware Recommendation with Signed Hypergraph Convolutional Network. ACM Transactions on Information Systems (TOIS), Vol. 39, 2020, No. 1, Art. No. 8, doi: 10.1145/3423322.

[7] Cai, Z.—Yuan, G.—Qiao, S.—Qu, S.—Zhang, Y.—Bing, R.: FG-CF: Friends-Aware Graph Collaborative Filtering for POI Recommendation. Neurocomputing, Vol. 488, 2022, pp. 107–119, doi: 10.1016/j.neucom.2022.02.070.

[8] Zhang, H.—Gan, M.—Sun, X.: Incorporating Memory-Based Preferences and Point-of-Interest Stickiness Into Recommendations in Location-Based Social Networks. ISPRS International Journal of Geo-Information, Vol. 10, 2021, No. 1, Art. No. 36, doi: 10.3390/ijgi10010036.

[9] Febre, L.—Valdiviezo-Diaz, P.—Reátegui, R.: Collaborative Filtering Enhanced by Demographic Information for Tourist Sites Recommendations. 2022 17th

Iberian Conference on Information Systems and Technologies (CISTI), 2022, pp. 1–6, doi: 10.23919/CISTI54924.2022.9820110.

[10] Li, D.—Gong, Z.: A Deep Neural Network for Crossing-City POI Recommendations. IEEE Transactions on Knowledge and Data Engineering, Vol. 34, 2022, No. 8, pp. 3536–3548, doi: 10.1109/TKDE.2020.3033841.

[11] Chen, L.—Wu, Z.—Cao, J.—Zhu, G.—Ge, Y.: Travel Recommendation via Fusing Multi-Auxiliary Information Into Matrix Factorization. ACM Transactions on Intelligent Systems and Technology (TIST), Vol. 11, 2020, No. 2, Art. No. 22, doi: 10.1145/3372118.

[12] Cepeda-Pacheco, J. C.—Domingo, M. C.: Deep Learning and Internet of Things for Tourist Attraction Recommendations in Smart Cities. Neural Computing and Applications, Vol. 34, 2022, No. 10, pp. 7691–7709, doi: 10.1007/s00521-021-06872-0.

[13] Kamble, S. J.—Kounte, M. R.: Dynamic Traveling Route Planning Method for Intelligent Transportation Using Incremental Learning-Based Hybrid Deep Learning Prediction Model with Fine-Tuning. Transport and Telecommunication Journal, Vol. 23, 2022, No. 4, pp. 293–310, doi: 10.2478/ttj-2022-0024.

[14] Madani, R.—Ez-Zahout, A.—Idrissi, A.: An Overview of Recommender Systems in the Context of Smart Cities. 2020 5$^{\text{th}}$ International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), 2020, pp. 1–9, doi: 10.1109/CloudTech49835.2020.9365877.

[15] Wang, D.—Wang, X.—Xiang, Z.—Yu, D.—Deng, S.—Xu, G.: Attentive Sequential Model Based on Graph Neural Network for Next POI Recommendation. World Wide Web, Vol. 24, 2021, No. 6, pp. 2161–2184, doi: 10.1007/s11280-021-00961-9.

[16] Nan, X.—Kanato, K.—Wang, X.: Design and Implementation of a Personalized Tourism Recommendation System Based on the Data Mining and Collaborative Filtering Algorithm. Computational Intelligence and Neuroscience, Vol. 2022, 2022, Art. No. 1424097, doi: 10.1155/2022/1424097.

[17] Xin, M.—Chen, S.—Zang, C.: A Graph Neural Network-Based Algorithm for Point-of-Interest Recommendation Using Social Relation and Time Series. International Journal of Web Services Research (IJWSR), Vol. 18, 2021, No. 4, pp. 51–74, doi: 10.4018/IJWSR.2021100103.

[18] Jiang, S.—He, W.—Cui, L.—Xu, Y.—Liu, L.: Modeling Long-and Short-Term User Preferences via Self-Supervised Learning for Next POI Recommendation. ACM Transactions on Knowledge Discovery from Data, Vol. 17, 2023, No. 9, Art. No. 125, doi: 10.1145/3597211.

[19] Wang, D.—Chen, C.—Di, C.—Shu, M.: Exploring Behavior Patterns for Next-POI Recommendation via Graph Self-Supervised Learning. Electronics, Vol. 12, 2023, No. 8, Art. No. 1939, doi: 10.3390/electronics12081939.

[20] Cao, K.—Guo, J.—Meng, G.—Liu, H.—Liu, Y.—Li, G.: Points-of-Interest Recommendation Algorithm Based on LBSN in Edge Computing Environment. IEEE Access, Vol. 8, 2020, pp. 47973–47983, doi: 10.1109/ACCESS.2020.2979922.

[21] Zhang, J.—Liu, X.—Zhou, X.—Chu, X.: Leveraging Graph Neural Networks for Point-of-Interest Recommendations. Neurocomputing, Vol. 462, 2021, pp. 1–13, doi:

10.1016/j.neucom.2021.07.063.

[22] Wu, S.—Sun, F.—Zhang, W.—Xie, X.—Cui, B.: Graph Neural Networks in Recommender Systems: A Survey. ACM Computing Surveys, Vol. 55, 2022, No. 5, Art. No. 97, doi: 10.1145/3535101.

[23] Sandryhaila, A.—Moura, J. M. F.: Discrete Signal Processing on Graphs. IEEE Transactions on Signal Processing, Vol. 61, 2013, No. 7, pp. 1644–1656, doi: 10.1109/TSP.2013.2238935.

[24] Sandryhaila, A.—Moura, J. M. F.: Discrete Signal Processing on Graphs: Frequency Analysis. IEEE Transactions on Signal Processing, Vol. 62, 2014, No. 12, pp. 3042–3054, doi: 10.1109/TSP.2014.2321121.

[25] Peng, S.—Sugiyama, K.—Mine, T.: Less Is More: Reweighting Important Spectral Graph Features for Recommendation. Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), 2022, pp. 1273–1282, doi: 10.1145/3477495.3532014.

[26] Wang, X.—He, X.—Wang, M.—Feng, F.—Chua, T. S.: Neural Graph Collaborative Filtering. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), 2019, pp. 165–174, doi: 10.1145/3331184.3331267.

[27] Kingma, D. P.—Ba, J.: Adam: A Method for Stochastic Optimization. CoRR, 2014, doi: 10.48550/arXiv.1412.6980.

[28] Chen, B.—Li, C.—Qi, X.—Ma, H. J.: Adaptive NN-Based Distributed Consensus Control for Nonlinear Multi-Agent Systems Under Direct Graphs. Neural Computing and Applications, Vol. 35, 2023, No. 24, pp. 17795–17807, doi: 10.1007/s00521-023-08646-2.

[29] Rendle, S.—Freudenthaler, C.—Gantner, Z.—Schmidt-Thieme, L.: BPR: Bayesian Personalized Ranking from Implicit Feedback. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09), 2009, pp. 452–461, https://dl.acm.org/doi/abs/10.5555/1795114.1795167.

[30] He, X.—Liao, L.—Zhang, H.—Nie, L.—Hu, X.—Chua, T. S.: Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web (WWW '17), 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[31] Zhang, H.—Ganchev, I.—Nikolov, N. S.—Ji, Z.—O'Droma, M.: Weighted Matrix Factorization with Bayesian Personalized Ranking. 2017 Computing Conference, IEEE, 2017, pp. 307–311, doi: 10.1109/SAI.2017.8252119.

[32] He, X.—Deng, K.—Wang, X.—Li, Y.—Zhang, Y.—Wang, M.: LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), 2020, pp. 639–648.

**Lan Pan** is a Master's student. Her main research interests include massive data statistics and analysis. She is with the Guizhou Key Laboratory of Pattern Recognition and Intelligent System, Guizhou Province University Machine Intelligent Product Research and Development Innovation Team.

**Jiayin Wei** is Ph.D., Associate Professor. His main research interests include big data analysis and processing, recommendation algorithm design and analysis. He is with the College of Data Science and Information Engineering, Guizhou University, Guiyang, China, and also with the Guizhou Key Laboratory of Pattern Recognition and Intelligent System, Guizhou Province University Machine Intelligent Product Research and Development Innovation Team.

**Youjun Lu** is Ph.D., Associate Professor. His main research interests include big data analysis and processing, network communication dynamics research. He is with the College of Data Science and Information Engineering, Guizhou University, Guiyang, China, and also with the Guizhou Key Laboratory of Pattern Recognition and Intelligent System, Guizhou Province University Machine Intelligent Product Research and Development Innovation Team.

**Fujian Feng** is Ph.D., Professor. His research interests include evolutionary computation and micro computation. He is with the College of Date Science and Information Engineering, South China University of Technology, Guangzhou, China, and also with the Guizhou Key Laboratory of Pattern Recognition and Intelligent System.