CAAI Transactions on Intelligence Technology

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

A Paradigm of Temporal-Weather-Aware Transition Pattern for POI Recommendation

Junyang Chen[1,2] | Jingcai Guo[3] | Huan Wang[4] | Zhihui Lai[1] | Qin Zhang[1] | Kaishun Wu[5] | Liang-Jie Zhang[1]

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China | [2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China | [3]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China | [4]College of Informatics, Huazhong Agricultural University, Wuhan, China | [5]DSA Thrust and IoT Thrust Center, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

Correspondence: Jingcai Guo (jingcai.guo@gmail.com)

Received: 24 August 2024 | **Revised:** 28 February 2025 | **Accepted:** 18 March 2025

Funding: This work was supported by Stable Support Project of Shenzhen (20231120161634002), Shenzhen Science and Technology Programme (JCYJ20240813141417023), Natural Science Foundation of Guangdong Province of China (2025A1515010233), Guangdong Provincial Department of Education (2024KTSCX060), Tencent 'Rhinoceros Birds'—Scientific Research Foundation for Young Teachers of Shenzhen University, Open Project of State Key Laboratory for Novel Software Technology of Nanjing University (KFKT2025B22), Hong Kong RGC General Research Fund (No. 152211/23E and 15216424/24E), PolyU Internal Fund (No. P0043932, P0048988) and NVIDIA AI Technology Centre.

Keywords: data mining | decision making | multimedia

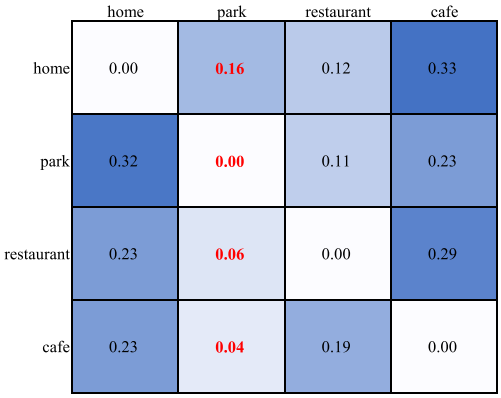
ABSTRACT

Point of interest (POI) recommendation analyses user preferences through historical check-in data. However, existing POI recommendation methods often overlook the influence of weather information and face the challenge of sparse historical data for individual users. To address these issues, this paper proposes a new paradigm, namely temporal-weather-aware transition pattern for POI recommendation (TWTransNet). This paradigm is designed to capture user transition patterns under different times and weather conditions. Additionally, we introduce the construction of a user-POI interaction graph to alleviate the problem of sparse historical data for individual users. Furthermore, when predicting user interests by aggregating graph information, some POIs may not be suitable for visitation under current weather conditions. To account for this, we propose an attention mechanism to filter POI neighbours when aggregating information from the graph, considering the impact of weather and time. Empirical results on two real-world datasets demonstrate the superior performance of our proposed method, showing a substantial improvement of 6.91%–23.31% in terms of prediction accuracy.

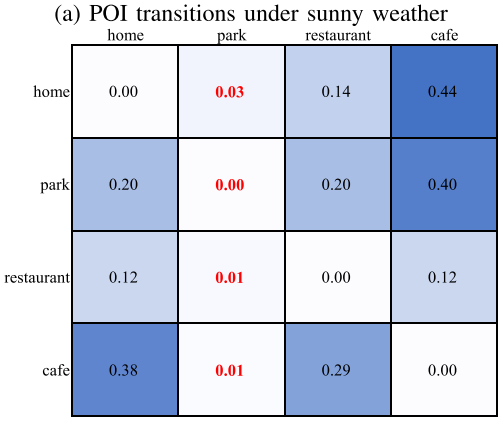
1 | Introduction

With the widespread use of mobile devices equipped with location systems, location-based social networks (LBSN) have become prevalent [1]. This enables users to share real-time POI information, including their visits to points of interest (POI) along with the associated time and sentiments. Given that LBSN provides an excellent opportunity to understand user movement patterns, POI recommendation has attracted attention for predicting user-interested next POI in travel planning and smart transportation [2–4].

As shown in Figure 1, we conducted a statistical analysis of real user check-in records from the Boston dataset in Foursquare and found it essential to consider the impact of weather on travel behaviour. For instance, Figure 1a illustrates the transition probabilities between POIs under sunny and rainy weather conditions. It reveals that on sunny days, users are more likely to transition to parks from other POIs. For example, the probability of transitioning from a residence to a park is 0.16 on sunny days. Conversely, as shown in Figure 1b, the frequency of park visits decreases on rainy days, with the probability of transitioning from a residence to a park dropping to 0.03.



(a) POI transitions under sunny weather



(b) POI transitions under rainy weather

FIGURE 1 | Statistical probabilities of POI transitions under sunny weather and rainy weather conditions.

Furthermore, on rainy days, users tend to reduce visits to parks and prefer venues such as restaurants and cafés. Additionally, both real-life observations and our experimental data confirm that weather changes can occur within a single hour, such as transitioning from sunny to heavy rain. Therefore, it is crucial to consider weather changing within a day when designing POI recommendation models.

Mining personalised visit preferences from user historical trajectories in POI recommendation [5] faces significant challenges due to factors including time, distance and weather. To tackle these challenges, several approaches including RNN-based methods [6, 7] and graph-based methods [8, 9] are proposed. In summary, despite their significant progress in modelling personalised visit preferences, existing methods still face the following limitations:

- **Impact of User Behaviour Changes over Weather and Time:** Recent methods, such as those in studies by [10–12], often fail to account for the transition patterns between time and weather factors. For instance, a user's movement pattern might be 'home at 10:00 on a sunny day' transitioning to 'park/cafe' as illustrated in Figure 2. These transitions indicate that users' subsequent decisions are influenced by their initial choices, suggesting that time and weather factors play a significant role in shaping user behaviour. Our approach explicitly models these transitions to capture such dependencies.

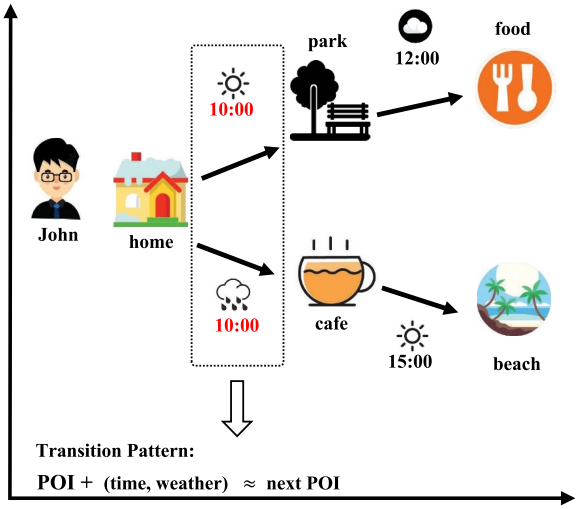


FIGURE 2 | Impact of time and weather in POI transition.

- **Mitigating Noise in Information Passing Under Dynamic Conditions:** Current graph-based methods, as seen in the work in refs. [4, 13], are often susceptible to noise in information passing under varying time and weather conditions. This refers to situations where, if the weather conditions differ from users' expectations, the model should disregard certain POI information related to those unexpected weather scenarios. By refining how information is filtered and passed under different conditions, our method aims to improve the accuracy of recommendations.

To address above challenges, this paper proposes a method called TWTransNet. **For the first problem, we employ the translation paradigm to capture the relationship between POI transition under different times and weather conditions.**

Definition of translation paradigm: It refers to a concept inspired by knowledge graph-embedding methods, such as TransE, which represent relationships between entities as vector translations in the embedding space. In the context of our paper, we use this paradigm to model the relationship between POI transitions under different times and weather conditions. Specifically, we treat the transition between two POIs as a 'translation' from one POI to another, influenced by time and weather. To clarify this for readers, we will include a detailed explanation of the translation paradigm when it first appears in the paper, including a brief comparison to its use in knowledge graphs and its relevance to our proposed model.

Then, we model the POI transitions with time and weather by designing a triplet relationship. Specifically, head entity h and tail entity t represent two POIs, and r represents the time and weather information when transitioning from POI h to POI t. We measure the relationship of POIs by the distance calculating by time and weather. And TWTransNet learns node and edge representations from the constructed POI transition graph. **For**

the second problem, we filtrate POI neighbours based on learnt representations with time and weather consideration, uncovering noise information. As shown in Figure 3, we intend to enhance the predictive performance of Seq1 regarding the next POI by incorporating Seq 2 and Seq 3. We plan to model sequences of all POIs using graph-based methods. Blue nodes represent the nodes in the graph where we need to strengthen information aggregation, red nodes indicate nodes that become available for effective aggregation after our method and black nodes represent nodes that should be discarded after considering time and weather. Finally, we design a simple self-attention model for the ultimate POI recommendations. It is worth noting that our approach can be combined with existing POI recommendation models [12, 14] and significantly enhance their original experimental performance. Our contributions are as follows:

1. We propose a new paradigm, namely the temporal-weather-aware transition pattern for POI recommendation.
2. We specifically design a translation paradigm to capture the relationship between POI transitions under different times and weather conditions. Furthermore, we filter POI neighbours based on learnt representations, taking into account time and weather considerations to uncover noise information.
3. Comprehensive experiments on three real datasets validate the effectiveness of our proposed method. The results demonstrate superior POI recommendation performance compared to existing methods.

2 | Related Work

This work is relevant to POI recommendation based on sequence methods. Additionally, since graph-based approaches can effectively explore various factors influencing user preferences, hence our work is related to both of these.

2.1 | Sequential-Based Recommendation

First, FPMC [15] integrates Markov chain and matrix decomposition techniques to grasp personalised movement patterns of users. In the era of deep learning, RNN-based architectures

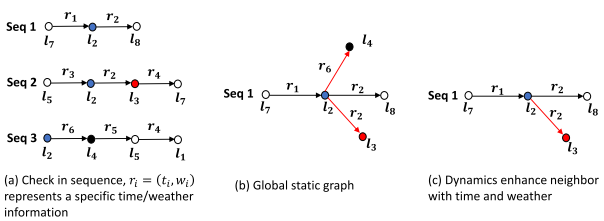


FIGURE 3 | Filter POI transition neighbours. (a) Check in sequences; (b) the previous method of building POI transition neighbours without considering time and weather effects, which can lead to the introduction of noisy neighbours that do not meet user transition preferences (node that marked in black); (c) POI (node that marked in red) is selected for effective aggregation.

including LSTM and GRU [6, 7, 11, 16] have been widely applied to model sequential user check-in data. For example, the LSTPM [17] uses LSTM to fuse geographic information to model users' short-term preferences, whereas the STGN [16] uses a spatio-temporal gating mechanism based on LSTM to fuse time and geographic information to capture users' short-term and long-term sequential preferences. More recently, DeepMove [18] combines the attention mechanism with GRU to capture sequential preferences, whereas the STAN model [12] uses this attention mechanism to fuse spatio-temporal interval information to capture transfer patterns between continuous and discontinuous POIs.

Despite their success, these methods still face challenges in inadequate consideration of time and weather factors in POI transition.

2.2 | Graph-Based Recommendation

Graph-based methods including GCN and GAT [19] have gained popularity in recommendation tasks due to their ability to capture complex relationships among factors (including but not limited to POIs, time and weather). For POI recommendation, researchers have explored constructing graphs from user historical trajectories or POI distances to model transition patterns [8, 9]. These approaches typically treat POIs as nodes and user transitions as edges, constructing static POI-POI graphs to capture global relationships. For example, the STP-UDGAT [9] uses GAT to model the correlation between POIs from a global and local perspective based on the POI-POI graph. ARNN [20] uses meta-path-based methods to mine the transfer neighbours of locations based on the graph, and SGRec [13] generates richer POI representations based on graph-enhanced POI sequences to improve prediction performance. GUGEN [21] introduces a global user interaction graph to model social relationships and behaviour similarities, integrating spatio-temporal features to capture dynamic user preferences. SECTS [22] leverages temporal patterns and cross-domain data (e.g., check-ins across platforms) using a graph-based framework. It dynamically aligns user preferences across domains while accounting for time-sensitive context shifts. SLS-REC [23] utilises contrastive learning to disentangle long-term stable preferences and short-term dynamic interests via a spatio-temporal hypergraph neural network. MAWI [24] integrates severe weather data with user behaviour patterns to improve context-aware recommendations. It uniquely addresses environmental impacts on mobility and POI accessibility. PK boosting [25] combines user activity levels (e.g., check-in frequency, temporal patterns) with multi-contextual factors (geographical, categorical and popularity scores) using adaptive weighting mechanisms. However, these methods are insufficient modelling information passing under different times and weather conditions.

Moreover, DPSR [26] introduces a heuristic-based approach for POI recommendation, taking into account time, weather, POI location and category. In contrast, our approach leverages deep learning and addresses several novel aspects for the first time: (1) The direct relationship between location transitions and weather conditions; (2) the consideration of user relationships,

specifically how the interests of similar users influence the target user, which was overlooked in DPSR; (3) in the final recommendation prediction phase, we employ an attention mechanism to effectively integrate information on weather, location and user interests.

2.3 | Challenges and Limitations

Although significant progress has been made in POI recommendation research, several challenges and limitations persist. One major challenge is the inadequate consideration of time and weather factors in existing methods. Time and weather play crucial roles in influencing user access behaviour, and effectively incorporating these factors into recommendation models remains an open problem. Additionally, many existing methods struggle with the sparse trajectory issue, where users' check-in records are limited, making it challenging to capture accurate user preferences. Although some methods leverage social relationships to address this issue, obtaining reliable social relationships can be difficult, especially in scenarios where users have limited interactions with each other. The task of leveraging user visit trajectories to discover preference relationships among users in the absence of social relationships remains a significant challenge.

3 | Preliminary

We formulate users, POIs, time and weather as follows: $U = \{u_1, u_2, ..., u_{|U|}\}$, $P = \{l_1, l_2, ..., l_{|P|}\}$, $T = \{t_1, t_2, ..., t_{|T|}\}$ and $W = \{w_1, w_2, ..., w_{|W|}\}$. Each POI l_i is denoted by geographic coordinates (lon, lat), that is, longitude and latitude.

Definition Check-in: A check-in is defined as $c = (u_i, l_i, t_i, w_i)$, representing the user u_i visiting POI l_i at time t_i and under weather conditions w_i.

Definition User Trajectory: A user trajectory is defined as $S_u = \{c_1, c_2, ...c_n\}$, representing the sequence of check-ins by user u in chronological order.

Definition Transition Graph: A POI transition graph is defined as $G_T = (V_T, E_T)$ where V_T represents the set of POIs. If a user consecutively visits POIs l_i and l_j, a directed edge is created between nodes l_i and l_j. The edge carries information $r_i = (t_i, w_i)$, indicating the transition from l_i to l_j under weather w_i and time t_i as r_i.

Definition User-POI Graph: A user-POI interaction graph is defined as $G_U = (V_U, E_U)$, an undirected bipartite graph where V_U represents user sets and POI sets. If a user u visits POI l_i, an undirected edge is created between nodes u and l_i. The edge carries information (t_i, w_i), indicating the visit's time t_i and weather w_i.

Definition POI Recommendation: Given user set U and POI set L, where each user has a visit trajectory S_u, our objective is to recommend the top-k interesting POIs that a user may visit in the next step.

4 | Proposed Model

The proposed TWTransNe framework, as illustrated in Figure 4, consists of several main components: a temporal-weather-aware enhancement module, a user relationship enhancement module and a POI recommendation module.

4.1 | Temporal-Weather-Aware Enhancement Module

Time and weather significantly influence POI transitions. We argue that the simplistic concatenation or summation used in existing approaches [10, 11] fails to model the intricate relationships between POI transitions under different times and weather conditions. Drawing inspiration from knowledge graphs [27, 28], we naturally transform POI transitions into triplets (l_i, r_i, l_j), where $r_i = (t_i, w_i)$ represents a user's transition from POI l_i to l_j under weather w_i and time t_i. This is analogous to triplets (h, r, t) in knowledge graphs, where l_i and l_j represent head entity h and tail entity t, and r_i represents the relationship. **To model the connection between POI transitions and**

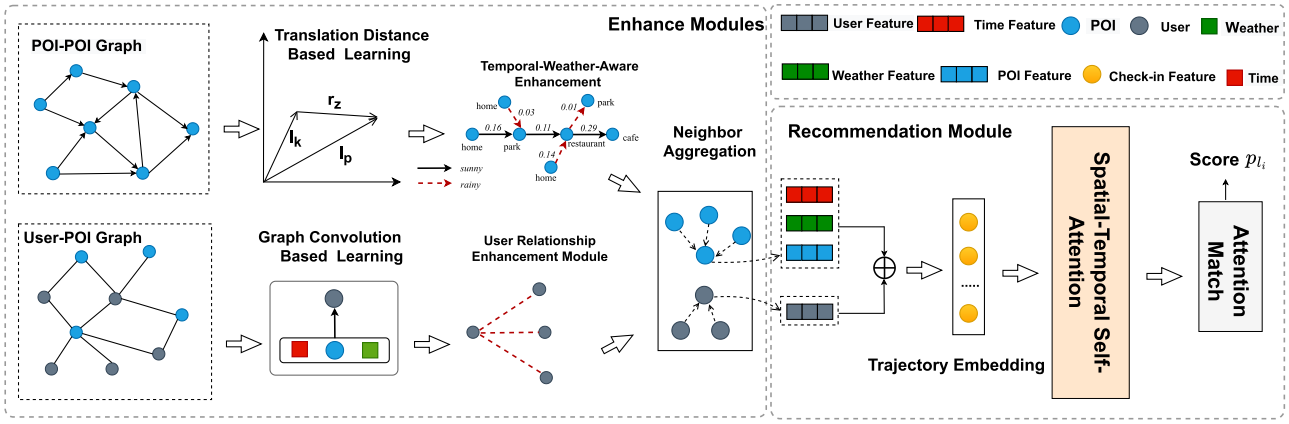


FIGURE 4 | The overall framework of our proposed model.

time/weather, we propose representing the temporal-weather-aware paradigm for learning the representation of the transition graph G_T. This paradigm, derived from the concept of translational invariance (i.e., transE), ensures that the vectors of head entities move closer to tail entities in space after translation through the relationship. This means that a user is more likely to transition from l_i to l_j at time t_i and under weather w_i, as depicted in Figure 2: 'home + (10:00, rain) ⇒ cafe'. Let e_{l_i}, e_{t_i} and $e_{w_i} \in \mathbb{R}^d$ denote the representations of POI l_i, time t_i and weather w_i, respectively. The transition formulation is given by

$$g(l_i, r_i, l_j) = \left\| e_{l_i} + e_{r_i} - e_{l_j} \right\|_2^2, \quad \text{where } e_{r_i} = e_{t_i} + e_{w_i}, \tag{1}$$

where the smaller the value of the distance function $g(l_i, r_i, l_j)$, the higher the probability of the POI transition occurs, and vice versa. **Physical meaning of the translation distance-based learning** (Equation (1)): It evaluates the plausibility of a POI transition (l_i, r_i, l_j) by calculating the squared L_2-norm distance between the vector $e_{l_i} + e_{r_i}$ (representing the transition from POI l_i under relation r_i) and the vector e_{l_j} (representing the target POI l_j). A smaller distance indicates a higher likelihood of the transition, meaning that the temporal and weather context makes l_j a more probable destination from l_i.

Then, for each POI transition, we introduce a corresponding negative sample (l_i, r_i, l_n) obtained by randomly replacing the tail entity. The optimisation process can be defined by

$$L_T = \sum_{(l_i, r_i, l_j) \in G_T} \max(0, g(l_i, r_i, l_j) + \gamma - g(l_i, r_i, l_n)), \tag{2}$$

where the $\max(a, b)$ function selects the maximum value between a and b, and γ is the margin used for adjusting the gap between positive and negative samples. **Purpose of negative samples**: Negative samples are used to train the model to differentiate valid POI transitions from invalid ones. By contrasting positive triplets (l_i, r_i, l_j) (true transitions) with negative triplets (l_i, r_i, l_n) (invalid transitions), the model learns meaningful representations of the POI transitions under different temporal and weather conditions. **Process of negative sampling**: For a given positive triplet (l_i, r_i, l_j), the negative sample (l_i, r_i, l_n) is generated by randomly replacing the tail entity l_j with another POI l_n from the dataset. The replacement POI l_n is selected under the following constraints: (a) $l_n \neq l_j$, ensuring that the negative sample differs from the positive one. (b) l_n should not be a POI that the user has transitioned to under the same temporal and weather context r_i, ensuring the negative sample is semantically incorrect.

Besides, since existing approaches [4, 8] neglect personalised POI transition patterns for users under different times and weather conditions, we design a method to dynamically enhance user trajectories using the learnt relationships between POI transitions and time/weather. The overall process is illustrated in Figure 4. Based on the transition in Equation (1), we define the likelihood of user u transitioning from l_i to l_j under the current time and weather r_i as follows:

$$s(l_i, r_i, l_j) = \exp(-g(l_i, r_i, l_j)). \tag{3}$$

Based on the above likelihood, we select top k POIs as the neighbour of POI i to construct an enhanced user trajectory graph, where each edge incorporates the influence of time and weather capturing personalised transition patterns.

4.2 | User Relationship Enhancement Module

POI recommendation faces the challenge of sparse user trajectories. Some existing methods [5] mitigate this issue by introducing social relationships, but acquiring social relationships is often challenging. In this section, we design a user relationship enhancement module based on graph neural networks to explore users' latent visitation interests. Firstly, we conduct representation learning on the user-place interaction graph G_U. Drawing inspiration from context-aware recommendation methods [29, 30], we model user representations by considering more granular information about time and weather to depict user preferences more comprehensively. For each user representation update, we give

$$e_u^{(h)} = \frac{1}{|G(u)|} \sum_{l_i \in G(u)} \left(e_{l_i}^{(h-1)} + e_{t_i}^{(h-1)} + e_{w_i}^{(h-1)} \right), \tag{4}$$

where $G(u)$ represents the set of places visited by user u, and t_i and w_i represent the weather and time when the place is visited. $e_u^{(h)} \in \mathbb{R}^d$ denotes the updated user representation in the h-th layer. $e_{l_i}^{(h-1)}$, $e_{t_i}^{(h-1)}$ and $e_{w_i}^{(h-1)} \in \mathbb{R}^d$ represent the updated representations of the POI, time and weather in the $(h-1)$-th layer. To capture collaborative information among users and depict visiting preferences, information aggregation and representation updates are performed for POIs, times and weather in G_U.

Then, we further define each POI representation $e_{l_i}^{(h)}$ by

$$e_{l_i}^{(h)} = \frac{1}{|G(l_i)|} \sum_{u \in G(l_i)} \left(e_u^{(h-1)} + e_{t_i}^{(h-1)} + e_{w_i}^{(h-1)} \right), \tag{5}$$

where $G(l_i)$ represents the set of users who visited POI l_i. Besides, for time representation $e_{t_i}^{(h)}$, we have

$$e_{t_i}^{(h)} = \frac{1}{|G(t_i)|} \sum_{(u, l_i) \in G(t_i)} \left(e_u^{(h-1)} + e_{t_i}^{(h-1)} \right), \tag{6}$$

where $G(t_i)$ represents check-in records at time t_i, such as $(u, l_i) \in G(t_i)$ indicating that user u visited POI l_i at time t_i. Similarly, we can obtain weather representation $e_{w_i}^{(h)}$ as follows:

$$e_{w_i}^{(h)} = \frac{1}{|G(w_i)|} \sum_{(u, w_i) \in G(w_i)} \left(e_u^{(h-1)} + e_{w_i}^{(h-1)} \right), \tag{7}$$

where $G(w_i)$ represents check-in records at weather w_i. For example, $(u, w_i) \in G(w_i)$ indicating that the user u visited POI l_i under weather w_i. After learning the representation of the user-POI interaction graph G_U with graph neural networks, we

obtain user representations e_{u_i} that reflect user visitation preferences. Next, we use a similarity function to measure the preference correlation between users. Here, we adopt the K-nearest neighbour method, calculating the similarity among users. The more similar the user representations, the more similar the visiting preferences. We select the top k users with the highest similarity as neighbours, establishing connection relationships with user u, denoted as $N(u)$. It is noteworthy that our enhancement module serves as a portable data preprocessing component, allowing for easy and seamless integration with other methods.

4.3 | POI Recommendation Module

After enhancing user trajectories and relationships, in this section, we aggregate the enhanced neighbours to obtain user and POI representations. The neighbour aggregation module takes POI embeddings along with their corresponding contextual information, such as time and weather, as inputs. It then outputs refined representations that effectively capture spatio-temporal dependencies, enabling a more accurate modelling of user movement patterns.

Inspired by the LightGCN [19], we design a similar aggregation function. Each visited POI l_i in the historical trajectory S_u of user u has

$$e_{l_i} = \sigma\left(\frac{1}{K}\sum_{l_i \in \mathcal{N}(u)} W_1 e_{l_i}\right), \tag{8}$$

where $e_{l_i} \in \mathbb{R}^d$ is the embedding of POI l_i, $W_1 \in \mathbb{R}^{d\times d}$ denotes the weight matrix, K is the size of the neighbour set and $\sigma(\cdot)$ is the activation function. Similar to Equation (8), we aggregate the user neighbours $\mathcal{N}(u)$ using the same approach to obtain the user representation e_u, encoding the user's general preferences. Next, we use multi-modal embedding that incorporates user, POI, time and weather. We divide time into 24 periods, each representing a time slot in a day and weather into eight categories: *Clear, Cloudy, Rain, Fog, Snow, Wind* etc. Instead of using one-hot encoding, we represent user, POI, time and weather information using low-dimensional dense vectors, reducing computation and avoiding sparse representations. For each check-in record of user u, we have the check-in representation $c_i \in \mathbb{R}^d$, which is the sum of user, POI, time and weather representations. Finally, we obtain the check-in representation encoding user, POI, time and weather information, which will be used in the subsequent POI prediction layer.

The spatial-temporal self-attention module: For POI recommendation, we use a self-attention mechanism to obtain predicted POI by aggregating information from other relevant visited POIs. Inspired by works like STAN [12, 14, 16], we believe that spatial distance and time intervals of POIs will affect transition patterns. Specifically, we input the history trajectory of a user u, then we can obtain the predicted POI i as follows:

$$\begin{aligned} Q_i = E(u)W^Q, \quad K_i = E(u)W^K, \quad V_i = E(u)W^V, \\ A(u) = \frac{Q_i K_i^T}{\sqrt{d}}, \end{aligned} \tag{9}$$

where $E(u) = \{c_1, c_2, ..., c_n\} \in \mathbb{R}^{(n\times d)}$ denotes the input trajectory of user u, and $W^Q, W^K, W^V \in \mathbb{R}^{(d\times d)}$ are parameter matrices. For each attention score $a_{ij} \in A(u)$, we incorporate temporal-spatial information between POIs l_i and l_j in the user's trajectory S_u:

$$a'_{ij} = a_{ij} + d_s(l_i, l_j) + d_t(l_i, l_j) \tag{10}$$

where $d_s(*)$ denotes POI distance by Haversine [31], and $d_t(*)$ denotes the time interval. The details are as follows:

$$\begin{aligned} &d_t(l_i, l_j) = |t_j - t_i|, \\ &d_s(l_i, l_j) = 2R\ \sin^{-1} \\ &\left(\sqrt{\sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_{l_i})\cos(lat_{l_j})\sin^2\left(\frac{\Delta lon}{2}\right)}\right), \end{aligned} \tag{11}$$

where R is the radius of Earth. After incorporating temporal-spatial information, we obtain a new attention score matrix $A'(u)$, and then we further obtain the updated trajectory embedding as follows:

$$R(u) = \text{Softmax}(A'(u))V_i, \tag{12}$$

where $R(u) \in \mathbb{R}^{(n\times d)}$ is the trajectory embedding updated through the temporal-spatial self-attention layer, which will be used in the subsequent modules. We use an attention mechanism to calculate the prediction scores for each candidate POI l_i as follows:

$$p_{l_i} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}} + d_s + d_t\right)W_2, \tag{13}$$

where $Q \in \mathbb{R}^d$ is the POI embedding, K is the trajectory embedding $R(u_i) \in \mathbb{R}^{(n\times d)}$ and $W_2 \in \mathbb{R}^{(n\times 1)}$ is a parameter matrix. Similar to the temporal-spatial self-attention mechanism layer, when calculating the recommendation scores for candidate POIs, we also incorporate temporal-spatial gap information between candidate POIs and user trajectory check-in records. Next, we use a cross-entropy to form the final loss:

$$L = -\sum_{l_i \in G_U}\left(\log\sigma(p_{l_i}) + \sum_{l_n \in \text{neg}} \log\sigma(1 - p_{l_n})\right), \tag{14}$$

where p_{l_i} represents the probability that the user will visit POI l_i next, neg corresponds to negative samples and $\sigma(\cdot)$ is the sigmoid activation function.

5 | Experiments

In this section, we conduct experiments on real-world LBSN datasets to evaluate our model. Our evaluations aim to answer the following questions:

Question 1. How does our method compare against state-of-the-art methods for POI recommendation?

Question 2. Is our method applicable to other POI recommendation approaches (scalability on existing POI recommendation backbones)?

Question 3. How do the designed components impact the overall performance?

Question 4. Can our method handle the sparse trajectory problem in POI recommendation and improve its effectiveness?

Question 5. What is the impact of key parameters on performance?

Question 6. How TWTransNet can be deployed in real-world recommendation systems?

5.1 | Datasets

We evaluate our model on three public datasets collected from POI-based service platforms [12]: FourSquare-Boston, FourSquare-Honolulu and FourSquare-Miami. Table 1 shows the statistics of the three datasets. FourSquare-Boston dataset is collected from April 2012 to September 2013. FourSquare-Honolulu and FourSquare-Miami datasets consist of check-ins from April 2013 to September 2013. Each check-in record includes user ID, POI ID, GPS coordinates, timestamp, POI category and weather category. We sort the check-in records of each user in chronological order by the timestamp. We follow the data partitioning setup of previous work [11, 12]. Each user has m check-in records. We consider the $[1, m-2]$ sequence for training, with the $[1, m-3]$ sequence as the input and the $[2, m-2]$ sequence as the label. We take the $[1, m-2]$ check-ins as the input sequence for validation and the $(m-1)$-th visited POI is used as the label. For testing, we use the $[1, m-1]$ check-ins as the input sequence and the m-th check-in as the label.

5.2 | Baseline Methods

We adopt the following representative methods as the baselines to demonstrate the effectiveness of our model in the experiments.

TABLE 1 | Statistic of datasets.

Dataset	Boston	Honolulu	Miami
#User	469	224	299
#POI	1961	818	1216
#Check-ins	44,190	22,513	28,700
Avg.# check-ins per POI	22.53	27.52	23.61

LSTM (1997) [32]: It captures long-term dependency with a memory cell and multiplicative gates.

Flashback (2020) [14]: A RNN-based model, which uses the spatio-temporal information to consider the relationship between the current state and multiple previous states in prediction.

GeoSAN (2020) [10]: A sequential POI recommendation model that uses hierarchical gridding of GPS POIs for encoding geographical information of POIs and uses the self-attention layer to capture POI transition patterns.

STAN (2021) [12]: An attention-based model utilises the spatio-temporal information of check-ins within the user historical trajectory to capture the correlation between nonadjacent check-ins.

CARAN (2022) [11]: A context-aware attention-based model captures POI transition patterns, explicitly using multiple contextual information such as time, weather and POI category.

KGNext (2024) [33]: It is a knowledge graph-enhanced transformer model that constructs a transitional-interactive knowledge graph to capture complex relationships between users, POIs and POI categories while integrating historical trajectory information to generate user trajectory embeddings. Then, it utilises these encoded features for fine-grained predictions, enabling accurate next POI recommendation.

5.3 | Experiment Settings

We implement our model using the PyTorch framework with one Nvidia A800 GPU. We list the following key hyper-parameters in our model. We set the dimension size to 50 for all embeddings. We use the Adam optimiser to optimise the model parameters with the learning rate of 0.0003. We set the maximum length of the historical trajectory sequence to 100. We run our model 100 epochs in training. The number of negative samples is set to 10.

5.4 | Evaluation Metrics

To evaluate the performance of all models, we adopt average Accuracy@N (Acc@N) and mean reciprocal rank (MRR) as evaluation metrics, which are commonly used in next POI recommendation tasks.

First, Acc@N computes the rate of true positive samples in top-N recommended positive samples which is defined by

$$Acc@N = \frac{1}{|U|}\sum_{u\in U}\frac{|S_u^N \cap S_u^L|}{|S_u^L|}, \tag{15}$$

where S_u^N represents the top-N recommended POIs for user u and S_u^L is the ground truth label for user u. We set $N = \{5, 10, 20\}$ in our experiments for Acc@N. The larger the value for Acc@N, the better the performance.

Then, MRR is a metric used to evaluate the quality of a ranked list of results. It measures the average of the reciprocal ranks of the first ground-truth item in a list of search results across multiple queries. It is defined as

$$\mathrm{MRR} = \frac{1}{|U|} \sum_{u \in U}^{|U|} \frac{1}{\mathrm{rank}_u} \tag{16}$$

where rank_u denotes the rank position of the ground-truth item for user u. Note that the length of $|\mathrm{rank}_u|$ is not fixed because it corresponds to the position of the first item in the user's recommendation list that appears in the ground-truth results.

5.5 | Experimental Results (Question 1)

Tables 2–4 provide a comprehensive comparison of recommendation performance across all models on the three datasets. The results underscore the superior performance of our model compared to other baseline methods. In general, we observe that

1. On the Boston dataset, our model exhibits remarkable improvements of 7.99%, 9.75%, 11.93% and 14.65% in Acc@5, Acc@10, Acc@20 and MRR respectively, compared to the second-best approach, as shown in Table 2. Similarly, on the Honolulu dataset, our model outperforms the best competitors with an improvement of 23.31%, 17.59%, 11.65% and 9.91%, as shown in Table 3. Besides, on the Miami dataset, our method achieves 6.91%, 8.89%, 10.94% and 11.69% improvements, as shown in Table 4.
2. Notably, our model surpasses CARAN, which oversimplifies the integration of time and weather information, thus limiting its ability to capture the intricate relationship between POI transition and time and weather. Moreover, compared to KGNext, STAN and Flashback, our model demonstrates superior performance. This discrepancy can be attributed to the fact that these models overlook the impact of weather information.
3. We compared the performance of our proposed method across three datasets and observed the most significant accuracy improvement on the Honolulu dataset. This can be attributed to the dataset characteristics, as shown in the statistical summary in Table 1, where the 'Avg.# check-ins per POI' is higher. The longer sequences in the Honolulu dataset allow our method to more effectively capture the impact of different weather conditions on POI transition probabilities. These findings robustly establish the superiority of our model. They underscore the effectiveness of leveraging graph neural networks and the translation distance paradigm to enhance POI transition and user preference relationships through the integration of time and weather information. Additionally, Flashback outperforms attention-based models in our experiments, primarily due to its proficiency in modelling short-term user preferences within sparse user mobility scenarios.

TABLE 2 | Performance comparison in Acc@K and MRR on Boston.

	Boston			
Method	**Acc@5**	**Acc@10**	**Acc@20**	**MRR**
LSTM	0.1241	0.1833	0.2421	0.0827
Flashback	0.1677	0.2432	0.3235	0.1324
GeoSAN	0.1034	0.1621	0.2092	0.0978
STAN	0.1428	0.2124	0.3073	0.1033
CARAN	0.1577	0.2302	0.3155	0.1141
KGNext	0.1442	0.2113	0.3060	0.1027
TWTransNet (Ours)	**0.1811**	**0.2669**	**0.3621**	**0.1518**
Improv. (%)	7.99%	9.75%	11.93%	14.65%

Note: Improv. is compared with the second-best method. Bold indicates the best-performing method.

TABLE 3 | Performance comparison in Acc@K and MRR on Honolulu.

	Honolulu			
Method	**Acc@5**	**Acc@10**	**Acc@20**	**MRR**
LSTM	0.1564	0.2053	0.2745	0.1098
Flashback	0.1956	0.2581	0.3563	0.1615
GeoSAN	0.1584	0.2143	0.2723	0.1495
STAN	0.1651	0.2234	0.2965	0.1219
CARAN	0.1794	0.2323	0.3035	0.1287
KGNext	0.1732	0.2283	0.2911	0.1259
TWTransNet (Ours)	**0.2412**	**0.3034**	**0.3978**	**0.1775**
Improv. (%)	23.31%	17.59%	11.65%	9.91%

Note: Improv. is compared with the second-best method. Bold indicates the best-performing method.

5.6 | Universality of Enhancement Module (Question 2)

To demonstrate the scalability of TWTransNet as a paradigm, we integrate our enhancement module with existing models, denoted as Flashback+, STAN+, CARAN+ and KGNext+ yielding improved performance compared to their respective base models. Flashback+ achieves higher accuracy metrics

TABLE 4 | Performance comparison in Acc@K and MRR on Miami.

	Miami			
Method	**Acc@5**	**Acc@10**	**Acc@20**	**MRR**
LSTM	0.1694	0.2453	0.3245	0.1167
Flashback	0.3213	0.3805	0.4733	0.1932
GeoSAN	0.1974	0.2623	0.3623	0.1431
STAN	0.2641	0.3377	0.4414	0.1622
CARAN	0.2813	0.3473	0.4635	0.1831
KGNext	0.2691	0.3412	0.4405	0.1654
TWTransNet (Ours)	**0.3435**	**0.4147**	**0.5251**	**0.2158**
Improv. (%)	6.91%	8.98%	10.94%	11.69%

Note: Improv. is compared with the second-best method. Bold indicates the best-performing method.

across all evaluation points compared to Flashback, with increases in Acc@5, Acc@10, Acc@20 and MRR. Similarly, STAN+, CARAN+ and KGNext+ outperform their base models, STAN, CARAN, and KGNext, respectively, demonstrating enhanced accuracy in predicting trajectories. Specifically, STAN+ achieves the highest accuracy among all models, indicating the effectiveness of integrating our enhancement module with the STAN model. These results underscore the benefits of our proposed enhancement module in improving the predictive performance of existing trajectory prediction models. The details are as follows:

1. From Table 5, Flashback+ demonstrates significant improvements over Flashback, with increases of 0.45%, 1.81%, 2.33% and 2.10% in Acc@5, Acc@10, Acc@20 and MRR, respectively. Similarly, STAN+ and CARAN+ exhibit substantial enhancements compared to their base models. STAN+ achieves improvements of 3.68%, 5.09%, 4.87% and 5.46%, whereas CARAN+ shows enhancements of 2.55%, 1.21%, 1.58% and 6.09% in Acc@5, Acc@10, Acc@20 and MRR, respectively, when compared to STAN and CARAN. KGNext+ demonstrates a clear performance improvement over KGNext, with increases across all evaluation metrics: 12% in Acc@5, 17.3% in Acc@10, 11.2% in Acc@10% and 4.08% in MRR.
2. From Table 6, Flashback+ displays notable advancements relative to Flashback, boasting increases of 2.38%, 5.66%, 4.98% and 1.70% in Acc@5, Acc@10, Acc@20 and MRR, correspondingly. Similarly, STAN+ showcases significant enhancements compared to STAN, achieving improvements of 24.01%, 16.82%, 13.74% and 7.36%. Furthermore, CARAN+ exhibits substantial progress over CARAN, with gains of 21.07%, 21.58%, 10.32% and 7.25%, respectively. KGNext+ achieves a significant performance boost compared to KGNext, with improvements of 26.8% in Acc@5, 24.0% in Acc@10, 13.9% in Acc@20% and 4.15% in MRR.
3. From Table 7, Flashback+ displays significant performance gains over its base model, with notable increases of 12.63%, 9.90%, 3.97% and 1.04% in Acc@5, Acc@10, Acc@20 and MRR, respectively. Similarly, STAN+ demonstrates remarkable enhancements when compared to STAN, achieving substantial improvements of 28.27%, 28.54%, 26.06% and 6.49%. Additionally, CARAN+ exhibits substantial progress over CARAN, with gains of 28.51%, 25.94%, 25.74% and 4.83%. KGNext+ demonstrates a notable performance improvement over KGNext, with increases of 19.4% in Acc@5, 21.7% in Acc@10, 12.8% in Acc@20% and 4.41% in MRR, respectively.

TABLE 5 | Adapt to current methods in Boston.

	Acc@5	Acc@10	Acc@20	MRR
Flashback	0.1677	0.2432	0.3235	0.1324
Flashback+	**0.1722**	**0.2613**	**0.3468**	**0.1534**
STAN	0.1428	0.2124	0.3073	0.1033
STAN+	**0.1796**	**0.2633**	**0.3560**	**0.1579**
CARAN	0.1577	0.2302	0.3155	0.1141
CARAN+	**0.1832**	**0.2723**	**0.3713**	**0.1750**
KGNext	0.1442	0.2113	0.3060	0.1027
KGNext+	**0.1622**	**0.2478**	**0.3402**	**0.1435**

Note: Bold indicates the best-performing method.

TABLE 6 | Adapt to current methods in Honolulu.

	Acc@5	Acc@10	Acc@20	MRR
Flashback	0.1956	0.2581	0.3563	0.1615
Flashback+	**0.2203**	**0.2838**	**0.3705**	**0.1785**
STAN	0.1651	0.2234	0.2965	0.1219
STAN+	**0.2117**	**0.2873**	**0.3736**	**0.1955**
CARAN	0.1794	0.2323	0.3035	0.1287
CARAN+	**0.2304**	**0.2925**	**0.3816**	**0.2012**
KGNext	0.1732	0.2283	0.2911	0.1259
KGNext+	**0.2195**	**0.2831**	**0.3316**	**0.1674**

Note: Bold indicates the best-performing method.

5.7 | Ablation Study (Question 3)

We conducted an ablation study to assess the efficacy of key design components in our model. Specifically, we implemented three variants by removing different components, outlined as follows: (i) w/o TM model, which excludes the weather feature in the temporal-weather-aware module, (ii) w/o TEM model, which excludes the enhancement of temporal-weather-aware module, (iii) w/o UREM model, which excludes the enhancement of the user relationship module and (iv) w/o TEM + UREM, which eliminates the above modules and only keeps the POI recommendation module. The experiments were carried out on all datasets, and the results of the ablation study are presented in Tables 8–10. The key findings from the statistical analysis are as follows:

1. The full model attains the highest performance, showcasing its comprehensive effectiveness. On Boston, TWTransNet achieves 0.1811, 0.2669, 0.3621 and 0.1518 performance in terms of Acc@5, Acc@10, Acc@20 and MRR. Then, on Honolulu, our method obtains 0.2412, 0.3034, 0.3978 and 0.1775. And it gets 0.3435, 0.4147, 0.5251 and 0.2158 on Miami accordingly.

TABLE 7 | Adapt to current methods in Miami.

	Acc@5	Acc@10	Acc@20	MRR
Flashback	0.3213	0.3805	0.4733	0.1932
Flashback+	**0.3451**	**0.4023**	**0.4966**	**0.2036**
STAN	0.2641	0.3377	0.4414	0.1622
STAN+	**0.3277**	**0.3947**	**0.5018**	**0.2271**
CARAN	0.2813	0.3473	0.4635	0.1831
CARAN+	**0.3411**	**0.4223**	**0.5117**	**0.2314**
KGNext	0.2691	0.3412	0.4405	0.1654
KGNext+	**0.3213**	**0.4152**	**0.4968**	**0.2095**

Note: Bold indicates the best-performing method.

2. The w/o TEM module has a great decline compared with other baselines by solely focusing on enhancing trajectories. This suggests that leveraging the translation distance paradigm to enhance POI transition relationships by integrating time and weather information significantly aids in capturing personalised user transition patterns. More concretely, the w/o TEM decreases to 0.1598, 0.2388, 0.3326 and 0.1260 in terms of Acc@5, Acc@10, Acc@20 and MRR on Boston. Besides, it drops to 0.1978, 0.2533, 0.3462 and 0.1271 on Honolulu, while decreases to 0.2756, 0.3745, 0.4682 and 0.1603 on Miami.
3. The w/o TEM + UREM module, which excludes both the enhancement of user relationships and time-weather information, demonstrates the worst performance. This outcome unequivocally highlights the effectiveness of our model, emphasising the importance of both enhanced user relationships and trajectory information for optimal performance.

5.8 | Recommendation Analysis Under Sparse Trajectories (Question 4)

Addressing the challenge posed by sparse trajectories is imperative for robust POI recommendation. The limited number of check-ins in a trajectory poses difficulty in capturing users' nuanced preferences. To evaluate the model's performance under such constraints, we systematically sort user trajectories in the test set by length, designating the bottom 15% as sparse trajectories. We report the performance on the above sparse trajectories as shown in Tables 11–13. We have the following observations:

1. From Table 11, TWTransNet exhibits remarkable performance improvements over the second-best method, Flashback, in predicting sparse trajectories on the Boston dataset. Compared to Flashback, TWTransNet achieves substantial enhancements across all metrics. Specifically, TWTransNet demonstrates a 33.48% improvement in Acc@5, a 29.89% improvement in Acc@10 and a 38.33% improvement in Acc@20.
2. On the Honolulu dataset of Table 12, TWTransNet achieves the highest accuracy across all evaluation metrics, with Acc@5 of 0.0357, Acc@10 of 0.0535 and Acc@20 of 0.0892, outperforming Flashback, STAN and CARAN. Similarly, on the Miami dataset of Table 13, TWTransNet consistently outperforms the competing methods, achieving the highest accuracy metrics of Acc@5, Acc@10 and Acc@20, with values of 0.0516, 0.0738 and 0.1134, respectively. These results demonstrate the effectiveness of TWTransNet in accurately predicting sparse trajectories in urban environments. These results underscore the superior predictive capability of TWTransNet in handling sparse trajectory data.

TABLE 8 | Ablation results on Boston.

Boston	Acc@5	Acc@10	Acc@20	MRR
w/o TM	0.1724	0.2457	0.3441	0.1213
w/o TEM	0.1598	0.2388	0.3326	0.1260
w/o UREM	0.1720	0.2530	0.3436	0.1328
w/o TEM + UREM	0.1587	0.2262	0.3115	0.1057
TWTransNet	**0.1811**	**0.2669**	**0.3621**	**0.1518**

Note: Bold indicates the best-performing method.

TABLE 9 | Ablation results on Honolulu.

	Acc@5	Acc@10	Acc@20	MRR
w/o TM	0.2105	0.2638	0.3604	0.1420
w/o TEM	0.1978	0.2533	0.3462	0.1271
w/o UREM	0.2326	0.2891	0.3829	0.1644
w/o TEM + UREM	0.1805	0.2427	0.3069	0.0815
TWTransNet	**0.2412**	**0.3034**	**0.3978**	**0.1775**

Note: Bold indicates the best-performing method.

TABLE 10 | Ablation results on Miami.

	Acc@5	Acc@10	Acc@20	MRR
w/o TM	0.2859	0.3966	0.4831	0.1723
w/o TEM	0.2756	0.3745	0.4682	0.1603
w/o UREM	0.3245	0.4026	0.5013	0.1977
w/o TEM + UREM	0.2407	0.3043	0.3914	0.1164
TWTransNet	**0.3435**	**0.4147**	**0.5251**	**0.2158**

Note: Bold indicates the best-performing method.

TABLE 11 | Performance on sparse trajectories of Boston.

	Acc@5	Acc@10	Acc@20
Flashback	0.0343	0.0489	0.0663
STAN	0.0275	0.0383	0.0592
CARAN	0.0319	0.0467	0.0639
TWTransNet	**0.0458**	**0.0634**	**0.0917**

Note: Bold indicates the best-performing method.

TABLE 12 | Performance on sparse trajectories of Honolulu.

	Acc@5	Acc@10	Acc@20
Flashback	0.0312	0.0447	0.0758
STAN	0.0243	0.0336	0.0492
CARAN	0.0267	0.0401	0.0581
TWTransNet	**0.0357**	**0.0535**	**0.0892**

Note: Bold indicates the best-performing method.

TABLE 13 | Performance on sparse trajectories of Miami.

	Acc@5	Acc@10	Acc@20
Flashback	0.0434	0.0635	0.0914
STAN	0.0301	0.0511	0.0768
CARAN	0.0364	0.0568	0.0873
TWTransNet	**0.0516**	**0.0738**	**0.1134**

Note: Bold indicates the best-performing method.

3. Moreover, the performance trend of Acc@5 < Acc@10 < Acc@20 is commonly observed in recommendation systems due to the expanding range of evaluation. Acc@N measures whether the correct POI appears within the top-N recommendations. As N increases, the likelihood of including the correct POI in the candidate list naturally grows, leading to higher accuracy. Additionally, user preferences are diverse, and although the model may generate several plausible POI options, its ability to rank the correct one higher may be limited. Consequently, larger N values provide more opportunities to capture the correct POI, resulting in improved accuracy metrics.

5.9 | Impact of Neighbour Layers on Prediction Performance (Question 5)

To investigate the impact of the key parameter K (Equation (8)), we vary $K \in [2, 4, 6, 8]$ and report the results as follows:

1. From Figure 5, we observe fluctuations in performance metrics on Boston. As k increases, Acc@5 and Acc@10 reach their peak values at $K = 6$, followed by a slight decline. However, Acc@20 achieves its highest value at $K = 4$, with a minor decrease afterwards.
2. In Figure 6, as n increases, Acc@5, Acc@10 and Acc@20 initially exhibit improvements until reaching a peak at $K = 6$. Beyond this point, there is a slight decrease in performance, indicating a balance between capturing local and global information.
3. Similarly, in Figure 7, the highest accuracies are observed at $K = 6$, with values of 0.3435, 0.4147 and 0.5251 for Acc@5, Acc@10 and Acc@20, respectively. However, beyond $K = 6$, there is a slight decline in performance, with Acc@5, Acc@10 and Acc@20 dropping to 0.3143, 0.3946 and 0.4816, respectively, at $K = 8$.

To sum up, these fluctuations are stemmed from the change in neighbouring points caused by different K values, subsequently impacting the model's performance. The experimental results reveal that as the value of K gradually increases, the enhanced user relationships and POI transition patterns help the model better capture users' visit preferences and personalised POI transition behaviours, thereby improving the recommendation performance. However, when K becomes too large, the recommendation performance starts to decline instead of continuing to improve. This is likely because an excessively large K introduces additional noise, which degrades the model's performance. Furthermore, it can be observed that the Miami and Boston datasets are more sensitive to the choice of K; their recommendation performance drops significantly when K is too large. Based on these observations, K should not be set too high, and we suggest that the optimal values for K are 4 or 6. And our method achieves best performance when $K = 6$ on all datasets.

5.10 | Deployment of TWTransNet in Real-World Recommendation Systems (Question 6)

The deployment of TWTransNet in real-world POI recommendation systems requires addressing the following key aspects:

1. Real-time data pipeline. Implement weather API integration and temporal context extraction through device clocks. Develop lightweight embedding servers to handle POI/time/weather feature vectors. Design incremental graph updates using edge computing to maintain fresh transition patterns.
2. Hybrid Architecture. Combine offline pretraining of the core translation paradigm (Equations 1–3) with the online adaptation of user relationship graphs (Equations 4–7).

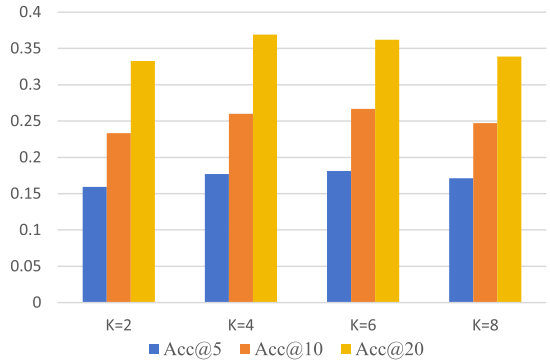


FIGURE 5 | Performance on Boston.

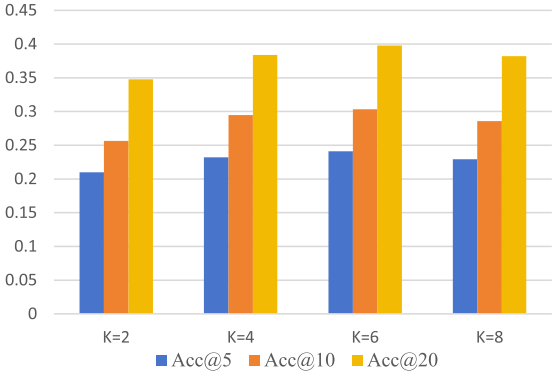


FIGURE 6 | Performance on Honolulu.

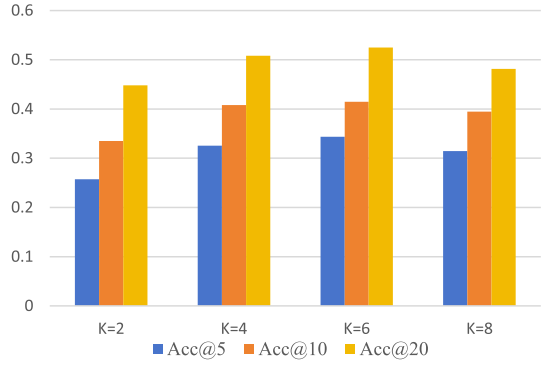


FIGURE 7 | Performance on Miami.

Deploy spatial-temporal attention (Equations 12 and 13) as microservice for real-time scoring. Use Redis for caching frequent POI transitions under common weather-time combinations.

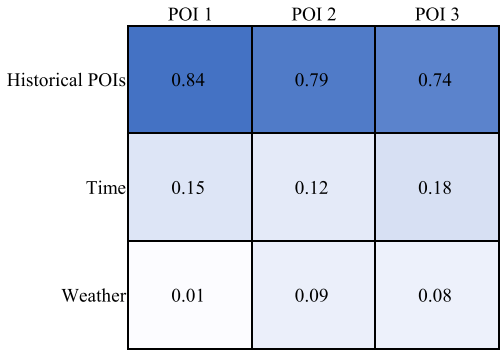
5.11 | Case Studies

The case studies, as shown in Figure 8, analyse the impact of sunny and rainy weather on a user's next POI decision. In these cases, 'Historical POIs' represents the influence probability of previously visited POIs, whereas 'Time' and 'Weather' indicate the respective influence probabilities of time and weather on the decision-making process.

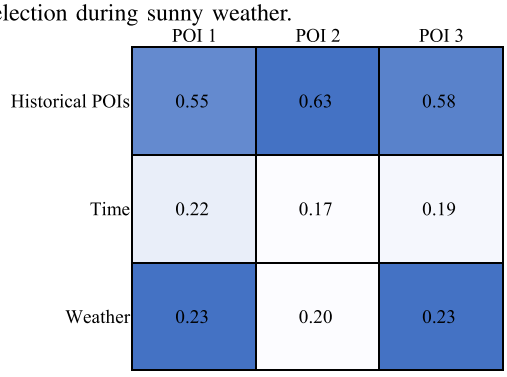
Figure 8a illustrates that on sunny days, a user's decisions are primarily influenced by their historical preferences, with the historical POIs carrying a higher weight in the decision-making process. In contrast, Figure 8b shows that on rainy days, user decisions are more significantly impacted by weather features. This highlights the importance of considering multiple feature sources to better understand users' POI decision-making behaviours.

6 | Conclusion

In this paper, we propose a new solution for POI recommendation. Our approach is anchored in a meticulous construction of a POI transition graph, dynamically enhanced by the integration of time and weather information through the translation distance paradigm. This innovative fusion deepens the understanding of the intricate relationship between POI transitions and time/weather conditions within user trajectories. To address data sparsity challenges, we introduce a user-POI interaction graph, harnessing graph neural networks to model user preferences. Further refinement of user preference relationships is achieved through similarity functions. The seamless integration of enhanced POI transition and user preference relationships culminates in the application of advanced spatio-temporal self-attention methods, effectively capturing nuanced user transition patterns. The deployment of attention mechanisms in our approach ensures the delivery of personalised POI recommendations. The empirical results on three real datasets consistently highlight the superior performance of our proposed method. Ablation experiments and in-depth analyses further validate the effectiveness of each module within our method.



(a) Influence of different features on a user's next POI selection during sunny weather.



(b) Influence of different features on a user's next POI selection during rainy weather.

FIGURE 8 | Influence of multiple features on next POI selection.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The authors have nothing to report.

References

1. B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning Geographical Preferences for Point-of-Interest Recommendation," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2013), 1043–1051.

2. M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "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* (ACM, 2011), 325–334.

3. R. Li, Y. Shen, and Y. Zhu, "Next Point-of-Interest Recommendation With Temporal and Multi-Level Context Attention," in *2018 IEEE International Conference on Data Mining (ICDM)* (IEEE, 2018), 1110–1115.

4. B. Chang, G. Jang, S. Kim, and J. Kang, "Learning Graph-Based Geographical Latent Representation for Point-of-Interest Recommendation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (ACM, 2020), 135–144.

5. G. Suganeshwari and S. Syed Ibrahim, "A Survey on Collaborative Filtering Based Recommendation System," in *Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16')* (Springer, 2016), 503–518.

6. Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the Next Location: A Recurrent Model With Spatial and Temporal Contexts," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30, No. 1 (AAAI, 2016), https://doi.org/10.1609/aaai.v30i1.9971.

7. Y. Zhang, H. Dai, C. Xu, et al., "Sequential Click Prediction for Sponsored Search With Recurrent Neural Networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28, No. 1 (AAAI, 2014), https://doi.org/10.1609/aaai.v28i1.8917.

8. J. Zhang, X. Liu, X. Zhou, and X. Chu, "Leveraging Graph Neural Networks for Point-of-Interest Recommendations," *Neurocomputing* 462 (2021): 1–13, https://doi.org/10.1016/j.neucom.2021.07.063.

9. N. Lim, B. Hooi, S.-K. Ng, et al., "STP-UDGAT: Spatial-Temporal-Preference User Dimensional Graph Attention Network for Next POI

Recommendation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (ACM, 2020), 845–854.

10. D. Lian, Y. Wu, Y. Ge, X. Xie, and E. Chen, "Geography-Aware Sequential Location Recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, 2020), 2009–2019.

11. M. B. Hossain, M. S. Arefin, I. H. Sarker, M. Kowsher, P. K. Dhar, and T. Koshiba, "CARAN: A Context-Aware Recency-Based Attention Network for Point-of-Interest Recommendation," *IEEE Access* 10 (2022): 36299–36310, https://doi.org/10.1109/access.2022.3161941.

12. Y. Luo, Q. Liu, and Z. Liu, "STAN: Spatio-Temporal Attention Network for Next Location Recommendation," in *Proceedings of the Web Conference 2021* (ACM, 2021), 2177–2185.

13. Y. Li, T. Chen, Y. Luo, H. Yin, and Z. Huang, "Discovering Collaborative Signals for Next POI Recommendation With Iterative Seq2graph Augmentation," *arXiv preprint arXiv:2106.15814* (2021): 1491–1497, https://doi.org/10.24963/ijcai.2021/206.

14. D. Yang, B. Fankhauser, P. Rosso, and P. Cudre-Mauroux, "Location Prediction Over Sparse User Mobility Traces Using RNNS," in *Proceedings of the 29 International Joint Conference on Artificial Intelligence* (IJCAI, 2020), 2184–2190.

15. S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing Personalized Markov Chains for Next-Basket Recommendation," in *Proceedings of the 19th International Conference on World Wide Web* (ACM, 2010), 811–820.

16. P. Zhao, A. Luo, Y. Liu, et al., "Where To Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation," *IEEE Transactions on Knowledge and Data Engineering* 34, no. 5 (2020): 2512–2524, https://doi.org/10.1109/tkde.2020.3007194.

17. K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, and H. Yin, "Where To Go Next: Modeling Long-and Short-Term User Preferences for Point-of-Interest Recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 1 (AAAI, 2020), 214–221, https://doi.org/10.1609/aaai.v34i01.5353.

18. J. Feng, Y. Li, C. Zhang, et al., "DeepMove: Predicting Human Mobility With Attentional Recurrent Networks," in *Proceedings of the 2018 World Wide Web Conference* (ACM, 2018), 1459–1468.

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

20. Q. Guo, Z. Sun, J. Zhang, and Y.-L. Theng, "An Attentional Recurrent Neural Network for Personalized Next Location Recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 1 (AAAI, 2020), 83–90, https://doi.org/10.1609/aaai.v34i01.5337.

21. C. Zuo, X. Zhang, L. Yan, and Z. Zhang, "GUGEN: Global User Graph Enhanced Network for Next POI Recommendation," *IEEE Transactions on Mobile Computing* 23, no. 12 (2024): 14975–14986, https://doi.org/10.1109/tmc.2024.3455107.

22. M. Acharya and K. K. Mohbey, "Time-Aware Cross-Domain Point-of-Interest Recommendation in Social Networks," *Engineering Applications of Artificial Intelligence* 139 (2025): 109630, https://doi.org/10.1016/j.engappai.2024.109630.

23. J. Fu, R. Gao, Y. Yu, et al., "Contrastive Graph Learning Long and Short-Term Interests for POI Recommendation," *Expert Systems with Applications* 238 (2024): 121931, https://doi.org/10.1016/j.eswa.2023.121931.

24. B. A. Duncan, S. Kallumadi, T. Berg-Kirkpatrick, and J. Mcauley, "MAWI Rec: Leveraging Severe Weather Data in Recommendation," in *Proceedings of the 18th ACM Conference on Recommender Systems* (ACM, 2024).

25. A. Noorian, "A Personalized Context and Sequence Aware Point of Interest Recommendation," *Multimedia Tools and Applications* 83, no. 32 (2024): 77565–77594, https://doi.org/10.1007/s11042-024-18522-3.

26. J. Chen, W. Jiang, J. Wu, K. Li, and K. Li, "Dynamic Personalized POI Sequence Recommendation With Fine-Grained Contexts," *ACM Transactions on Internet Technology* 23, no. 2 (2023): 1–28, https://doi.org/10.1145/3583687.

27. Q. Guo, F. Zhuang, C. Qin, et al., "A Survey on Knowledge Graph-Based Recommender Systems," *IEEE Transactions on Knowledge and Data Engineering* 34, no. 8 (2020): 3549–3568, https://doi.org/10.1109/tkde.2020.3028705.

28. Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph Embedding by Translating on Hyperplanes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28, No. 1 (AAAI, 2014), https://doi.org/10.1609/aaai.v28i1.8870.

29. J. Wu, X. He, X. Wang, et al., "Graph Convolution Machine for Context-Aware Recommender System," *Frontiers of Computer Science* 16, no. 6 (2022): 166614, https://doi.org/10.1007/s11704-021-0261-8.

30. M. Unger and A. Tuzhilin, "Hierarchical Latent Context Representation for Context-Aware Recommendations," *IEEE Transactions on Knowledge and Data Engineering* 34, no. 7 (2020): 3322–3334, https://doi.org/10.1109/TKDE.2020.3022102.

31. N. R. Chopde and M. Nichat, "Landmark Based Shortest Path Detection by Using A* and Haversine Formula," *International Journal of Innovative Research in Computer and Communication Engineering* 1, no. 2 (2013): 298–302, https://www.researchgate.net/profile/Mangesh-Nichat-2/publication/282314348_Landmark_based_shortest_path_detection_by_using_A_Algorithm_and_Haversine_Formula/links/56389bb708ae4bde5021b0f5/Landmark-based-shortest-path-detection-by-using-A-Algorithm-and-Ha.

32. J. Schmidhuber and S. Hochreiter, "Long Short-Term Memory," *Neural Computation* 9, no. 8 (1997): 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

33. X. Kong, Z. Chen, J. Li, J. Bi, and G. Shen, "KGNext: Knowledge-Graph-Enhanced Transformer for Next POI Recommendation With Uncertain Check-Ins," *IEEE Transactions on Computational Social Systems* 11, no. 5 (2024): 6637–6648, https://doi.org/10.1109/tcss.2024.3396506.