Modeling Personalized Short-term and Periodic Long-term Preferences for Enhanced Next POI Recommendations

Mo Li[1], Zhaosong Zhao[1], Linlin Ding[1]✉, Mingyang Ma[2]✉

[1] School of Information, Liaoning University, China
[2] Security Department, Liaoning University, China
dinglinlin@lnu.edu.cn, mmy@lnu.edu.cn

Abstract. Next point-of-interest (POI) recommendation plays a crucial role in enhancing user travel experiences and driving platform revenues by suggesting potentially appealing locations to users. Existing works have focused on capturing users' general preferences and dynamic interests by modeling long-term and short-term check-in sequences. However, current long-term models struggle to accurately capture periodic user behaviors, while short-term models often fail to account for users' personalized geographical preferences within their trajectories. To address these limitations, we propose a novel model: **P**ersonalized **S**hort-term and **P**eriodic **L**ong-term Preferences Modeling Network (PSPL). This model integrates users' short-term spatio-temporal preferences and their long-term periodic location preferences. Specifically, we introduce a S^{2}Graph (**S**patial **S**pan **Graph**) used for GCN to model users' short-term personalized spatial span preferences and devise an ASL Block (Self-**A**ttention and **S**pan **L**STM) to capture spatio-temporal preferences and sequential information. Additionally, we employ a Discrete Fourier Transform (DFT)-based method to effectively capture long-term periodic patterns. The integration of these two types of features significantly enhances the accuracy of next POI recommendations. Extensive experiments on real-world datasets demonstrate the superiority of our model, achieving an average improvement of 6.39% in Recall and 6.54% in NDCG compared to the state-of-the-art methods.

1 Introduction

The proliferation of Location-Based Social Networks (LBSNs) enables users to share their locations, generating a vast dataset of geographical information [1]. This huge amount of data provides unprecedented opportunities to learn user preferences and enhance POI recommendations [2,3]. The Next POI recommendation task aims to generate a ranked list of locations that users are most likely to visit, based on the analysis of their recent check-in trajectories [4]. This task not only optimizes travel arrangements but promotes business expansion.

Existing research primarily falls into two categories: one focuses on modeling user trajectories directly, without distinguishing between temporal preferences [3-14], while the other addresses the separate modeling of users' long-term and short-term preferences [15-18]. In the first approach, researchers employ various models—including RNN-based [5,4], LSTM-based [6,7,8], Transformer-based [9,10,11], and GNN-based [12,13,14,15]—to encode users' spatio-temporal information without categorizing it into short-term and long-term preferences. In the realm of long- and short-term modeling, studies [16,17,18,19] have developed methods using LSTM or attention mechanisms that capture short-term preferences with granularity at hourly intervals and specific areas. These approaches also evaluate spatial-temporal effects and incorporate either a user-based linear combination or attention mechanisms to integrate multiple interests seamlessly.Although the aforementioned models have achieved satisfactory performance, we contend that addressing the following challenges is essential for further enhancing their effectiveness.

Challenge I. Next POI recommendation is significantly influenced by geographical, social, and temporal factors, with geographical influence being perhaps the most critical aspect to consider [2]. Existing research has extensively utilized geographical factors; for instance, [20] capture spatial influence by learning non-linear dependency representations over POIs based on users' historical check-in activities, while [21] model geographical influence using a binary tree structure. Although some recent studies have transformed trajectory distance span information into embeddings to aid sequence modeling [9,14,13], **existing researches predominantly adhere to the assumption that users prefer to visit visited or nearby locations**. However, this overlooks the reality that many individuals frequently choose new and distant locations, possibly because they enjoy driving or live near subway stations. As illustrated in the left part of Fig. 1, the average distance span shows that while users tend to visit nearby POIs, a significant portion also prefers to visit POIs located more than 10 miles away. Additionally, users' distance preferences can vary on different days, as depicted in the right part of Fig. 1. Therefore, reasonably integrating personalized geographical preferences is crucial for the next POI recommendation.

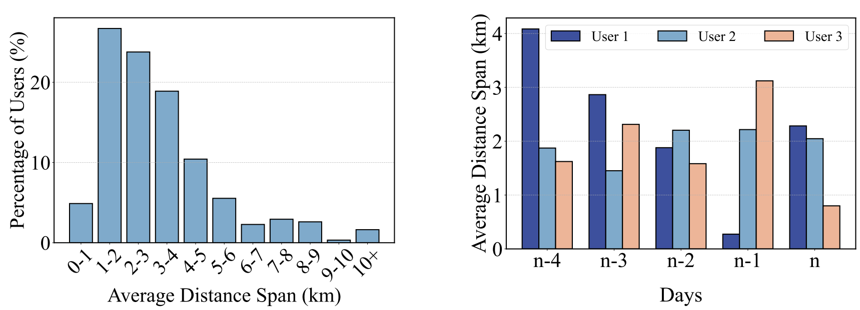

Fig. 1. Distance span statistics and samples of users' trajectories.

Challenge II. **Most models inadequately capture the periodic behaviors that reflect users' long-term preference in POIs**. Human mobility demonstrates long-term periodic regularities; for instance, an employee typically visits the workplace on weekdays and may frequent the gym three times a week. Regrettably, in next POI recommendation, only a few studies have considered these periodicities. Common approaches include segmenting check-ins by weekdays and weekends or on a monthly basis, employing average embeddings to model periodic preferences [1,17,22]. Alternatively, [8] utilizes a Discrete Fourier Series-based periodic attention mechanism to represent users' inherent periodic activities. However, the former fails to recognize that users' periodic behaviors extend beyond weekly or monthly cycles, while the latter imposes strict requirements on input data and cannot accommodate many real-world discrete datasets.

Contributions. To address the aforementioned challenges, we propose a novel framework, **P**ersonalized **S**hort-term and **P**eriodic **L**ong-term Preferences Modeling Network (PSPL), designed to capture both personalized short-term and periodic long-term preferences for next POI recommendation. Specifically, for short-term preference modeling, we first design a new graph construction method to construct S^2Graph (**S**patial **S**pan **Graph**) to capture users' personalized spatial span preferences and use a Graph Convolutional Network (GCN) to incorporate users' distance preferences into the embeddings of POIs. Next, we divide the sequence of users in days and employ our proposed ASL Block (Self-**A**ttention and **S**pan **L**STM) to capture users' short-term spatio-temporal preferences and sequential information, thereby tackling Challenge I. For long-term preference modeling, we use the Discrete Fourier Transform (DFT)-based approach to extract periodic feature extraction and aggregat embeddings of users' sequences, which are processed by ASL Block and take into account the users' spatio-temporal preferences. Such method can incorporate users' periodic preferences profiles when aggregating long-term sequences, thereby tackling Challenge II. Moreover, extensive experiments conducted on three real-world datasets demonstrate the superiority of our approach in next POI recommendations.

Our main contributions are summarized as below:

- We formulate a novel S^2Graph that reflects users' personalized distance span preferences, and design a new module, the ASL Block, to capture users' spatio-temporal spanning preferences.
- We devise a novel PSPL model for next POI recommendation tasks, which taking into account both users' personalized short-term distance span preferences by using GCN with S^2Graph and ASL BLock, and periodic long-term preferences by using DFT-based method.
- Extensive experiments on three real-world datasets demonstrate the superiority of our model, achieving average improvements of 6.39% in Recall and 6.54% in NDCG compared to state-of-the-art models.

2 Problem Definition

We consider a LBSN comprising a set of users, denoted as $\mathcal{U} = \{u_1, u_2, ..., u_M\}$, and a set of POIs, $\mathcal{L} = \{l_1, l_2, ..., l_N\}$, where M and N represent the total numbers of users and POIs, respectively. Each POI $l \in \mathcal{L}$ corresponds to a real-world geographical location, identified by a unique pair of longitude and latitude coordinate (lon, lat). To better account for temporal variations, we discretize a day into 24 hourly slots, denoted as $\mathcal{T} = \{t_1, t_2, \cdots, t_{24}\}$, and categorize days into weekdays and weekends, represented by $\mathcal{D} = \{d_0, d_1\}$. These classifications allow us to formally define the check-in record and trajectory of a user.

Definition 1 (Check-in Record). *We denote a check-in record of a user as a 4-tuple $c = (u, l, t, d)$, indicating the user u visited POI l at time t on day type d (weekday or weekend).*

Definition 2 (User's Sequence). *We define the sequence of a user as their check-in trajectory, represented as $S(u) = \{c_1, c_2, ..., c_{|S|}\}$, where $|S|$ denotes the number of check-ins of the trajectory of user u. To more effectively capture the periodicity of user's sequence, we segment it into daily sequence, i.e., $S_d(u) = \{S_1^{day}, S_2^{day}, ..., S_n^{day}\}$, where n is the total number of days.*

Definition 3 (Short-Term Sequence). *The short-term sequence of user u is a part of $S(u)$, denoted as $S^u_{short} = \{c_1, c_2, ..., c_m\}$, where m denotes the number of latest locations visited by the user which is highly responsive to the latest activity of user u.*

Problem 1 (Next POI Recommendation). Given a target user u and her/his sequence $S(u)$, our next POI recommendation problem aims to recommend the most likely top-k POI that the user will visit at next timestamp.

3 Methodology

In this section, we provide a detailed explanation of the PSPL model. As illustrated in Fig. 2, the overall framework consists of four main components: i) Personalized Spatial Span GCN, employing a novel S^{2}Graph constructed from users' short-term sequences to update POI representations via GCN; ii) Spatio-Temporal Sequence Encoder, designed to encode users' check-in sequences using the ASL Block; iii) Long- and Short-Term Preference Aggregation, which builds users' long- and short-term representations using the ASL Block and DFT-based aggregation, respectively; and iv) Prediction and Optimization, aiming to predict users' next visit preferences and optimize model parameters.

3.1 Personalized Spatial Span GCN

In this subsection, we explain the rationale behind using an S^{2}Graph to model users' movements and detail the construction of the S^{2}Graph. Additionally, we describe how the GCN encoder is utilized to encode POIs.

Construction of S^{2}Graph. As illustrated in Fig. 1, users often show varied preferences for spatial distances, which can differ significantly among individuals. Additionally, the same user may exhibit different distance preferences at various times. Existing research often focuses on geographical distances or similarities between POIs [14]. However, these studies typically overlook scenarios in which users, particularly those with cars, might prefer visiting distant POIs rather than those near their location. Furthermore, while some studies have considered distance span information [15,14], they fail to recognize the importance of short-term distance preferences or to incorporate this information into the

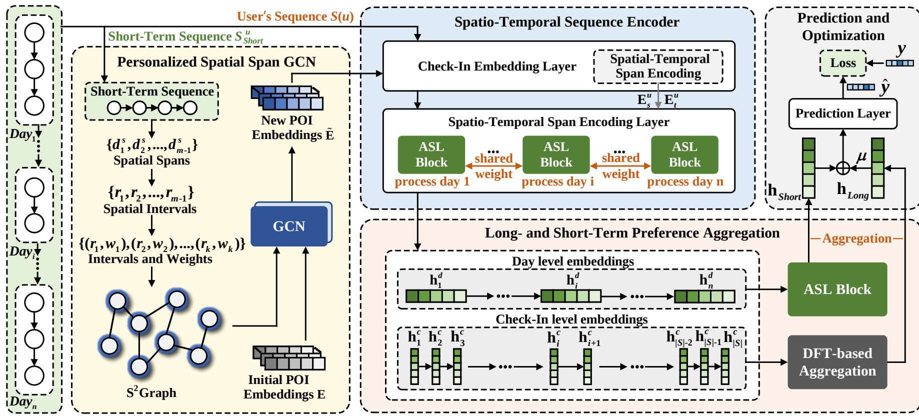

Fig. 2. The overview of our proposed PSPL model.

graph construction process. To address these issues, we propose a novel S^2Graph construction method, which leverages the user's short-term sequence to capture their individual distance preferences.

The S^2Graph is constructed in four steps as follows. **Step 1:** Based on user's short-term sequence $S^u_{short} = \{c_1, c_2, ..., c_m\}$, we first compute the spatial spans for user u at each time the location shifts and note them as $D^u_s = \{d^s_1, d^s_2, \ldots, d^s_{m-1}\}$, where $d^s_i = haversine(l_{i+1}, l_i)$ denotes the distance between l_{i+1} and l_i calculated by haversine formula. **Step 2:** Given a specified interval length γ, a hyperparameter, we generate several spatial intervals $\{r_1, r_2, \ldots, r_{last}\}$ such that the last interval, r_{last}, contains the maximum value of the spatial spans, i.e., $max(d^s_1, d^s_2, \ldots, d^s_{m-1}) \in r_{last}$. Upon determining all intervals, we construct a set of corresponding intervals $R^u_s = \{r_1, r_2, \ldots, r_{m-1}\}$ for user u, where each d^s_i falls within r_i. **Step 3:** We calculate the number of occurrences for each intervals in R^u_s and assign weights accordingly to derive a set of distinct intervals along with their respective weights $W^u_s = \{(r_1, w_1), (r_2, w_2), \ldots, (r_k, w_k)\}$, where k is number of different intervals and the weight w_i can be formulated as:

$$w_i = \frac{count\{R^u_s(r_i)\}}{m-1} \tag{1}$$

Example 1. Given $S^u_{short} = \{c_1, c_2, ..., c_5\}$ and a specified interval length $\gamma = 0.5$, assume the spatial span is $D^u_s = \{0.2, 0.3, 0.4, 1.2\}$, where the maximum distance is 1.2. This maximum distance allows us to define the spatial intervals as $\{(0, 0.5], (0.5, 1], (1, 1.5]\}$, with 1.2 falling within the last interval $(1, 1.5]$. Consequently, R^u_s is determined as $\{(0, 0.5], (0, 0.5], (0, 0.5], (1, 1.5]\}$. Based on the frequency of occurrences within these intervals, W^u_s is then calculated as $\{((0, 0.5], 0.75), ((0.5, 1], 0.25)\}$, where each weight reflects the relative frequency within R^u_s.

Step 4: Based on W^u_s, S^2Graph $\mathbf{A}_S \in \mathbb{R}^{N\times N}$ is calculated. Here, $\mathbf{A}_S(i, j) = w_k$ if the distance between POIs l_i and l_j falls within the interval r_k, that is, $\mathbf{A}_d(i, j) \in r_k$; otherwise $\mathbf{A}_S(i, j)$ is set to 0. The distance matrix $\mathbf{A}_d \in \mathbb{R}^{N\times N}$ records the distance between different POIs, where $\mathbf{A}_d(i, j) = haversine(l_i, l_j)$.

GCN with S^{2}Graph. To obtain a POI representation that better reflects user preferences, we perform graph convolution operations on the S^{2}Graph to update the embedding representations of the POIs.

We first initialize a trainable embeddings $\mathbf{E} \in \mathbb{R}^{N \times d_l}$ for all POIs $\mathcal{L} = \{l_1, l_2, ..., l_N\}$, where d_l denotes the dimension of the location embedding. We next perform the GCN operation using S^{2}Graph:

$$\mathbf{E}^{l+1} = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{A}_S \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{E}^l) \tag{2}$$

where $\mathbf{E}^0$ is $\mathbf{E}$, σ denotes the activation function, $\hat{\mathbf{D}}$ is the degree matrix of $\mathbf{A}_S$.

After performing l-layer GCN, $\{\mathbf{E}^0, \mathbf{E}^1, \ldots, \mathbf{E}^l\}$ is obtained. To alleviate the over-smoothing problem and maintain features captured from different order of neighbors, layer aggregation strategy is applied to obtain new embeddings:

$$\tilde{\mathbf{E}} = agg(\mathbf{E}^0, \mathbf{E}^1, \ldots, \mathbf{E}^l) \tag{3}$$

To avoid the redundant model parameter, aggregation function is light but effective sum-pooling function and the final POIs' embeddings $\tilde{\mathbf{E}} = \{\mathbf{e}_1^l, \mathbf{e}_2^l, \ldots, \mathbf{e}_N^l\}$ is obtained, where $\mathbf{e}_i^l$ is embedding of l_i.

3.2 Spatio-Temporal Sequence Encoder

In this subsection, we will encode spatio-temporal features and sequence order features for the check-in sequence. The spatial and temporal spans of the check-in sequence are first extracted and encoded into embeddings. Then the feature encoding of the check-in sequence is performed using our proposed ASL Block.

Check-In Embedding Layer. To effectively capture user check-in behavior, it is crucial to consider not only the users and POIs, but also contextual factors such as the time of day and whether the check-in occurs on a weekend. Inspired by [19], to model these patterns, we use the embedding layer to encode the user, location, time and day of week into embeddings as $\mathbf{e}^u \in \mathbb{R}^{d_u}$, $\mathbf{e}^l \in \mathbb{R}^{d_l}$, $\mathbf{e}^t \in \mathbb{R}^{d_t}$ and $\mathbf{e}^d \in \mathbb{R}^{d_d}$, where d_u, d_l, d_t and d_d are the dimensions of the above embeddings. Then, we concat them to represent a check-in $\mathbf{e} \in \mathbb{R}^d$ where $\mathbf{e} = [\mathbf{e}^u; \mathbf{e}^l; \mathbf{e}^t; \mathbf{e}^d]$ and $d = d_u + d_l + d_t + d_d$.

With the user check-in sequence $S(u)$, using the above method we can obtain the embeddings of the sequence, denoted as $\mathbf{H}_u = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{|S|}\}$ where $\mathbf{H}_u \in \mathbb{R}^{|S| \times d}$. To more clearly illustrate the operations in the model, we represent $\mathbf{H}_u$ in another form $\{\mathbf{E}_{day\_1}, \mathbf{E}_{day\_2}, ..., \mathbf{E}_{day\_n}\}$ which is divided in days.

Spatio-Temporal Span Encoding. Spatio-temporal context has a significant impact on the user's next visit decision, and explicitly modeling such factors has been shown to be useful in capturing user preferences [9,14,13], so we take a similar approach. Given a user's sequence $S(u) = \{c_1, c_2, ..., c_{|S|}\}$, we calculate the spatial span and temporal span for each position change, which denote as:

$$s_i = min(\Delta s, \lfloor haversine(l_i, l_{i-1}) \rfloor) \tag{4}$$

$$t_i = min(\Delta t, \lfloor \frac{|t_i - t_{i-1}|}{t_{min}} \rfloor) \tag{5}$$

where $t_{\min}$ represents the minimum time interval for user location transitions, Δs and Δt are thresholds to prevent excessively large intervals which is no point in modeling features. Then we generate embeddings $\mathbf{e}_i^s \in \mathbb{R}^d$ and $\mathbf{e}_i^t \in \mathbb{R}^d$ for each s_i and t_i, we can obtain embeddings $\mathbf{E}_s = \{\mathbf{e}_1^s, \mathbf{e}_2^s, \ldots, \mathbf{e}_{|S|}^s\}$ and $\mathbf{E}_t = \{\mathbf{e}_1^t, \mathbf{e}_2^t, \ldots, \mathbf{e}_{|S|}^t\}$, where $\mathbf{e}_1^s$ and $\mathbf{e}_1^t$ are 0 vectors as padding. In addition, to more clearly illustrate the operations in our model, we represent $\mathbf{E}_{s,t}$ in another form $\{\mathbf{E}_{day\_1}^{s,t}, \mathbf{E}_{day\_2}^{s,t}, ..., \mathbf{E}_{day\_n}^{s,t}\}$ which is divided in days.

ASL Block. One typical feature of LBSNs is data sparsity, where each user's check-in sequence is limited in length and not necessarily continuous in time. If there are gaps in a user's check-in sequence where no data is recorded, the check-in records before and after these gaps are likely to have a smaller correlation. If this situation is ignored and the sequence is directly modeled as a feature, some unreasonable features will be extracted. To avoid this situation, we divide the user sequence into several sub-samples by days. We then extract spatio-temporal features and sequence order features using our proposed ASL Block.

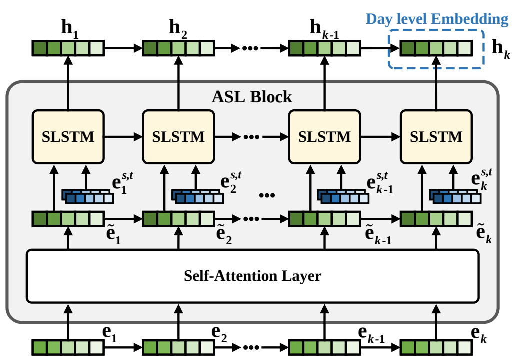

Fig. 3. Structure of ASL Block

The structure of the ASL Block is shown in Fig. 3 and below is the motivation and implementation details of it. To clarify the operational details, we use $\mathbf{E}_{day} \in \mathbb{R}^{k\times d}$ to represent the embeddings of a check-in sequence on a specific day supposing there are k check-ins on this day, and $\mathbf{E}_{day}^{s,t} \in \mathbb{R}^{k\times d}$ to represent the spatio-temporal span embeddings corresponding to the sequence. First, we apply self-attention, which is an approach commonly used in sequence encoding, to enable the sequence elements to interact and capture interrelated features, which can be formulated as:

$$\tilde{\mathbf{E}}_{day} = softmax(\frac{\mathbf{E}_{day}\mathbf{W}_s^Q(\mathbf{E}_{day}\mathbf{W}_s^K)^T}{\sqrt{d}})(\mathbf{E}_{day}\mathbf{W}_s^V) \tag{6}$$

where $\tilde{\mathbf{E}}_{day} = \{\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \ldots, \tilde{\mathbf{e}}_k\}$, and $\mathbf{W}_s^Q$, $\mathbf{W}_s^K$ and $\mathbf{W}_s^V$ is trainable parameters.

Next, we construct sequential features to capture the order of the sequence. The LSTM unit demonstrates limitations in capturing long sequences but performs well with short sequences. Typically, daily sequences are not excessively long, making LSTM well-suited for handling them. However, LSTM has inherent deficiencies in capturing spatial and temporal spans. To address this issues, we propose a novel Spatio-Temporal **S**pan **LSTM** (SLSTM), which aggregates spatio-temporal span information of sequence. Specifically, $\tilde{\mathbf{E}}_{day}$, $\mathbf{E}^s_{day}$, and $\mathbf{E}^t_{day}$ are used as inputs to enhance the LSTM's ability to capture sequence features. It's computational methodology is as follows:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \tilde{\mathbf{e}}_t, \mathbf{e}^s_t, \mathbf{e}^t_t] + \mathbf{b}_i), \\ \mathbf{f}_t &= \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \tilde{\mathbf{e}}_t, \mathbf{e}^s_t, \mathbf{e}^t_t] + \mathbf{b}_f), \\ \tilde{\mathbf{C}}_t &= \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \tilde{\mathbf{e}}_t, \mathbf{e}^s_t, \mathbf{e}^t_t] + \mathbf{b}_C), \\ \mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t, \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \tilde{\mathbf{e}}_t, \mathbf{e}^s_t, \mathbf{e}^t_t] + \mathbf{b}_o), \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \end{aligned} \tag{7}$$

where $\mathbf{W}_*$ and $\mathbf{b}_*$ are trainable parameters, σ is activation function, and $\odot$ is the Hadamard Product. After processing $\tilde{\mathbf{E}}_{day}$, $\mathbf{E}^s_{day}$, and $\mathbf{E}^t_{day}$ by SLSTM, the day level output $\mathbf{h}_k$ and check-in level output $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_k\}$ are computed as:

$$\mathbf{h}_k = \overrightarrow{\text{ASL Block}}(\mathbf{E}_{day}, \mathbf{E}^s_{day}, \mathbf{E}^t_{day}) \tag{8}$$

$$\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_k\} = \overrightarrow{\text{ASL Block}}_{all}(\mathbf{E}_{day}, \mathbf{E}^s_{day}, \mathbf{E}^t_{day}) \tag{9}$$

Spatio-Temporal Sequence Encoding Layer. To encode user's sequence effectively, we employ ASL Block to process the sets $\{\mathbf{E}_{day\_1}, ..., \mathbf{E}_{day\_n}\}$ and $\{\mathbf{E}^{s,t}_{day\_1}, ..., \mathbf{E}^{s,t}_{day\_n}\}$ on a day-by-day basis. After that, we can obtain all day level embeddings $\mathbf{H}_{day} = \{\mathbf{h}^d_1, \mathbf{h}^d_2, \ldots, \mathbf{h}^d_n\}$ and all check-in level embeddings $\mathbf{H}_{check-in} = \{\mathbf{h}^c_1, \mathbf{h}^c_2, \ldots, \mathbf{h}^c_{|S|}\}$ where $\mathbf{H}_{check-in} \in \mathbb{R}^{|S|\times d}$, $\mathbf{H}_{day} \in \mathbb{R}^{n\times d}$ and n denotes the number of days included in the user's sequence.

3.3 Long- and Short-Term Preference Aggregation

In this subsection, we design an ASL Block to aggregate user's day level embeddings, effectively capturing recent span preferences, and leverage the DFT-based method to model user's check-in level embeddings, enabling the capture of long-term periodic preferences.

ASL-based Short-Term Preference Aggregation. In order to further enhance the capture of user spatio-temporal preferences and sequence information, we use ASL Block to aggregate day level embeddings $\mathbf{H}_{day}$, because it is usually not very long, which can be effectively aggregated by ASL Block. Then we can obtain the short-term embedding of the user:

$$\mathbf{h}_{Short} = \overrightarrow{\text{ASL Block}}(\mathbf{H}_{day}, \mathbf{E}^s_{span}, \mathbf{E}^t_{span}) \tag{10}$$

where $\mathbf{E}^s_{span}$ and $\mathbf{E}^t_{span}$ are the embeddings set of the user's check-in span on different days, i.e., the spatial and temporal span of the last check-in record of the previous day and the first check-in record of the next day.

DFT-based Long-Term Preference Aggregation. In POI recommendation tasks, users may exhibit not only recent preferences but also periodic patterns. Research [8] utilized Discrete Fourier Series (DFS) to capture the periodic patterns for next POI recommendation. However, DFS is limited to strictly periodic signals and may not effectively handle finite-length data. In contrast, Discrete Fourier Transform (DFT) leverages its implicit periodic extension to capture periodic characteristics and has an advantage in terms of computational speed while accommodating non-periodic inputs. This makes DFT more suitable for analyzing real-world data with underlying periodic patterns; thus, we use DFT to capture the periodic preferences present in the user sequence. DFT converts a sequence x of length N into a sequence X of the same length N. The transformation is given by the formula:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi}{N}kn} \tag{11}$$

where X_k denotes the k-th element in X, j is imaginary unit.

Using the concept of DFT, to capture periodic user preferences based on sequence $S(u)$, we can define the probability that the next location visited by the user is the location visited before time period τ_i as follows:

$$p(l_{|S|} = l_i|\tau_i) = \sum_{n=1}^{|S|} s_n \cdot e^{-j\frac{2\pi}{N}\tau_i^{'}n} \tag{12}$$

where $\tau_i = t_{|S|} - t_i$, t_i and l_i is the time and location of i-th check-in in $S(u)$, $\tau_i^{'} = t_i - t_1$, and s_n is 1 if $l_n = l_{|S|}$ otherwise 0.

Based on the user's periodic preferences, we estimate the probability that the user's next visit will be a POI visited before τ_i, and then using this probability score, we compute the periodic attention weight, which calculated as follows:

$$p(l_{|S|+1} = l_i|\tau_i) = p(l_{|S|} = l_{i-1}|\tau_{i-1}) \tag{13}$$

$$\beta_{t_i} = p(l_{|S|+1} = l_i|\tau_i) \tag{14}$$

$$\gamma_i = \frac{\exp(\beta_{t_i})}{\sum_{t\in\{t_1,\ldots,t_{|S|}\}} \exp(\beta_t)} \tag{15}$$

where β_{t_i} is the weight of time t_i in sequence, γ_i is normalized weight.

With the obtained weights, we aggregate the check-in level embeddings of user's sequence, $\mathbf{H}_{check-in}$, to derive the long-term periodic preferences:

$$\mathbf{h}_{Long} = \sum_{i=1}^{|S|} \gamma_i \cdot \mathbf{h}_i^c \tag{16}$$

3.4 Prediction and Optimization

To effectively combine the user's short-term and long-term preference embeddings, we introduce an aggregation weight to merge the two embeddings, and then we obtain the user's aggregated embedding representation:

$$\mathbf{h}^u = \mathbf{h}_{Short} + \mu\mathbf{h}_{Long} \tag{17}$$

where μ is a hyperparameter which adjust the aggregation weight of long-term embedding. We then calculate the visiting probability for N different POIs:

$$\hat{y} = \mathrm{softmax}(\mathbf{h}^u \mathbf{W}^N) \tag{18}$$

where $\hat{y} \in \mathbb{R}^N$ represents the predicted probabilities of the user visiting each of the N POIs, and $\mathbf{W}^N \in \mathbb{R}^{d \times N}$ is a learnable matrix.

Given a training sample, the cross-entropy loss function is employed to measure the mismatch between prediction result $\hat{y}$ and ground truth label y, which is defined as:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{19}$$

where y_i equals to 1 if l_i is the real visit of user u at the i-th step and 0 otherwise.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our model on three datasets collected from two real-world LBSNs platforms: Gowalla[3] and Foursquare[4]. The Gowalla dataset contains users' check-in records from February 2009 to October 2010. The Foursquare dataset includes two subsets, one from Phoenix (PHO) and the other from New York City (NYC). The PHO subset is collected from April 2012 to September 2013, while the NYC subset spans from April 2012 to February 2013. Detailed statistics of the datasets are provided in Table 1.

Table 1. Statistics of datasets

Datasets	Users	POIs	Check-in	Avg.Visit
PHO	2,573	6,559	43,018	16.71
NYC	3,506	10,459	132,562	37.81
Gowalla	8,247	25,831	387,127	46.94

We sort users' interaction records chronologically in each dataset. Following the approach of [6,19], POIs with fewer than 10 interactions and trajectories with fewer than three check-ins are filtered out. Additionally, users with fewer than five trajectories are excluded to ensure sufficient activity levels. The remaining check-in records are then split into training, validation, and test sets with a ratio of 8:1:1, which is a common practice in next POI recommendation tasks.

Baselines. To show the effectiveness of our method, we compare our PSPL with following methods, including the RNN-based method (ST-RNN [5]), LSTM-based methods (STGN [7], LSTPM [6]), attention-based methods (STAN [9], CLSPRec [19]) and GNN-based methods (DRAN [12], STHGCN [13], ImNext [15]).

[3] https://sites.google.com/site/yangdingqi/home
[4] http://snap.stanford.edu/data/loc-gowalla.html

Metrics. We employ standard evaluation metrics commonly used in recommender systems, including Recall at cutoff k (Recall@k) and Normalized Discounted Cumulative Gain at cutoff k (NDCG@k). Recall@k measures the fraction of times the target POI is included among the top-k recommendations, focusing on the presence of positive items. NDCG@k, on the other hand, emphasizes the ranking position of positive items, rewarding methods that rank relevant POIs higher within the top-k list. Each metric is calculated 10 times, and the averaged results are reported for $k \in \{1, 5, 10\}$.

Implementation Details. We conduct experiments on a Windows machine with an Intel Xeon Platinum 8163 CPU and a NVIDIA Tesla V100 GPU. In our experiments, the dimensions for user, POI, time, date, and hidden layer, denoted as d_u, d_l, d_t, d_d, and d, are set to 32, 64, 16, 16, and 128, respectively, for the PHO dataset, while they are set to 64, 128, 32, 32, and 256 for the NYC and Gowalla datasets. The distance span interval length is set to 0.5 km, and the number of GCN layers is set to 1. The dropout rate is set to 0.4 for PHO and 0.2 for NYC and Gowalla. The aggregation weight of long-term embedding is set to 1 for all datasets.

4.2 Overall Performance Comparison

Table 2. Performance of the models on the PHO, NYC and Gowalla datasets compared based on the Recall and NDCG metrics, highlighting the best results in bold, and underlining the second best results.

Dataset	Metric	Models									Improve
		ST-RNN	STGN	LSTPM	STAN	DRAN	CLSPRec	STHGCN	ImNext	**Ours**	
PHO	Recall@1	0.1309	0.1752	0.1960	0.2162	0.2110	0.2305	0.2487	0.2267	**0.2695**	+8.36%
	Recall@5	0.2794	0.3641	0.3841	0.4337	0.4558	0.5263	0.5302	0.4951	**0.5712**	+7.73%
	Recall@10	0.3216	0.4509	0.4553	0.5630	0.5732	0.6316	0.6211	0.5833	**0.6588**	+4.31%
	NDCG@1	0.1309	0.1752	0.1960	0.2162	0.2110	0.2305	0.2487	0.2267	**0.2695**	+8.36%
	NDCG@5	0.2167	0.2894	0.2954	0.3415	0.3462	0.3765	0.4055	0.3749	**0.4345**	+7.15%
	NDCG@10	0.2421	0.3177	0.3255	0.3703	0.3861	0.4104	0.4239	0.4018	**0.4523**	+6.70%
NYC	Recall@1	0.1318	0.1874	0.2075	0.2240	0.2318	0.2538	0.2574	0.2520	**0.2764**	+7.38%
	Recall@5	0.2676	0.3414	0.3512	0.4251	0.4363	0.4894	0.5003	0.5146	**0.5480**	+6.49%
	Recall@10	0.3017	0.4100	0.4139	0.5274	0.5249	0.5670	0.5728	0.5805	**0.6152**	+5.98%
	NDCG@1	0.1318	0.1874	0.2075	0.2240	0.2318	0.2538	0.2574	0.2520	**0.2764**	+7.38%
	NDCG@5	0.2030	0.2873	0.2876	0.3253	0.3357	0.3761	0.3761	0.3904	**0.4149**	+6.28%
	NDCG@10	0.2305	0.3180	0.3273	0.3704	0.3762	0.4045	0.4086	0.4122	**0.4330**	+5.05%
Gowalla	Recall@1	0.1073	0.1341	0.1407	0.1635	0.1660	0.1893	0.1940	0.1992	**0.2085**	+4.67%
	Recall@5	0.1533	0.2254	0.2466	0.3132	0.3168	0.3447	0.3492	0.3570	**0.3773**	+5.69%
	Recall@10	0.2162	0.2854	0.3070	0.3638	0.3627	0.4026	0.4135	0.4265	**0.4561**	+6.94%
	NDCG@1	0.1073	0.1341	0.1407	0.1635	0.1660	0.1893	0.1940	0.1992	**0.2085**	+4.67%
	NDCG@5	0.1365	0.1698	0.1903	0.2249	0.2195	0.2705	0.2732	0.2801	**0.2955**	+5.50%
	NDCG@10	0.1547	0.1833	0.2082	0.2571	0.2596	0.2840	0.2860	0.3029	**0.3264**	+7.76%

To validate the effectiveness of our PSPL, we compare it's performance against baseline models across three datasets. Table 2 presents the recommendation performance of all models on the PHO, NYC, and Gowalla datasets. Based on these results, we provide the following observations and analyses.

- It is evident that our proposed PSPL model outperforms all baselines across all evaluation metrics and datasets. Notably, PSPL achieves an average improvement of 6.80%, 6.62%, and 5.77% in Recall@k ($k \in 1, 5, 10$) on PHO, NYC, and Gowalla, respectively. Similarly, PSPL shows an average improvement of 7.40%, 6.24%, and 5.98% in NDCG@k ($k \in 1, 5, 10$) on PHO, NYC, and Gowalla, respectively.

- We attribute the performance gains of our model to several key factors: (1) The Personalized Spatial Span GCN effectively captures users' recent spatial span preferences, which are crucial for accurately predicting their next POI; (2) The ASL Block efficiently extracts sequential features and spatial-temporal spans from users' sequences while mitigating the negative impact of sequence discontinuities; (3) The DFT-based aggregation is employed to capture users' long-term periodic preferences, addressing the limitations of short-term features in recognizing periodic behaviors.
- Attention-based methods outperform RNN-based and LSTM-based methods due to the superior capability of the self-attention mechanism in modeling sequence dependencies. Among attention-based methods, CLSPRec employs a unified attention-based model to capture users' long- and short-term preferences, and applies contrastive learning making it more competitive than STAN, which models spatio-temporal information without considering long- and short-term preferences. However, these methods tend to overlook the combination of periodic behavior patterns and spatial span preferences.
- The two most competitive baselines are ImNext and STHGCN. ImNext uses GAT to incorporate irregular intervals in the check-in sequence, while STHGCN utilizes hypergraph convolution to extract patterns from both individual historical and collaborative user sequences. These observations highlight the effectiveness of GNN-based representation learning and underscore the importance of considering irregular spatial and temporal intervals in users' sequences. However, both models overlook periodic behavior patterns and the synthesis of long- and short-term preferences.

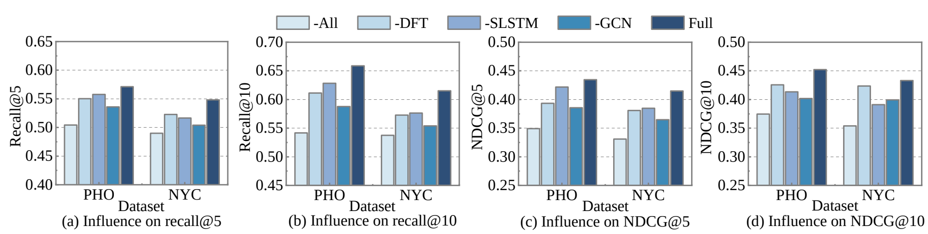

Fig. 4. Ablation results.

4.3 Ablation Analysis

To evaluate the effectiveness of the main components in PSPL, we conducted an ablation study by creating variants of our model with specific modifications as follows: (1) -GCN: the Personalized Spatial Span GCN module is removed; (2) -SLSTM: the SLSTM in ASL Block is replaced with original LSTM; (3) -DFT: the DFT aggregation layer is replaced with an attention-based aggregation layer; (4) -All: simultaneously removing GCN, SLSTM, and DFT. As shown in Fig. 4, we can draw the following conclusions:

- Removing the Personalized Spatial Span GCN module significantly decreases performance, indicating the importance of capturing users' recent span preferences from POI embeddings. These preferences largely reflect the user's current state, enhancing next POI prediction accuracy.
- Replacing our proposed SLSTM with traditional LSTM also results in a performance drop, demonstrating that SLSTM better captures the spatio-temporal features in user sequences.
- Eliminating the DFT-based aggregation component, which captures users' long-term periodic preferences, reduces accuracy since user sequences often show regular patterns, such as visiting on specific locations on fixed days of the week or month.
- Removing all the aforementioned components significantly reduces prediction accuracy, demonstrating that the designed modules play a crucial role in enhancing performance in recommendation tasks.

4.4 Hyper-parameter Sensitivity Experiment

We analyze the impact of four key hyperparameters on the performance: the number of GCN layers (l), the interval length (γ), and the aggregation weigh of long- and short-term embeddings (μ). Fig. 5 presents the results.

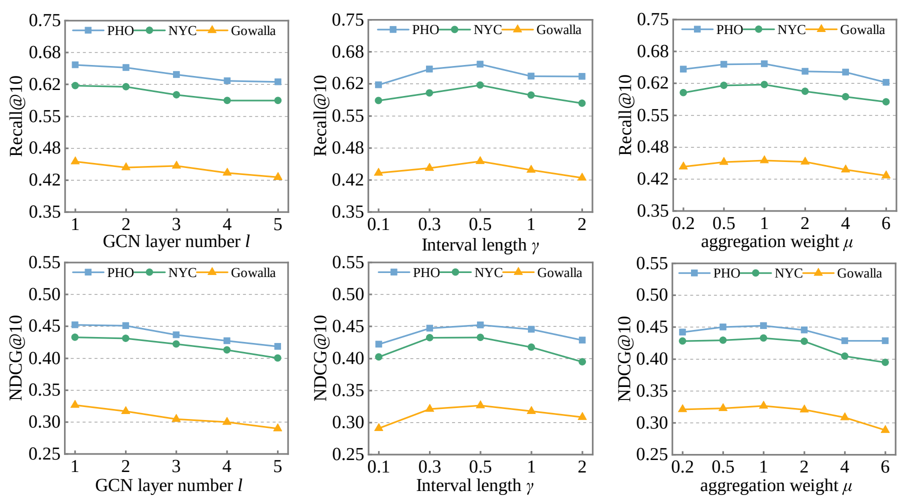

Fig. 5. Sensitivity analysis of hyperparameters

- **Impact of the number of GCN layers**: We evaluated model performance with the number of GCN layers l ranging from 1 to 5. The best performance was observed with a single GCN layer, with similar results for two layers, but a decline in performance with more layers. The graph structure in the GCN captures potential next POIs based on users' recent distance span preferences, and using too many layers can obscure these preferences, resulting in reduced performance.

- **Impact of the interval length**: We adjusted the interval length γ within the range {0.1, 0.3, 0.5, 1, 2} and found that an interval length of 0.5 yielded the best results. Smaller interval lengths did not generalize well in capturing distance span preferences, while larger interval lengths failed to capture fine-grained preferences effectively.
- **Impact of aggregation weight**: We tested how different weights of the long-term embedding affected the results of the experiment. The weight μ was tested within {0.2, 0.5, 1, 2, 4, 6}, and results indicated that excessively high weights led to over-reliance on long-term embedding which lacks of capture of spatial and temporal spans, thereby reducing prediction accuracy.
- **Generalizability analysis**: Based on the aforementioned analysis, the model demonstrates robust performance across different datasets when key hyper-parameters are controlled within specific ranges. For instance, setting the GCN layer l to 1, the interval length γ to 0.5, and the aggregation weight μ within the range of 0.5 to 1 yields near-optimal performance. This indicates that the model possesses strong generalizability, allowing it to be effectively extended to diverse datasets and real-world applications.

5 Related Work

Next POI Recommendation. Next POI recommendation aims to predict a user's next possible visit based on their historical check-in data. Early methods [23] relied on Markov chain assumptions, focusing on transition probabilities between POIs, which struggled with high-order sequential patterns. With the rise of deep learning, RNN-based and attention-based neural methods [5,6] became dominant due to their superior ability to model sequential dependencies from diverse perspectives. Later, to capture the geographic impact on user preferences, STGN [7] extended LSTM with spatio-temporal gates to capture short-term preferences, while attention-based models like STAN [9] leveraged spatial and temporal matrices as biases to enhance relationships between non-consecutive POIs in check-in sequences. However, these models focus solely on capturing spatio-temporal relationships within individual check-in sequences and overlook the complex geographic dependencies between POIs.

In order to solve the above problems, GNN-based methods are proposed which can construct high-quality POI embeddings. GETNext [10] applies graph convolution on trajectory graphs. Later, to avoid the lack of personalisation associated with predefined graphs, AGRAN [14] leverages graph structure learning for expressive POI representations. In addition, from other perspectives, STHGCN [13] uses a spatiotemporal hypergraph to capture patterns from individual and collaborative trajectories, and ImNext [15] models irregular check-in intervals in user check-ins and latent tasks. However, above methods insufficiently integrate long- and short-term preferences and often neglect personalized distance preferences, limiting recommendation performance.

Long- and Short-Term User Preferences in Recommendations. In next POI recommendation, long- and short-term modeling is essential for capturing users' stable preferences and dynamic interests. Early methods ST-LSTM [16] explicitly modeled long- and short-term behaviors incorporates spatial-temporal influence to address data sparsity, while [20] extended this idea with attention-based mechanisms for next POI recommendation. Later works introduced advanced techniques to integrate long- and short-term preferences more effectively, such as [17], which combines an intra-level attention-based long-term module with a short-term module that captures hourly and areal granularities. Moreover, PLSPL [18] merges long- and short-term preferences via user-specific linear combinations. More recently, researchers have explored unified approaches to enhance efficiency and accuracy. CLSPRec [19] employed a shared-parameter encoder to model both long- and short-term sequences, leveraging contrastive learning to distinguish individual and group travel preferences. However, the above methods fail to adequately consider periodic preferences when constructing long-term user preferences.

6 Conclusion

In this paper, we propose PSPL, which integrates users' personalized short-term spatio-temporal span preferences and periodic long-term periodic preferences for next POI recommendation. PSPL leverages GCN with S^2Graph to capture users' recent distance-span preferences, then encodes spatio-temporal span features and sequential order using the ASL Block. Finally, ASL Block is applied to aggregate the user's short-term spatio-temporal preferences, while DFT captures long-term periodic patterns. This comprehensive approach achieves superior prediction accuracy. Extensive experiments on real-world datasets validate the effectiveness of our proposed model.

7 Acknowledgment

This study was funded by the National Key Research and Development Program of China (No.2022YFC3004603); National Natural Science Foundation of China (No.62072220); Natural Science Foundation of Liaoning Province (Nos.2022-KF-13-06, 2022-BS-111); the Youth Talent Support Program of 'Xing Liao Talent Program' (No.XLYC2203003); Liaoning Provincial Department of Education Youth Project (No. JYTQN2023189).

References

1. J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, Deepmove: Predicting human mobility with attentional recurrent networks, in: The Web Conference, 2018, pp. 1459–1468.
2. P. Sánchez, A. Bellogín, Point-of-interest recommender systems based on location-based social networks: A survey from an experimental perspective, ACM Comput. Surv. 54 (11s) (2022) 223:1–223:37.
3. J. CHEN, J. WU, J. CHEN, X. XIN, Y. LI, X. HE, How graph convolutions amplify popularity bias for recommendation?, Front. Comput. Sci. (5) (2024) 185603.
4. P. Zhao, H. Zhu, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, X. Zhou, Where to go next: A spatio-temporal gated network for next POI recommendation, in: AAAI, 2019, pp. 5877–5884.

5. Q. Liu, S. Wu, L. Wang, T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts, in: AAAI, 2016, pp. 194–200.
6. K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, H. Yin, Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation, in: AAAI, 2020, pp. 214–221.
7. P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, X. Zhou, Where to go next: A spatio-temporal gated network for next POI recommendation, TKDE 34 (5) (2022) 2512–2524.
8. H. Zang, D. Han, X. Li, Z. Wan, M. Wang, CHA: categorical hierarchy-based attention for next POI recommendation, ACM Trans. Inf. Syst. 40 (1) (2022) 7:1–7:22.
9. Y. Luo, Q. Liu, Z. Liu, STAN: spatio-temporal attention network for next location recommendation, in: The Web Conference, 2021, pp. 2177–2185.
10. S. Yang, J. Liu, K. Zhao, Getnext: Trajectory flow map enhanced transformer for next POI recommendation, in: SIGIR, 2022, pp. 1144–1153.
11. J. Ou, H. Jin, X. Wang, H. Jiang, X. Wang, C. Zhou, STA-TCN: spatial-temporal attention over temporal convolutional network for next point-of-interest recommendation, TKDD 17 (9) (2023) 124:1–124:19.
12. Y. Wang, Y. Song, S. Li, C. Cheng, W. Ju, M. Zhang, S. Wang, Disencite: Graph-based disentangled representation learning for context-specific citation generation, in: AAAI, 2022, pp. 11449–11458.
13. X. Yan, T. Song, Y. Jiao, J. He, J. Wang, R. Li, W. Chu, Spatio-temporal hypergraph learning for next POI recommendation, in: SIGIR, 2023, pp. 403–412.
14. Z. Wang, Y. Zhu, C. Wang, W. Ma, B. Li, J. Yu, Adaptive graph representation learning for next POI recommendation, in: SIGIR, 2023, pp. 393–402.
15. X. He, W. He, Y. Liu, X. Lu, Y. Xiao, Y. Liu, Imnext: Irregular interval attention and multi-task learning for next POI recommendation, Knowl. Based Syst. 293 (2024) 111674.
16. D. Kong, F. Wu, HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction, in: IJCAI, 2018, pp. 2341–2347.
17. Q. Cui, C. Zhang, Y. Zhang, J. Wang, M. Cai, ST-PIL: spatial-temporal periodic interest learning for next point-of-interest recommendation, in: CIKM, 2021, pp. 2960–2964.
18. Y. Wu, K. Li, G. Zhao, X. Qian, Personalized long- and short-term preference learning for next POI recommendation, TKDE 34 (4) (2022) 1944–1957.
19. C. Duan, W. Fan, W. Zhou, H. Liu, J. Wen, Clsprec: Contrastive learning of long and short-term preferences for next POI recommendation, in: CIKM, 2023, pp. 473–482.
20. L. Huang, Y. Ma, S. Wang, Y. Liu, An attention-based spatiotemporal LSTM network for next POI recommendation, IEEE Trans. Serv. Comput. 14 (6) (2021) 1585–1597.
21. S. Feng, G. Cong, B. An, Y. M. Chee, Poi2vec: Geographical latent representation for predicting future visitors, in: AAAI, 2017, pp. 102–108.
22. X. Zhang, Z. Ye, J. Lu, F. Zhuang, Y. Zheng, D. Yu, Fine-grained preference-aware personalized federated POI recommendation with data sparsity, in: SIGIR, 2023, pp. 413–422.
23. C. Cheng, H. Yang, M. R. Lyu, I. King, Where you like to go next: successive point-of-interest recommendation, in: IJCAI, 2013, p. 2605–2611.