



Mining multiple sequential patterns through multi-graph representation for next point-of-interest recommendation

Mingxin Gan¹ · Caiping Tan¹

Received: 3 November 2021 / Revised: 28 May 2022 / Accepted: 1 August 2022 /
Published online: 24 August 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Next point-of-interest (POI) recommendation is important for users to help them find interesting venues to visit in the near future. Most previous work on this subject has incorporated geographical and temporal information into sequential patterns to predict next POIs. However, few studies have considered the influence of important factors such as users' reviews or POIs' popularity on sequential patterns, nor distinguished between factors of different importance for prediction. In addition, the relationships between entities in location-based social networks have been ignored in most previous work. To overcome these limitations, we proposed a model called MGCAN to flexibly incorporate various influential factors into different sequential patterns for next POI recommendation. We first used multiple graph convolutional networks and independent attention networks to model multiple sequential patterns with different influential factors. Furthermore, we designed corresponding modules to simultaneously capture general preferences of users and determine the impact of different influential factors on each user. Finally, we used multiple sequential patterns and the general preferences of users in the prediction module to predict the next POI. Experimental results on two datasets showed that the MGCAN model achieved better recommendation performance than benchmark models.

Keywords Next POI recommendation · Graph convolutional neural networks · Attention networks · Sequential patterns · Multiple influential factors

Caiping Tan contributed equally to this work.

✉ Mingxin Gan
ganmx@ustb.edu.cn

Caiping Tan
s20190991@xs.ustb.edu.cn

¹ School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China

1 Introduction

With the development of the Internet and mobile technology, location-based social networks (LBSNs) such as Yelp, Foursquare in the United States, and Dianping in China are becoming increasingly widespread in everyday life. On LBSNs, users upload photos related to venues they have visited, and share their experiences of and views on these venues with friends. However, it may be difficult for a user to quickly find interesting venues given the massive amount of information available. In order to solve this problem, point-of-interest (POI) recommendation is widely used in LBSNs to help users find interesting venues such as restaurants, shopping malls, movie theaters and many others, and push information about these venues to them; this serves users and also brings opportunities to businesses. Thus, POI recommendation has been the subject of widespread research [1–3].

There are two types of methods for point-of-interest recommendation in existing works: global recommendation [4–6] and next-POI recommendation [7, 8]. Global POI recommendation predicts venues that a user will probably be interested based on the overall historical check-in activities of users. Next POI recommendation aims to provide personalized recommendations to a user depending on where the user is likely to be next, which is determined based on users' historical check-in sequences. The latter approach has more important practical significance than the former, because it focuses on mining user behavior patterns with temporal information and predicting possible future visits. For instance, it suggests to a user where they can have dinner after work, or where to watch a movie after dinner. Thus, the topic of next POI recommendation has been extensively studied by researchers in recent years.

There are large numbers of sequential interactions between users and locations in LBSNs. The next POI visited has a strong relationship with the user's previous behaviors, and there is a certain sequential dependency in the user's interaction behaviors. For example, a user may watch a movie after having dinner, or visit a coffee shop in a shopping mall after going shopping. Therefore, most next POI recommendation methods focusing on mining users' sequential patterns (short-term preferences). Early methods used for recommendation of the next POI mostly used a Markov chain approach, for instance, the FPMC-LR model [9] calculated the transition probability between POIs, taking localized regions into consideration, to predict the next POI. However, such methods can only capture short-term sequential dependency. To capture long-term sequential dependency, recent work has used recurrent neural networks (RNNs) [10] with memory mechanisms, as well as RNN variants [11, 12], such as long short-term memory (LSTM) units and gated recurrent units (GRU). Inspired by model *Transformer* [13], many studies have attempted to use self-attention mechanisms to learn sequential patterns [14–16]. In addition, some models take various influential factors in LBSNs into account to improve the recommendation accuracy [4, 17, 18]. Some studies have also tried to capture rich semantic information with graph neural networks (GNNs) for use in POI recommendation [19, 20]. Although the existing methods for next POI recommendation have achieved great success, they have the following limitations.

(1) Most previous work has incorporated geographical–temporal factors into sequential patterns [21–23] but has not used review information and popularity information. Users' reviews of venues contain rich preference information and can reveal the different characteristics of venues. Popularity represents how popular venues are, and some users usually choose to visit places that are currently popular. In particular, time-related popularity reveals the time periods in which a venue is more popular. For example, movie theaters

are more popular in the evening, whereas restaurants are more popular during mealtimes. Ignoring such meaningful information may lead to failure to accurately capture user interests, resulting in lower recommendation accuracy.

(2) LBSNs include a variety of rich entity relationships, such as relationships among users or venues, and relationships between users and venues. When time, category, *etc.*, are considered as entities, there will be a richer network of relationships. Some studies have shown that using relationships between entities to model the higher-order features of users or POIs can lead to better representation. However, existing methods for next POI recommendation do not use multiple entity relationships.

(3) Most previous work has involved constructing check-in sequences of users based on time to capture a single sequential pattern. Some studies focused on sequential dependencies between check-ins of a user, whereas others emphasized the relationships between the user’s check-in activities and target check-in behaviors. However, for different sequential patterns, there will be a different impact of the user’s historical check-in activities on the target behavior. For example, for a check-in sequence $a \rightarrow b \rightarrow c \rightarrow d$, the relationship between historical check-in c and target behavior d may be closer in geographical–temporal sequential patterns, whereas check-in b has a greater impact on d in review sequential patterns. Therefore, we should consider incorporating multiple influential factors into sequential patterns and distinguish the impact of different sequential patterns.

To overcome the above limitations, we propose a novel next POI recommendation method, named MGCAN, which uses GNNs and attention networks to incorporate geographical–temporal factor, review factor, and popularity factor into next POI recommendation, as shown in Figure 1. Different from the existing studies, instead of directly fusing the embedding of various information together through concatenation or addition, we used different influential factors as contextual embedding independently, using multiple parameterized kernel functions to learn multiple sequential patterns and general preferences of users. As well as capturing the sequential dependencies of user’s behavior sequences, we additionally used multiple contextual attention mechanisms to distinguish the fine-grained impact of different influential factors on users’ behavior. The geographical influence factor

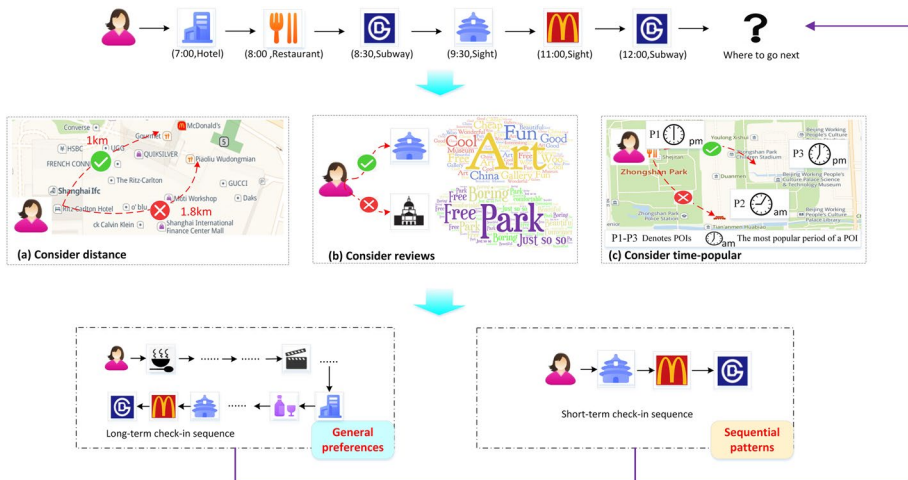


Fig. 1 Overview of proposed MGCAN model

is a unique information attribute in POI recommendation. Some previous studies used probability distribution functions to model geographical influence [6, 24, 25]. Others tried to capture the geographical relationship between POIs by deep learning techniques [20, 26], and demonstrated the effectiveness of geographical relationship embedding among POIs as geographical influence. Therefore, we constructed a POI–POI graph with geographical distance correlations and used GNNs to capture higher-order connectivity among POIs as for geographical–temporal embedding. In addition, most of the previous studies combined probability statistical methods and matrix factorization to model popularity, but their expressive ability is limited and it is difficult to capture complex semantic information. Therefore, we constructed a POI–time graph containing popularity information, and used time-aspect check-in frequency as the information of the edges in the graph. Then we extracted time–popularity relationships among POIs by recursively propagating nodes’ embedding on the graph to obtain excellent feature representations.

To summarize, our contributions are as follows. (1) We proposed a recommendation model, MGCAN, which used different influential factors as contextual embedding independently to obtain multiple sequential patterns and general preferences for use in recommendations. We effectively distinguished the impact of different influential factors on the behaviors of a user via the design of multiple contextual attention mechanisms. (2) We used GNNs to capture the high-level implicit relationships between nodes, in particular, for time–popularity relationships among POIs. We used the time-aspect check-in frequency as the edge information, and made full use of node information and edge information to obtain the better feature representation. (3) We conducted comparative experiments and discussed the recommendation performance and advantages of the MGCAN model, and revealed the improvements achieved with MGCAN compared with other methods.

The rest of this paper is structured as follows. Section 2 discusses related work on next POI recommendation. Section 3 describes the proposed MGCAN approach in detail. Section 4 presents an evaluation and discussion of the method. Section 5 describes the conclusions drawn from our results and proposes further work for the future.

2 Related works

2.1 Next POI recommendation

In contrast to traditional POI recommendation, next POI recommendation aims to help users find venues that they may be interested in next. This requires a focus on learning sequential patterns of users through mining their historical trajectories. Early work mainly used Markov chains to learn transition patterns between POIs, often combined with matrix factorization technology for next POI recommendation. For example, FPMC-LR [9] was the first model proposed to solve the successive POI recommendation problem, and used matrix factorization to incorporate personalized Markov chains and regional constraints of users’ behavior. LORE [27] first learned sequential feature from a historical check-in sequence, modeled as a location–location transition graph, and then used an additive Markov chain on the graph to predict venues that a user would probably visit next.

Methods including matrix factorization, tensor factorization, and embedding have been used to predict transition probability. Feng et al. [28] used metric spaces to model distances between POIs for prediction of transition probability. STELLAR [29] modeled sequence based on a tensor decomposition framework, and captured and utilized the spatio-temporal

effects of check-in. However, the STELLAR model only focused on a pair of continuous venues and failed to model the user's overall check-in sequence. Chang et al. [30] proposed the CAPE model, which utilized check-in sequence of users and text content about POIs. Zhang et al. [31] developed a word2vec-based framework to embed each POI into a space so as to learn the sequential relationship between POIs.

In recent years, the most commonly used methods have been based on neural network models, especially RNNs. These have obvious advantages compared with methods based on Markov chains and matrix factorization. RNNs have often been used to model sequential patterns in sequence-learning tasks [32, 33]. Cui et al. [10] proposed the Distance2Pre model, which used RNNs to capture a user's sequential preferences and then modeled the distances between successive POIs to obtain spatial preferences. Wu et al. [11] proposed the LSPL method, which learned the contextual features of POIs and used general preferences and sequential preferences of users to provide recommendations. LSTPM [34] is a hybrid long-term and short-term preference model that models temporal and spatial preferences as long-term preferences and constructs a check-in sequence based on location distance to obtain sequential features as short-term preferences. Manotumrukta et al. [35] proposed CRCF, which used a GRU-based RNN component to model users' sequence preferences and leveraged contextual information of check-in sequences of users. Gan et al. [36] proposed the DeepAssociate approach, which utilized RNNs, LSTM, and GRU to learn sequence patterns and explored the sequential influence by different methods.

RNNs overemphasize strongly dependent adjacent interactions; however, adjacent interactions do not always have a strong dependency in the real world owing to noisy data in the sequence. Previous studies [37, 38] have used an attention mechanism to capture the truly relevant interactions in users' sequences. In the past 2 years, many studies [14, 16, 39] have used self-attention to learn sequential features, demonstrating excellent performance and efficiency in the task of sequential recommendation. STGCN [19] required the construction of multiple graphs based on check-in time and used GNNs to extract user–region and user–POI periodic patterns, but it failed to consider temporal continuity. GEAPR [40] combined structural context, neighbor influence, users' attributes, and geographical influence to predict users' preferences, and used an attention mechanism to distinguish the impact of different influential factors on users' behavior. However, GEAPR only focused on users' global preferences and did not incorporate their sequential preferences. ASGNN [41] used gated GNNs to model users' behavior patterns and a personalized hierarchical attention network to learn the correlations between users and POIs in the check-in sequence.

Most early work on next POI recommendation focused on mining sequential patterns, and exploring the sequential dependency of POIs in the check-in sequence or the correlations between POIs in the check-in sequence and target POIs. Most studies only incorporated geographical–temporal influence into next POI recommendation. For example, GT-HAN [23] used geographical–temporal attention to model geographical–temporal influence, and collaborative attention to distinguish the impact of historical check-in activities on user preferences. Some studies also incorporated social relations into sequential patterns. For example, Yang et al. [3] used H-deepwalk to capture social relations and geographical influence, and further learned long- and short-term preferences for next POI recommendation. SSSER [42] used a metric learning method to capture the social relationships among users for next POI recommendation. Recently, some studies have shifted the focus to category-aware methods for next POI recommendation. The problem of data sparseness makes it difficult to mine user preferences from their checked-in POIs; however, POI category preference can be exploited to compensate for this deficiency [8]. ATCA-GRU [43] combines GRU and an attention-based category-aware method to predict the

next POI category. CHA [44] uses an attention-based hierarchical category knowledge graph to learn POI embedding, in order to embed the check-in sequence, and further captured sequential patterns. However, there has been a lack of effective exploration and utilization of other auxiliary information such as reviews, popularity, and visual contents in the task of next POI recommendation.

2.2 Graph neural networks for recommendation

Traditional deep learning techniques such as RNNs and convolutional neural networks have shown a great advantages in capturing potential patterns of Euclidean data [45], but they are not suitable for processing graphics data. In order to apply deep learning technology to graphic data, many studies have explored the use of new deep neural network technologies to process graphic data, called GNNs, such as GCNs [46]. GNNs can aggregate the feature information of neighbors and structural information, and have excellent representation learning ability.

GNNs have been extensively applied in many fields including machine translation [47], traffic prediction [48], chemistry [49] and medicine [50]. GNNs have also been used for recommendation tasks in recent years. Ying et al. [51] constructed interactive graphs and used an efficient random walk method and GCN method to obtain nodes' representations containing structural information in graphs, and nodes' own feature information for large-scale Web-level recommendation tasks. Wu et al. [52] used dual graph attention networks to learn feature representations for two-fold social influence extending from the user domain to the item domain, which could effectively alleviate the data sparsity problem and obtain static and dynamic different-depth representations of users and items. Another study [53] used attribute graphs instead of the commonly used user–item interaction graphs. Gated GNNs were used to effectively aggregate attributes of different modalities of neighbors to enrich presentations. Ji et al. [54] used GNNs to model relationships of users, news, and topics. In some studies on POI recommendation, GNNs were used to capture distance relationships between POIs or social relationships between users. Zhong et al. [20] constructed a POI geographic relationship graph and a user social relationship graph, and then used GCNs to learn POIs' location representations and users' social representations.

These studies demonstrate that GNNs have been extensively applied in various tasks and fields and have exhibited excellent performance. Thus, GNNs are expected to be powerful tools for processing various types of heterogeneous information and complex relationships between users and/or POIs in LBSNs.

3 The proposed model

In this section, we describe our MGCAN model in detail. First, we illustrate the formulations and definitions associated with the MGCAN model. Then, we explain the basic components of our model, which include embedding of multiple influential factors, multiple sequential patterns, and learning of general preferences, and a prediction module, as shown in Figure 2. Specifically, we consider various influential factors in LBSNs, including geographical–temporal factors, time–popularity, and review text. Next, we use two independent GCNs to model the representation of POIs by a neighborhood propagation mechanism on two graphs, i.e., the POI–POI graph and the POI–time graph. In addition, we use multiple attention networks to capture sequential patterns and general preferences while learning the

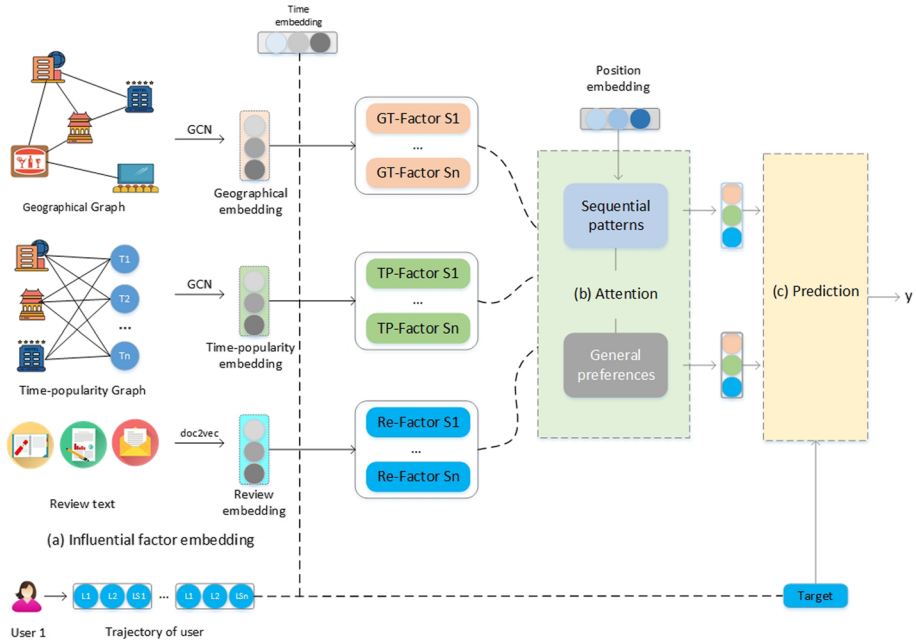


Fig. 2 Framework of the MGCAN model

impact of each check-in in a historical trajectory on the next predicted venue. Finally, we make predictions of possible POIs for each user to visit next based on their sequential patterns and general preferences.

3.1 Problem formulation

We provide a formulation and definition of the proposed method for next POI recommendation in this section. The main symbols used in this paper are shown in Table 1.

Definition 1 (User): We denote by $U = \{u_1, u_2, u_3, \dots, u_{|U|}\}$ a group of users in LBSNs.

Definition 2 (POI): We denote by $V = \{v_1, v_2, v_3, \dots, v_{|V|}\}$ a group of venues in LBSNs. Each venue v includes information on the geographical location (latitude and longitude), relevant reviews, and popularity.

Definition 3 (Check-in): We denote by $c_t^u = \{u, v, t\}$ a check-in, which indicates that user u visited venue v at t time.

Definition 4 (Check-in sequence): We use $L_t^u = \{c_{t-l+1}^u, c_{t-l+2}^u, c_{t-l+3}^u, \dots, c_t^u\}$ to denote a check-in sequence of user u , where L_t^u represents a list of POIs that that user u has visited before, in ascending order by time of visit t . l is the length of the sequence.

Definition 5 (Historical Trajectory): The historical trajectory is the continuous check-in sequence of a user. We denote by $S_t^u = \{c_{t-s+1}^u, c_{t-s+2}^u, c_{t-s+3}^u, \dots, c_t^u\}$ a historical trajectory of

Table 1 Key symbols

Symbols	Interpretation
U, V, T	A group of users, the POIs, and a time interval, respectively
u, v, t	User, POI, and check-in time
L_t^u, S_t^u	The check-in sequence and historical trajectory of user u before time t
s, l	The lengths of the short check-in sequence and long check-in sequence
d	Dimension of embedding
β	The number of heads of multi-head self-attention
h_u, h_v	Pre-training embedding of users and POIs by metapath2vec
h_{GT}, h_{Re}, h_{TP}	The factor-specific embedding of POIs, <i>i.e.</i> , geographical–temporal embedding, review embedding, and time–popularity embedding
$E_{GT}^S, E_{Re}^S, E_{TP}^S$	The factor-specific embedding of the short check-in sequence
$E_{GT}^L, E_{Re}^L, E_{TP}^L$	The factor-specific embedding of the long check-in sequence
$h_{GT}^S, h_{Re}^S, h_{TP}^S$	Sequential patterns based on factor-specific embedding
$h_{GT}^L, h_{Re}^L, h_{TP}^L$	General preferences based on factor-specific embedding
h^S, h^L	The final sequential pattern and general preferences of users
$y_v^{u,t}$	Predicted probability that user u will visit POI v at the next time t

user u before time t . s is the length of the trajectory. Trajectories of a user are generated from check-in sequences of the user within a certain time interval, so S_t^u is a subset of L_t^u . Each user has multiple historical trajectories, which contain different numbers of check-ins, as shown in Figure 3. We transform historical trajectories of different lengths into same-length trajectories by padding with zeros and masking off the padding in calculations.

Given a current historical trajectory S_t^u , a long check-in sequence L_t^u of user u , and location candidates V , the goal of our MGCAN model is to predict a list of venues that user u is most likely to visit next.

3.2 Multiple influential factors embedding

The influential factors embedding module consists of four parts: user–POI interaction embedding, geographical–temporal embedding, review feature embedding, and time–popularity embedding, as shown in Figure 2(a).

3.2.1 User–POI relationship embedding

As described by Dong [55], we constructed a user–POI interaction graph, used a random walk to generate walk-path text, and then used the doc2vec method to capture the feature representations of users and POIs, denoted by h_u and h_v , respectively. These random-walk-based meta-paths ensured that the pre-training presentations of users and POIs contained potential interaction relationships between users and POIs.

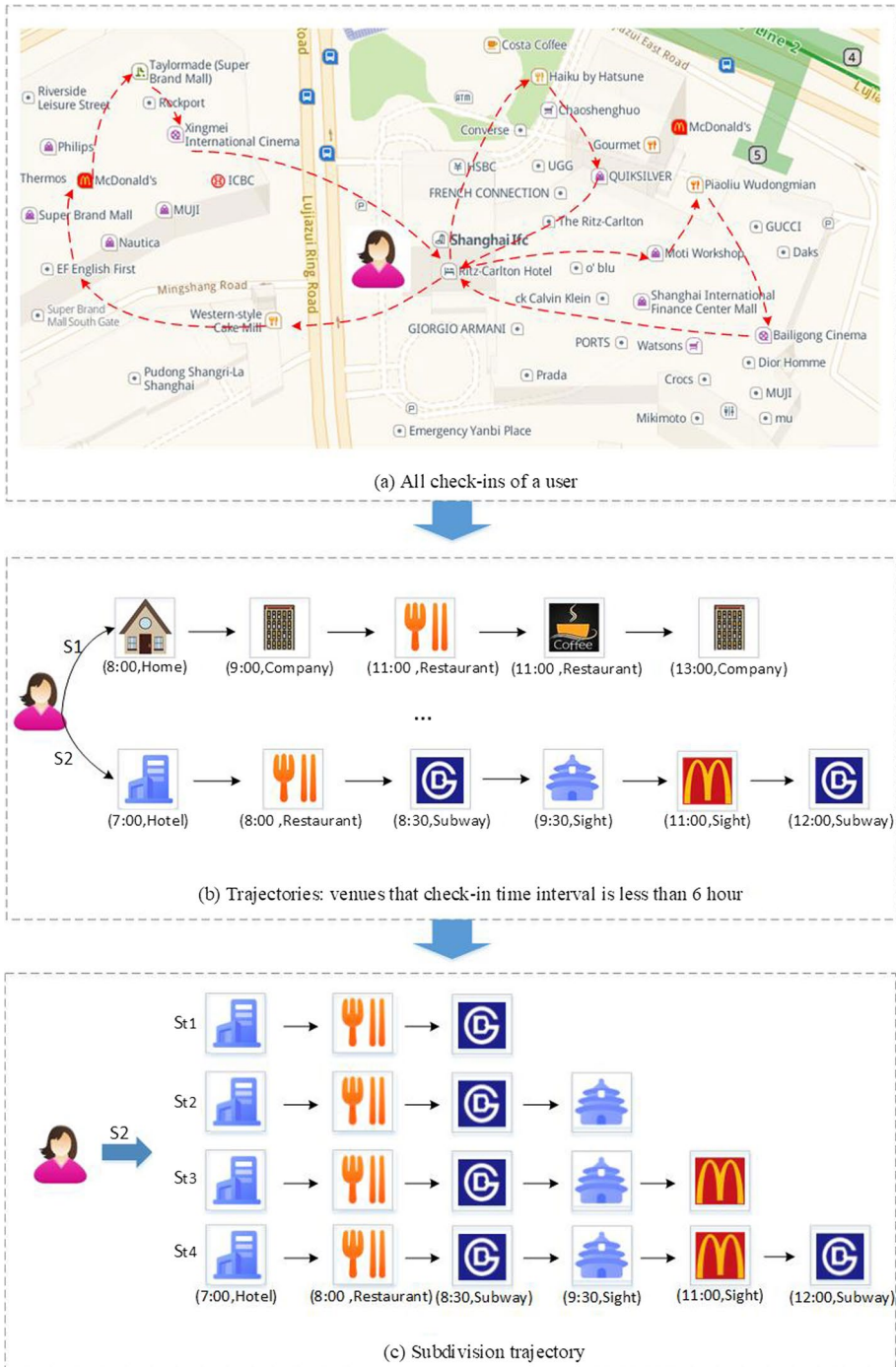


Fig. 3 Trajectories processing of a user

3.2.2 Geographical-temporal embedding

In order to embed temporal information, we adopted the method of Zhou et al. [56]. For a historical trajectory, we calculated the time interval between each check-in and the last check-in, and then mapped the gap lengths of intervals from $[0,1), [1,2), [2,4), \dots, [2^k, 2^{k+1})$ to categories of $0, 1, 2, \dots, k+1$. Then, we performed categorical feature lookups to obtain the time interval embedding in a historical trajectory of user u , denoted by h_T .

Most previous work has represented geographical influence via modeling Euclidean distances between two successive check-ins. However, it is hard to capture high-order relationships among venues in this way. For example, if venue a has a neighbor relationship with venue b , venue a also has a high-order neighbor relationship with venue c , which is a neighbor of b . Such high-order neighbor relationships of venues can be effectively captured by GNNs. Therefore, in order to capture better representations for POIs, we used GCNs to aggregate neighbor information and capture geographical relationships in the location graph.

First, we constructed a location graph $G = (V, E)$, where $V = \{v_1, v_2, v_3, \dots, v_{|V|}\}$ indicates a group of POIs and E indicates the edges between each pair of POIs. The weights of the edges represent the similarities between each pair of POIs. The similarity is defined by the Gaussian radial basis function kernel:

$$a_{ij} = \exp(-\eta \cdot \text{dist}(v_i, v_j)), \quad (1)$$

where $\text{dist}(v_i, v_j)$ indicates the distance between venue v_i and v_j , and η is a hyperparameter used to control the level of geographic correlation. We also use an adjacency matrix $A_g \in \mathbb{R}^{|V| \times |V|}$ to represent the information in the location graph, where a_{ij} is the element in the i -th row and j -th column of A_g .

Then, we obtain the symmetric normalized Laplacian matrix using the following formula:

$$L_g = \tilde{D}_g^{-\frac{1}{2}} \tilde{A}_g \tilde{D}_g^{-\frac{1}{2}}. \quad (2)$$

We added self-connections to the adjacency matrix to combine nodes' own features with neighbor features to update nodes, i.e. \tilde{A}_g . We added self-connections to the adjacency matrix to combine nodes' own features with neighbors' features to update nodes, i.e., \tilde{A}_g . Next, we initialize a matrix $V_{\text{location}} \in \mathbb{R}^{|V| \times d}$ as the feature vector of POIs and let V_{location} be the input of the first layer of GCNs, which is denoted by $H^{(0)}$. The output of the k -th layer is as follows:

$$H_g^{(k)} = \text{LeakyReLU}((L_g \times H_g^{(k-1)}) \times W_g + b_g), \quad (3)$$

Where non-linear activation function $\text{LeakyReLU}()$ allows messages to encode positive signals and small negative signals [57]. $W_g \in \mathbb{R}^{d \times d}$ and $b_g \in \mathbb{R}^{1 \times d}$ are two trainable parameter matrices. Moreover, we use three-layer graph convolution to learn the geographical embedding. The geographical embedding of a POI is denoted by h_G . We obtain geographical-temporal embedding by summing h_T and h_G ; the result is denoted by h_{GT} .

3.2.3 Review feature embedding

Users' comments on POIs represent their views and feelings about the venues, and users learn more about a venue through reviews, which help them to make decisions. In this work, we used pre-trained review feature embedding as feature representation of a POI based on the text of reviews about the POI. The review text of a POI includes all comments

on the POI by users, which obviously results in a long text. We learned the review feature embedding vectors of POIs using doc2vec [58], a simple and efficient method for learning document information, which simultaneously considers the context and semantic information of words, sentences, and paragraphs. We used the distributed memory model of sentence vectors (PV-DM) for pre-training, as it takes the word order into consideration and has usually worked well for most tasks. We used h_{Re} to represent the review feature embedding of a POI trained by doc2vec.

3.2.4 Time-popularity embedding

Unlike general popularity, time-based popularity indicates the time periods in which a venue is more popular [59]. Users tend to visit venues that are popular with the public. Moreover, a venue usually has higher popularity during a certain period of time. For example, restaurants are usually more popular at mealtimes, and bars are usually more popular in the evening. Therefore, we incorporated time-based popularity of POIs as a factor influencing users' decisions into our POI recommendations. However, most previous work has directly used the frequency of visitation by users as the popularity feature of POIs but failed to capture the fine-grained features of POIs based on popularity. For example, movie theaters and bars are more suitable for evening visits, and these two types of venue have similar popularity. However, the relationships between venues of similar popularity have not been fully exploited in most studies. To alleviate this problem, we adopted GCNs to model fine-grained time-popularity feature embedding of POIs.

First, we constructed a frequency matrix $F \in R^{|V| \times T}$, as shown in (4). Here, f_{v_i, T_s} represents the elements of F , referring to the ratio of the number of POIs v_i visited in time slot T_s to the total number of POIs v_i visited in all time slots; and T is the number of time intervals. We set 24 h as the time interval.

$$f_{v_i, T_s} = \frac{\sum_{u \in U} num_{u, v_i, T_s}}{\sum_{u \in U} \sum_{T_s \in T} num_{u, v_i, T_s}}, \quad (4)$$

Then, we obtained an adjacency matrix with self-connections \tilde{A}_{tp} for the POI–time frequency graph as follows:

$$\begin{bmatrix} 0 & F \\ F^T & 0 \end{bmatrix}, \quad (5)$$

$$\tilde{A}_{tp} = A_{tp} + I. \quad (6)$$

As in Section 3.2.2, we applied \tilde{A}_{tp} to (2) to obtain Laplacian matrix, and concatenated the initializations of POIs and time intervals. The result, denoted by $H_{tp}^{(0)} \in R^{(|V|+T) \times d}$, was applied to (3) to obtain the time-popularity embedding, denoted by h_{tp} .

3.3 Multiple sequential patterns and general preferences learning

Many studies of next POI recommendation have combined long-term and short-term preferences of users [11, 34, 60–62] to improve the accuracy of venue recommendations. However, the sequential pattern and general preference were usually single, and were obtained by directly integrating multiple types of information. In contrast to previous work, we

combined multiple fine-grained sequential patterns and general preferences of a user to provide recommendations.

As mentioned earlier, we consider three influence factors that have an impact on users’ travel decisions. We also set a unique attention module for each influential factor (e.g., geographical–temporal factor, review factor, and time-popularity factor). We construct attention networks to obtain multiple fine-grained sequential patterns and general preferences of a user. Moreover, we distinguish the different impacts of the geographical–temporal factor, review factor, and time-popularity factor on the behavior of a user by an attention mechanism.

3.3.1 Multiple sequential patterns

Accurate modeling of user sequential patterns is the basis and the premise of next POI recommendation. Behaviors of users usually changes over time, for example, one user may usually visit a company in the morning, visit a restaurant at noon, and go home in the evening. Another user may usually visit a company and some restaurants on weekdays, and entertainment venues such as movie theaters and KTVs on weekends. Therefore, we cannot mine a user’s interest preferences at weekends from their historical behavior on weekdays. To this end, in this subsection, we use an attention network to capture the sequential patterns of users based on different influential factors and at different times.

First, we divide the users’ check-ins into the different historical trajectories. Two consecutive check-ins with a check-in interval of less than 6 hours belong to the same trajectory, as a short-term sequence of users, which is defined as $S_t^u = c_{t-s+1}^u, c_{t-s+2}^u, c_{t-s+3}^u, \dots, c_t^u$. Based on the multiple influential factor embedding h_{GT}, h_{Re}, h_{TP} of POIs learned in Section 3.2, we embed the user’s short-term check-in sequence S_t^u . The factor-specific embedding matrix of the short-term check-in sequence is denoted by $E_\gamma^S \in R^{s \times d} (\gamma \in \{GT, Re, TP\})$. GT, Re, TP indicates the geographical–temporal factor, review factor, and time-popularity factor, respectively, where d is the dimension of embedding and s is the length of the historical trajectory.

Attention networks have been used for sequence modeling with promising results [13]; the main types are multi-head self-attention and feed-forward networks. Multi-head self-attention effectively extracts the sequential dependency in the check-in sequence of a user. However, the self-attention operation is not aware of the order of POIs in the check-in sequence, so we use the timing signal approach [13] to encode the positional embedding P and add the positional embedding to the factor-specific embedding matrix of the short-term check-in sequence E_γ^S . The check-in sequence embedding matrix with P is denoted by $\tilde{E}_\gamma^S (\gamma \in \{GT, Re, TP\})$. Then, we use self-attention to extract multiple sequential patterns of users as follows:

$$head_i = softmax\left(\frac{QK^\top}{\sqrt{d/h}}\right)V, \tag{7}$$

$$MultiheadAtt(\tilde{E}_\gamma^S) = concat(head_1, head_2, \dots, head_\beta)W^O, \tag{8}$$

$$A_\gamma^S = LayerNorm(MultiheadAtt(\tilde{E}_\gamma^S) + \tilde{E}_\gamma^S), \tag{9}$$

where a set of Q, K, V is constructed from the same factor-specific embedding matrix of the short-term check-in sequence, *i.e.*, $Q = \tilde{E}_\gamma^S W_i^Q$, $K = \tilde{E}_\gamma^S W_i^K$ and $V = \tilde{E}_\gamma^S W_i^V$. W_i^Q, W_i^K, W_i^V, W^O are the trained parameters, and β is the number of heads. Equation (8) is intended to concatenate the outputs of multi-head self-attention. Then, residual connection and layer normalization [63] are applied to leverage any low-level information in (9), where $LayerNorm(\cdot)$ is the layer normalization function.

The feed-forward network is usually applied after multi-head self-attention to introduce non-linearity into the model for better representation; this consists of a fully connected layer together with add and norm operations as follows:

$$FFN(A_\gamma^S) = ReLU(A_\gamma^S W_1 + b_1) W_2 + b_2, \quad (10)$$

$$F_\gamma^S = LayerNorm(FFN(A_\gamma^S) + A_\gamma^S), \quad (11)$$

where $ReLU(\cdot)$ is used to obtain the non-linear presentation, and (11) uses the $LayerNorm(\cdot)$ function for normalization. We finally obtain the users' multiple sequential patterns, $F_\gamma^S (\gamma \in \{GT, Re, TP\})$.

3.3.2 Multiple general preferences

In a process similar to that used to learn sequential patterns, as described in Section 3.3.1, the attention network can be used to model multiple general preferences of users. In our MGCAN model, the general preferences of a user are learned based on long-term check-in records. We perform a look-up operation on the multiple influential factor embedding (Section 3.2) to obtain the factor-specific embedding matrix of the long-term check-in sequence of a user, denoted by $E_\gamma^{\mathcal{L}} \in R^{l \times d} (\gamma \in \{GT, Re, TP\})$, where l is the length of the long-term check-in sequence.

As the general preferences of users do not focus on the order of check-in behaviors as much as the sequential patterns of users, we do not add the position embedding into the factor-specific embedding matrix of the long-term check-in sequence. Then, we respectively apply $E_\gamma^{\mathcal{L}} (\gamma \in \{GT, Re, TP\})$ to (7), (8), (9), (10), and (11) to obtain the multiple general preferences of a user, denoted by $F_\gamma^{\mathcal{L}} (\gamma \in \{GT, Re, TP\})$.

3.3.3 Distinguish of multiple influential factors

Our model uses historical trajectories based on embedding of multiple influential factors to predict the next behavior of a user. However, the user's current historical check-ins have different effects on their next behavior. For example, let $a(home) \rightarrow b(company) \rightarrow c(restaurant) \rightarrow d(movietheater) \rightarrow e(restaurant)$ be a user's historical trajectory. From the perspective of reviews, the user's check-in behavior e may be more related to c , because they all belong to the same category of venues with related review content, but neither the home a nor the company b has any information about reviews. From the perspective of location distance, the current position d may have a greater impact on behavior e of the user. Therefore, it is necessary to pay more attention to the important ones. We use a vanilla attention mechanism to distinguish the different impacts of historical check-ins based on different influential factors on the next behavior of a user. The attention score is calculated by the following formula:

$$\alpha = \text{softmax}\left(\frac{F_{\gamma}^{\epsilon}(h_{\gamma})^T}{\sqrt{d}}\right), \tag{12}$$

where α is the attention weight representing the impact of each check-in in a historical trajectory on the next check-in behavior of a user, and h_{γ} is the feature representation of the target POIs. $\epsilon \in \{S, \mathcal{L}\}$, which represents sequential pattern and general preference.

Then, we perform a weighted summation to obtain the representation h of a historical trajectory as follows:

$$h_{\gamma}^{\epsilon} = \frac{1}{n} \sum_i^n \alpha_i F_{\gamma i}^{\epsilon}, \tag{13}$$

where $F_{\gamma i}^{\epsilon}$ is the latent representation of a check-in of the feed-forward network output, and n is the length of a historical trajectory of a user.

We apply the output of the last feed-forward network layer $F_{GT}^S, F_{Re}^S, F_{TP}^S, F_{GT}^{\mathcal{L}}, F_{Re}^{\mathcal{L}}, F_{TP}^{\mathcal{L}}$ and the corresponding target POI representations h_{GT}, h_{Re}, h_{TP} to (12) and (13) to obtain $h_{GT}^S, h_{Re}^S, h_{TP}^S, h_{GT}^{\mathcal{L}}, h_{Re}^{\mathcal{L}}, h_{TP}^{\mathcal{L}}$.

3.4 Prediction

In the prediction module, we set three parameter matrices, W_{GT}, W_{Re}, W_{TP} , to incorporate both multiple sequential patterns and general preferences based on different influential factors. Thus, we obtain the final sequential patterns and general preferences of user u based on multiple influential factors as follows:

$$h^{\mathcal{L}} = h_{GT}^{\mathcal{L}} W_{GT} + h_{Re}^{\mathcal{L}} W_{Re} + h_{TP}^{\mathcal{L}} W_{TP}, \tag{14}$$

$$h^S = h_{GT}^S W_{GT} + h_{Re}^S W_{Re} + h_{TP}^S W_{TP}. \tag{15}$$

Based on the sequential patterns and general preferences of user u , we can predict the probability that user u will visit POI v at the next time t as follows:

$$y_v^{u,t} = (h^{\mathcal{L}})^T h_v + (h^S)^T h_v, \tag{16}$$

where h_v is obtained by a look-up operation on the pre-trained embedding of POIs.

3.5 Training

We took the venues that the user u actually visited at the next moment as positive samples, and randomly selected the POIs that a user did not visit as negative samples. The predicted probabilities $y_{v_p}^{u,t}, y_{v_n}^{u,t}$ of positive and negative POIs were computed via the forward propagation process of MGCAN. We used Bayesian personalized ranking optimization function [64] to train our model MGCAN as follows:

$$L = - \sum_u \sum_t \sum_{v_p, v_n} \log \sigma(y_{v_p}^{u,t} - y_{v_n}^{u,t}) + \frac{\lambda}{2} \|\Theta\|^2 \tag{17}$$

where σ is the activation function (which in this case is the softmax function), θ is the parameter set used for model training, and λ is the regularization parameter.

4 Experiments

4.1 Experimental settings

4.1.1 Datasets

We evaluated the recommendation performance of the MGCAN model on datasets from Foursquare for two cities, New York (NY) and Chicago (CH), as shown in Table 2. The datasets consisted of the following information: user ID, POI ID, check-in timestamp, location of POIs, and users' reviews of POIs.

For each of the two experimental datasets, we eliminated both the inactive users who had visited fewer than 30 POIs and the unpopular POIs that had been visited by fewer than 30 users. For each user, as shown in Figure 3, we grouped successive check-ins into the same trajectory, among which each pair of successive check-ins has a time interval less than 6 hours, as in the work of Cheng et al. [65]. Further, we removed those trajectories with fewer than five check-ins. Moreover, we defined a subdivision trajectory as a trajectory belonging to the above grouped trajectory that has more than five check-ins. Thus, as shown in Figure 3, if the grouped trajectory of a user had the minimum numbers of check-ins (that is, five), the user had at least three subdivision trajectories. For the two datasets NY and CH, we randomly divided the data of users' trajectories sorted by time into a training set (80%), validation set (10%), and test set (10%).

4.1.2 Baseline model

We compared the recommendation performance of our model MGCAN with those of the following baseline models.

- STRNN [66], a spatial temporal RNN model for predicting next venues, which models the differences of time intervals and geographical distances via spatial–temporal transition matrices.
- TMCA [67] which employs the LSTM method with two attention mechanisms to learn deep spatial–temporal representations. The original model considers category of venue, but in this paper we removed this for a better comparison.
- ATST-LSTM [68], which uses LSTM and an attention method to model spatio-temporal contextual information.
- LSTPM [34], which builds sequences with time intervals and distance interval, and uses LSTM to obtain model long-term and short-term preferences of users for recommendation.

Table 2 Statistics of the datasets

Dataset	User	POI	Check-in	Sparsity
NY	6,206	5,360	169,090	0.9949
CH	2,558	7,824	114,091	0.9943

4.1.3 Evaluation metrics

In this paper, two evaluation metrics, top-k recall rate ($Recall@k$) and normalized discounted cumulative gain ($NDCG@k$), were used to evaluate the performance of models. These metrics have been frequently used in previous work on next POI recommendation [34, 67]. $Recall@k$ measures the proportion of correctly predicted samples in all positive samples. $NDCG@k$ evaluates the gap between the top-k list and the actual list. In our experiments, we set $k = \{5, 10\}$ for evaluation.

4.1.4 Parameter settings

Our MGCAN model was implemented in Python 3.7 with the TensorFlow deep learning framework. We trained our MGCAN model and baselines on a computer server with four NVIDIA GPUs, each of which had 11,178 MiB memory.

In the sequential pattern learning, the length of the short-term historical trajectory was set to 50, and in the general preferences learning, the length of the long-term check-in record was set to 200 in our MGCAN model. We uniformly set the dimension size of the embedding of different influential factors and the hidden layer to 256. In the influence factor embedding module, the geographical relevance level η was set to 60 and the number of the GCNs layers was set to 3. In addition, we used five layers of multi-head self-attention with eight heads and five layers of feed-forward network blocks in the attention module. We chose 500 negative samples for each positive POI for training. The batch size for model training was set to 32. Regarding the gradient descent parameters, the initial learning rate was set to 0.3×10^{-4} , the decay rate is set to 0.96, and the regularization λ is set to 0.1×10^{-4} .

4.2 Results and discussions

4.2.1 Performance comparison

The recommendation performance of our proposed MGCAN model and those of the four baseline models on two real-world datasets are shown in Table 3 and Figure 4.

Our model MGCAN unequivocally outperformed all the other methods on both the NY and CH datasets. For example, for the NY dataset, LSTPM ranked second on accuracy, but MGCAN significantly outperformed LSTPM by 60.50% on $Recall@5$, 57.07% on $NDCG@5$, 29.29% on $Recall@10$, and 79.51% on $NDCG@10$. For the CH dataset, MGCAN significantly outperformed LSTPM by 19.92% on $Recall@5$, 56.14% on $NDCG@5$, 35.02% on $Recall@10$, and 69.02% on $NDCG@10$. In addition, we found that accuracy performance of MGCAN on NY was better than that on CH by 18.86% on $Recall@5$, 19.21% on $NDCG@5$, 21.11% on $Recall@10$, and 20.43% on $NDCG@5$; this was because, on the one hand, the average number of visits per POI in CH was less than that in NY, and, on the other hand, the reviews in CH were sparser than those in NY.

Among the baseline models, we noticed that LSTM-based models such as TMCA and LSTPM outperformed RNN-based ones. For example, for the NY dataset, TMCA outperformed STRNN by 85.53% on $Recall@5$, 110.93% on $NDCG@5$, 69.59% on $Recall@10$, and 80.97% on $NDCG@10$. For the CH dataset, TMCA outperformed STRNN by 95.60% on $Recall@5$, 134.47% on $NDCG@5$, 85.29% on $Recall@10$, and 61.90% on $NDCG@10$.

Table 3 Performance results of MGCAN and baselines

	NY				CH			
	Recall@5	NDCG@5	Recall@10	NDCG@10	Recall@5	NDCG@5	Recall@10	NDCG@10
STRNN	0.1133	0.0814	0.1437	0.1009	0.1068	0.0705	0.1353	0.1105
TMCA	0.2102	0.1717	0.2437	0.1826	0.2089	0.1653	0.2507	0.1789
ATST-LSTM	0.2171	0.1825	0.2171	0.1927	0.2381	0.1831	0.2922	0.2007
LSTPM	0.3681	0.2837	0.4279	0.3031	0.3339	0.2394	0.4200	0.2673
MGCAN-MLP	0.4525	0.4156	0.6543	0.5101	0.3857	0.3550	0.5582	0.4503
MGCAN	0.4759	0.4456	0.6868	0.5441	0.4004	0.3738	0.5671	0.4518
Improvement	29.29%	57.07%	60.50%	79.51%	19.92%	56.14%	35.02%	69.02%

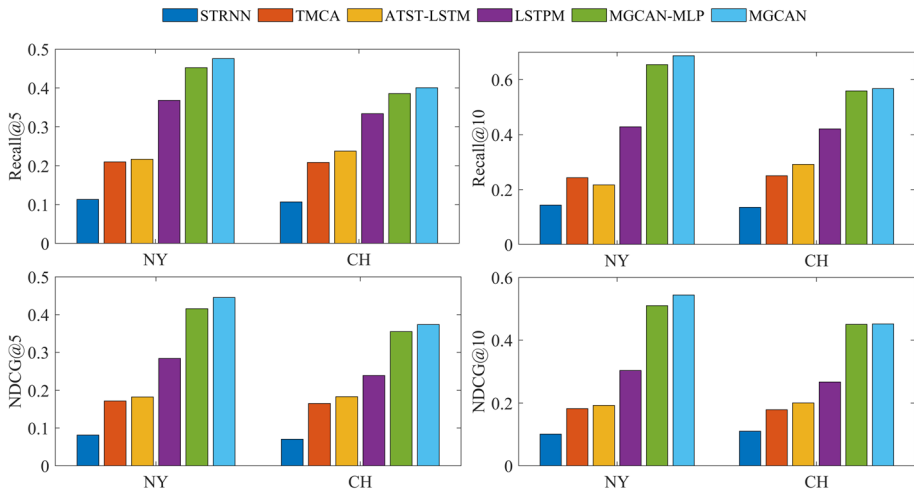


Fig. 4 Performance comparison

This was because the RNN-based models did not capture long-term dependencies in sequences, whereas the improved LSTM models captured long-term dependencies in sequences by controlling states of cell via gate structure. LSTPM had the best performance on the two datasets among the baselines, as it took the long-term and short-term preferences of users into consideration; this demonstrates the importance of modeling long-term preferences as well as short-term preference.

In all baseline models, temporal and geographical factors were considered. Only our model MGCAN took into account other influential factors to model multiple fine-grained sequential patterns and general preferences, enabling it to effectively distinguish the impact of different influential factors on the behavior of a user. The experimental results show that incorporation of various influential factors greatly improved performance. Moreover, in order to analyze the effectiveness of GCN, we constructed a variant of the model called MGCAN-MLP. The MGCAN-MLP model used multilayer perceptron (MLP) instead of GCN to learn the time-popularity embedding; its other components were the same as

those of MGCAN. Comparing the performance of the MGCAN with that of MGCAN-MLP (Table 4), we found that for the NY dataset, MGCAN outperformed MGCAN-MLP by 5.17% on Recall@5, 7.22% on NDCG@5, 4.97% on Recall@10, and 6.68% on NDCG@10. For the CH dataset, MGCAN outperformed MGCAN-MLP by 3.81% on Recall@5, 5.30% on NDCG@5, 1.59% on Recall@10, and 3.79% on NDCG@10. These results demonstrate that GCN captured high-order features to obtain better feature representation, which greatly improved the recommendation performance.

4.2.2 Impact of different influential factors

To analyze the effects of different influential factors on model performance, we designed ablation experiments using different combinations of influential factors in our model. To evaluate the effects of different components of the model, we built a series of model variants incorporating an increasing number of influential factors; thus, the variants of MGCAN were as follows.

None: without any influential factor.

GT: model with the geographical-temporal factor.

GT+R: model with geographical-temporal and review features factors.

GT+R+TP: model with geographical-temporal, review features, and time-popularity factors.

Table 4 shows the performance of our MGCAN model variants with different influential factors.

As shown in Figure 5, we found that as the number of influential factors included in the MGCAN model increased, the performance of the model improved greatly. First, the geographical-temporal factor is important when a user makes a decision. A user will consider location distance and time when traveling, and may be more likely to go to a venue that is close rather than one that is far away. Comparing variants GT and None, we found that the geographical-temporal factor provided a great improvement. For the NY dataset, adding the geographical-temporal factor led to improvements of 97.25%, 80.77%, 105.04%, and 87.86% on Recall@5, NDCG@5, Recall@10, and NDCG@10, respectively. For the CH dataset, adding this factor led to improvements of 43.27%, 38.11%, 39.71%, and 37.00% on Recall@5, NDCG@5, Recall@10, and NDCG@10, respectively. These results demonstrate the effectiveness of the geographical-temporal factor in improving recommendation performance.

Second, some users make decisions by observing others' experiences, and online reviews reflect a user's experience and point of view on a POI. By incorporating the reviews influential factor, our model achieved improvements on Recall@5, NDCG@5, Recall@10, and NDCG@10 of 17.07%, 14.04%, 15.91%, 14.17% for the NY dataset,

Table 4 Performance of MGCAN model variants with different influential factor variants

	NY				CH			
	Recall@5	NDCG@5	Recall@10	NDCG@10	Recall@5	NDCG@5	Recall@10	NDCG@10
None	0.1895	0.1935	0.2715	0.2310	0.2422	0.2472	0.349	0.2965
GT	0.3737	0.3498	0.5566	0.4339	0.3470	0.3114	0.4876	0.4062
GT+R	0.4375	0.3989	0.6452	0.4954	0.3743	0.3413	0.5488	0.4219
GT+R+TP	0.4759	0.4456	0.6868	0.5441	0.4004	0.3738	0.5671	0.4518

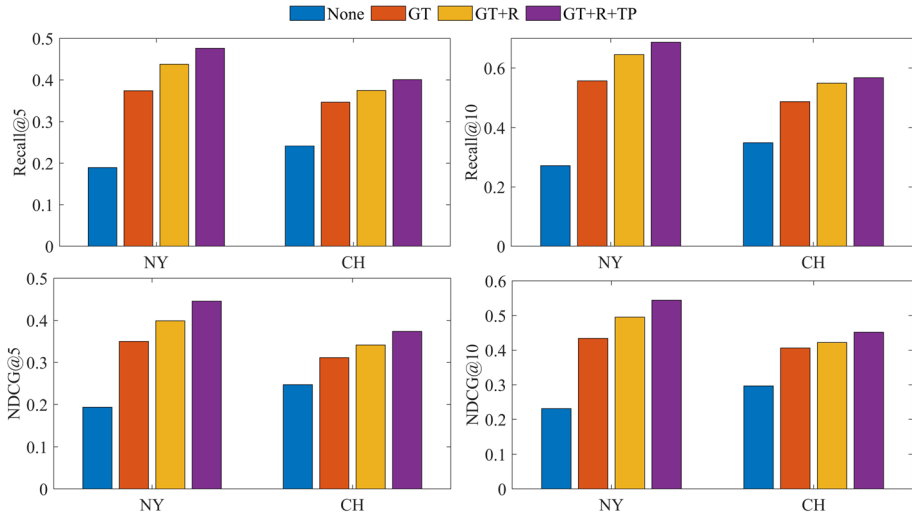


Fig. 5 Performance of different MGCAN variants

respectively. For the CH dataset, Recall@5, NDCG@5, Recall@10, and NDCG@10 were improved by 7.87%, 9.60%, 12.55%, and 3.87%, respectively.

Third, time-popularity is a global attribute of venues that describes the period in which a venue is more popular, for example, restaurants may be more popular at noon, whereas movie theaters are more popular at night. Adding the time-popularity influential factor resulted in a significant improvement in recommendation performance: it improved Recall@5, NDCG@5, Recall@10, and NDCG@10 by 8.78%, 11.71%, 6.46%, and 9.84% on the NY dataset, and by 6.97%, 9.52%, 3.33%, and 7.09%, on the CH dataset, respectively.

Owing to the importance of the geographical–temporal factor, reviews, and time-popularity in decision-making, the recommendation rankings generated by the incorporation of different influential factors can satisfy the interests of users more effectively. Therefore, the recommendation quality is significantly improved.

4.2.3 Analysis on parameter sensitivity

The number of negative samples has an impact on performance of the MGCAN model. To find a suitable number of negative samples, we set a series of numbers of negative samples, $s = \{1, 100, 200, 300, 400, 500, 600\}$, for experimentation. The results of experiments with different numbers of negative samples on the NY and CH datasets are shown in Figure 6. As shown in the figure, when the number of negative samples was greater than 500, the recommendation performance decreased on the NY dataset and tended to be stable on the CH dataset. Therefore, in the MGCAN model, we chose 500 negative samples for training.

The size of the embedding dimensions is an important parameter in our model. The larger the embedding dimensions, the better the representation ability of the model. However, when the embedding dimensions reaches a certain threshold, the model faces the over-fitting problem. Figure 7 shows the performance results when the embedding dimensions of the MGCAN model were 64, 128, 192, 256, and 320 on the NY and CH datasets. The results indicated that when the size of the embedding dimensions was less than 256, the

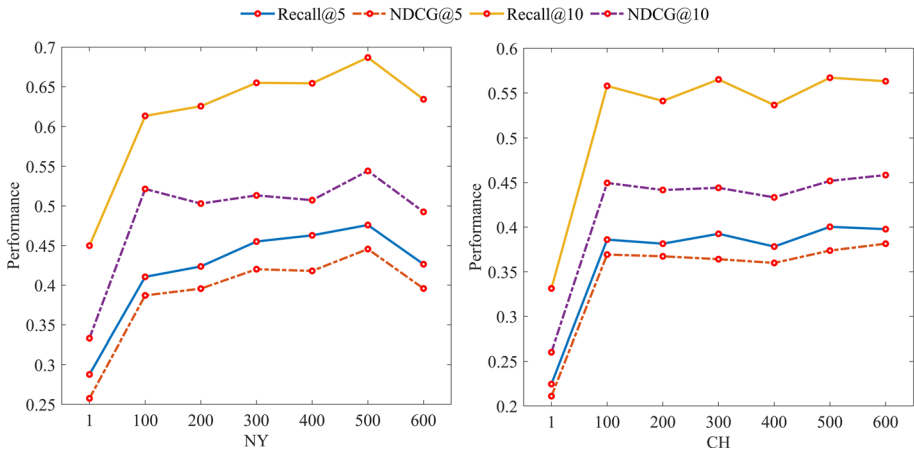


Fig. 6 Impact of negative samples

performance of the MGCAN model significantly increased as the embedding dimension increased; when the size of the embedding dimensions was more than 256, the performance of the model decreased slightly. Therefore, in the MGCAN model, we set the embedding dimensions parameter to 256.

4.2.4 Visualization of influential factors

We weighted the contributions of influential factors and historical check-ins by using attention networks to describe the effects of the geographical-temporal influential factor, review influential factor, and time-popularity influential factor on individual behavior. Figure 8 shows the impact of historical check-ins with different influential factors embedding on current behaviors of users, where (a), (b), and (c), respectively, shows the impact of geographical-temporal, review, and time-popularity factors on users' current behavior. We

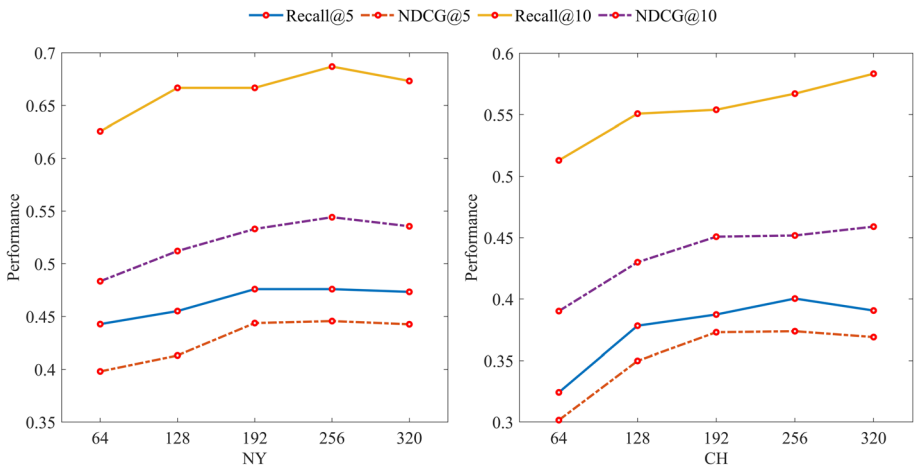


Fig. 7 Impact of embedding dimensions

selected historical trajectories of 30 users and observed the impact of 50 check-ins of each historical trajectory on current behavior of users. In Figure 8, the X-axis represents the historical trajectory of a user, consisting of the 50 latest venues that the user checked into, and the Y-axis represents the 30 users. The attention score of each historical check-in is as

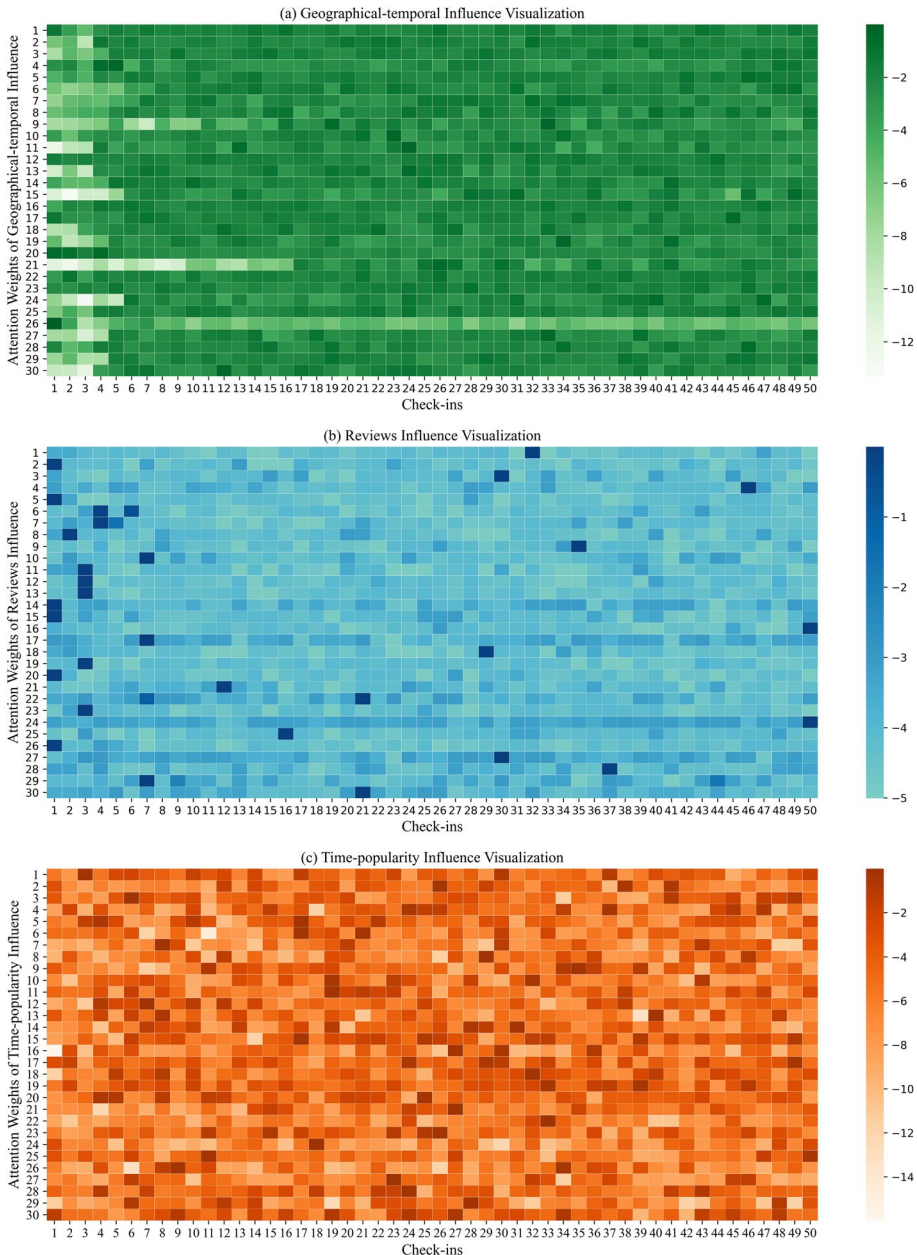


Fig. 8 Attention weights visualization of historical check-ins based on influential factors

expressed by the color of the cell; the darker the color, the greater the weight score. The weight score of each cell represented the impact of each historical check-in of a user on the current check-in behavior of the user. For example, as shown in Figure 8(a), for the 11-th user, the second historical check-in had the greatest impact on current behavior, and the first historical check-in had the second-highest impact, whereas the other check-ins had little impact. As shown in Figure 8(b), the historical check-ins with reviews embedding had different impacts on different users' current behaviors. For the first user, the first historical check-in had a great impact on current behavior, and the other check-ins had little impact, whereas for the third user, the 49-th historical check-in had a great impact and the other check-ins had little impact. Similarly, as shown in Figure 8(c), historical check-ins embedded by time-popularity had different impacts on different users' current behavior.

In order to more intuitively see the impacts of different influential factors on users, we selected the impact of a certain historical check-in of 30 users on the current behavior of the users from the dimension of influential factors, as shown in Figure 9. Similar to Figure 8, the X-axis represents the 30 users, the Y-axis represents the three types of influential factors, and each cell represents the impact of the different influential factors on the current behavior of users. For the first user, the geographical–temporal factor had more impact on behavior. For the second user, the reviews influential factor had more weight than the other influential factors. For the fifth user, the time-popularity influential factor had the greatest effect on behavior. These results demonstrate that different influential factors have different effects on different users. Therefore, recommendation performance could be improved greatly by determining the impact of different influential factors on a user.

5 Conclusion

In this paper, we propose a MGCAN model that uses multiple GCNs and multiple attention networks to incorporate various influential factors in LBSNs, i.e., geographical–temporal influence, influence of reviews, and influence of time-popularity. Specifically, we used GCNs to embed geographical features and time-popularity, and attention networks to capture multiple sequential patterns and general preferences based on different influential factors. Finally, we used sequential patterns and general preferences to predict the next POI. Experimental results on NY and CH datasets demonstrated that the MGCAN model outperformed baseline models. In particular, experiments using different combinations of components demonstrated the impact of different influential factors and the effectiveness

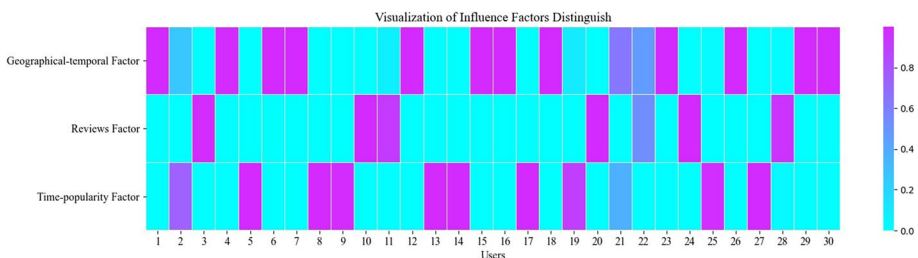


Fig. 9 Visualization of Influential Factors' impact on Current Behavior of Users

of obtaining representation through the neighbor propagation mechanism. In the future, we plan to incorporate more influential factors,

Acknowledgements This research was supported in part by the National Natural Science Foundation of China, under Grants Nos. 71871019, 71471016.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Seo, Y. -D., Cho, Y. -S.: Point of interest recommendations based on the anchoring effect in location-based social network services. *Expert Syst. Appl.* **114018**, 164 (2021)
2. Qiao, S., Han, N., Zhou, J., Li, R. -H., Jin, C., Gutierrez, L. A.: Socialmix: a familiarity-based and preference-aware location suggestion approach. *Eng. Appl. Artif. Intel.* **68**, 192–204 (2018). <https://doi.org/10.1016/j.engappai.2017.11.006>
3. Yang, K., Zhu, J.: Next poi recommendation via graph embedding representation from h-deepwalk on hybrid network. *IEEE Access* **7**, 171105–171113 (2019)
4. Aliannejadi, M., Rafailidis, D., Crestani, F.: A joint two-phase time-sensitive regularized collaborative ranking model for point of interest recommendation. *IEEE Trans. Knowl. Data Eng.* **32**, 1050–1063 (2020)
5. Mingxin, G., Ling, G.: Discovering memory-based preferences for poi recommendation in location-based social networks. *ISPRS International Journal of Geo-Information*, 8 (2019)
6. Wang, W., Chen, J., Wang, J., Chen, J., Gong, Z.: Geography-aware inductive matrix completion for personalized point-of-interest recommendation in smart cities. *IEEE Internet Things J.* **7**, 4361–4370 (2020)
7. Jiao, X., Xiao, Y., Zheng, W., Wang, H., Hsu, C. -H.: A novel next new point-of-interest recommendation system based on simulated user travel decision-making process. *Futur. Gener. Comput. Syst.* **100**, 982–993 (2019)
8. Yu, F., Cui, L., Guo, W., Lu, X., Li, Q., Lu, H.: A category-aware deep model for successive poi recommendation on sparse check-in data. In: *Proceedings of The Web Conference 2020. WWW '20*, pp. 1264–1274. Association for Computing Machinery (2020)
9. Cheng, C., Yang, H., Lyu, M. R., King, I.: Where you like to go next: Successive point-of-interest recommendation. In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI '13*, pp. 2605–2611 (2013)
10. Cui, Q., Tang, Y., Wu, S., Wang, L.: Distance2pre: Personalized spatial preference for next point-of-interest prediction. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 289–301. Springer (2019)
11. Wu, Y., Li, K., Zhao, G., Qian, X.: Long-and short-term preference learning for next poi recommendation. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2301–2304 (2019)
12. Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., Cai, D.: What to do next: Modeling user behaviors by time-lstm. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3602–3608 (2017)
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17*, pp. 6000–6010. Curran Associates Inc. (2017)
14. Kang, W. -C., McAuley, J.: Self-attentive sequential recommendation. In: *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206 (2018)
15. Wu, J., Cai, R., Wang, H.: Déjà vu: a contextualized temporal attention mechanism for sequential recommendation. In: *Proceedings of The Web Conference 2020*, pp. 2199–2209 (2020)
16. Ji, M., Joo, W., Song, K., Kim, Y. -Y., Moon, I. -C.: Sequential recommendation with relation-aware kernelized self-attention. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 4304–4311 (2020)

17. Gao, R., Li, J., Li, X., Song, C., Chang, J., Liu, D., Wang, C.: Stscr: Exploring spatial-temporal sequential influence and social information for location recommendation. *Neurocomputing* **319**, 118–133 (2018)
18. Hosseini, S., Yin, H., Zhou, X., Sadiq, S., Kangavari, M. R., Cheung, N. -M.: Leveraging multi-aspect time-related influence in location recommendation. *World Wide Web* **22**(3), 1001–1028 (2019)
19. Han, H., Zhang, M., Hou, M., Zhang, F., Wang, Z., Chen, E., Wang, H., Ma, J., Liu, Q.: Stgcn: A spatial-temporal aware graph learning method for poi recommendation. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 1052–1057. IEEE (2020)
20. Zhong, T., Zhang, S., Zhou, F., Zhang, K., Trajcevski, G., Wu, J.: Hybrid graph convolutional networks with multi-head attention for location recommendation. *World Wide Web* **23**(6), 3125–3151 (2020)
21. Zhao, P., Luo, A., Liu, Y., Zhuang, F., Xu, J., Li, Z., 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* (2020)
22. Cui, Q., Zhang, C., Zhang, Y., Wang, J., Cai, M.: St-pil: Spatial-temporal periodic interest learning for next point-of-interest recommendation. *arXiv:2104.02262* (2021)
23. Liu, T., Liao, J., Wu, Z., Wang, Y., Wang, J.: Exploiting geographical-temporal awareness attention for next point-of-interest recommendation. *Neurocomputing* **400**, 227–237 (2020)
24. Ye, M., Yin, P., Lee, W. -C., Lee, D. -L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325–334 (2011)
25. Cai, L., Wen, W., Wu, B., Yang, X.: A coarse-to-fine user preferences prediction method for point-of-interest recommendation. *Neurocomputing* **422**, 1–11 (2021)
26. Liu, W., Wang, Z. -J., Yao, B., Yin, J.: Geo-Alm: Poi recommendation by fusing geographical information and adversarial learning mechanism. In: IJCAI, vol. 7, pp. 1807–1813 (2019)
27. Zhang, J. -D., Chow, C. -Y., Li, Y.: Lore: Exploiting sequential influence for location recommendations. In: Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. SIGSPATIAL '14, pp. 103–112. Association for Computing Machinery. <https://doi.org/10.1145/2666310.2666400>(2014)
28. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y. M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
29. Zhao, S., Zhao, T., Yang, H., Lyu, M. R., King, I.: Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI'16, pp. 315–321 (2016)
30. Chang, B., Park, Y., Park, D., Kim, S., Kang, J.: Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. IJCAI'18, pp. 3301–3307. AAAI Press (2018)
31. Zhang, S., Xiong, H., Computer, S. O.: Geo-social-temporal sequential embedding rank for point-of-interest recommendation. *Appl. Res. Comput.* **36**, 2618–2624 (2019)
32. Gan, M., Cui, H.: Exploring user movie interest space: a deep learning based dynamic recommendation model. *Expert Syst. Appl.* **173**, 114695 (2021). <https://doi.org/10.1016/j.eswa.2021.114695>
33. Zhang, J., Mu, X., Zhao, P., Kang, K., Ma, C.: Improving current interest with item and review sequential patterns for sequential recommendation. *Eng. Appl. Artif. Intel.* **104**, 104348 (2021). <https://doi.org/10.1016/j.engappai.2021.104348>
34. Sun, K., Qian, T., Chen, T., Liang, Y., Yin, H.: Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation. *Proc. AAAI Conf. Artif. Intell.* **34**(1), 214–221 (2020)
35. Manotumruksa, J., Macdonald, C., Ounis, I.: A contextual recurrent collaborative filtering framework for modelling sequences of venue checkins. *Inf. Process. Manag.* **57**(6), 102092 (2020)
36. Ma, Y., Gan, M.: Deepassociate: a deep learning model exploring sequential influence and history-candidate association for sequence recommendation. *Expert Syst. Appl.* **115587**, 185 (2021). <https://doi.org/10.1016/j.eswa.2021.115587>
37. Liu, T., Liao, J., Wu, Z., Wang, Y., Wang, J.: A geographical-temporal awareness hierarchical attention network for next point-of-interest recommendation. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. ICMR '19, pp. 7–15. Association for Computing Machinery (2019)
38. Zhao, K., Zhang, Y., Yin, H., Wang, J., Zheng, K., Zhou, X., Xing, C.: Discovering subsequence patterns for next poi recommendation. In: Bessiere, C (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 3216–3222. International Joint Conferences on Artificial Intelligence Organization, Ma (2020)

39. Ren, R., Liu, Z., Li, Y., Zhao, W. X., Wang, H., Ding, B., Wen, J. -R.: Sequential recommendation with self-attentive multi-adversarial network. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 89–98 (2020)
40. Li, Z., Cheng, W., Xiao, H., Yu, W., Chen, H., Wang, W.: You are what and where you are: Graph enhanced attention network for explainable poi recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 3945–3954 (2021)
41. Wang, W. X. X. Z. Y. D. D. S. X. G. D.: Attentive sequential model based on graph neural network for next poi recommendation. *World Wide Web* **24**, 2161–2184 (2021). <https://doi.org/10.1007/s11280-021-00961-9>
42. Xu, Y., Li, X., Li, J., Wang, C., Gao, R., Yu, Y.: Ssser: Spatiotemporal sequential and social embedding rank for successive point-of-interest recommendation. *IEEE Access* **7**, 156804–156823 (2019). <https://doi.org/10.1109/ACCESS.2019.2950061>
43. Liu, Y., Pei, A., Wang, F., Yang, Y., Zhang, X., Wang, H., Dai, H., Qi, L., Ma, R.: An attention-based category-aware gru model for the next poi recommendation. *International Journal of Intelligent Systems* (2021)
44. Zang, H., Han, D., Li, X., Wan, Z., Wang, M.: Cha: Categorical hierarchy-based attention for next poi recommendation. *ACM Trans. Inf. Syst. (TOIS)* **40**(1), 1–22 (2021)
45. Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. *IEEE Signal Process. Mag.* **34**(4), 18–42 (2017)
46. Kipf, T. N., Welling, M.: Semi-supervised classification with graph convolutional networks. [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
47. Marcheggiani, D., Bastings, J., Titov, I.: Exploiting semantics in neural machine translation with graph convolutional networks. [arXiv:1804.08313](https://arxiv.org/abs/1804.08313)(2018)
48. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. [arXiv:1709.04875](https://arxiv.org/abs/1709.04875)(2017)
49. Alex Fout, J.B.S.: Protein interface prediction using graph convolutional networks (2018)
50. Li, S., Pan, X.: A computational drug repositioning model based on hybrid similarity side information powered graph neural network. *Futur. Gener. Comput. Syst.* **125**, 24–31 (2021)
51. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983 (2018)
52. Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H., Chen, G.: Dual Graph Attention Networks for Deep Latent Representation of Multifaceted Social Effects in Recommender Systems. In: The World Wide Web Conference. WWW '19, pp. 2091–2102. Association for Computing Machinery (2019)
53. Qian, T., Liang, Y., Li, Q.: Solving cold start problem in recommendation with attribute graph neural networks. [arXiv:1912.12398](https://arxiv.org/abs/1912.12398) (2019)
54. Ji, Z., Wu, M., Yang, H., Íñigo, J.E.A.: Temporal sensitive heterogeneous graph neural network for news recommendation. *Futur. Gener. Comput. Syst.* **125**, 324–333 (2021)
55. Dong, Y., Chawla, N. V., Swami, A.: Metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '17, pp. 135–144. Association for Computing Machinery (2017)
56. Zhou, C., Bai, J., Song, J., Liu, X., Zhao, Z., Chen, X., Gao, J.: Atrank: an attention-based user behavior modeling framework for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
57. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al: Rectifier nonlinearities improve neural network acoustic models. In: Proc. Icml, vol. 30, p. 3. Citeseer (2013)
58. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196. PMLR (2014)
59. Si, Y., Zhang, F., Liu, W.: An adaptive point-of-interest recommendation method for location-based social networks based on user activity and spatial features. *Knowl.-Based Syst.* **163**, 267–282 (2019)
60. Tang, J., Belletti, F., Jain, S., Chen, M., Beutel, A., Xu, C., Chi, H. E.: Towards neural mixture recommender for long range dependent user sequences. In: The World Wide Web Conference, Pp. 1782–1793 (2019)
61. Zhao, P., Zhu, H., Liu, Y., Li, Z., Xu, J., Sheng, V. S.: Where to go next: A spatio-temporal lstm model for next poi recommendation. [arXiv:1806.06671](https://arxiv.org/abs/1806.06671) (2018)
62. Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J. M., He, X.: A simple convolutional generative network for next item recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 582–590 (2019)
63. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

64. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv:[1205.2618](https://arxiv.org/abs/1205.2618) (2012)
65. Cheng, C., Yang, H., King, I., Lyu, M.: Fused matrix factorization with geographical and social influence in location-based social networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 26 (2012)
66. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 194–200 (2016)
67. Li, R., Shen, Y., Zhu, Y.: Next point-of-interest recommendation with temporal and multi-level context attention. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 1110–1115. <https://doi.org/10.1109/ICDM.2018.00144> (2018)
68. Huang, L., Ma, Y., Wang, S., Liu, Y.: An attention-based spatiotemporal lstm network for next poi recommendation. IEEE Trans. Serv. Comput., 1–1. <https://doi.org/10.1109/TSC.2019.2918310> (2019)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.