

Modeling Spatiotemporal Patterns for Next-POI Recommendation: A Unified Approach Using Graph Embedding and Self-Attention

Yu-Da Huang *, Chih-Chung Lo **, Chieh-Yuan Tsai *

* Department of Industrial Engineering and Management, Yuan Ze University, Taiwan, R.O.C.

** Department of Applied Informatics, Fo Guang University, Taiwan, R.O.C.

s1115421@mail.yzu.edu.tw, locc@mail.fgu.edu.tw, cytsai@saturn.yzu.edu.tw

Abstract— In recent years, the development of social media has made self-guided travel more convenient. However, the overwhelming amount of social media information has also made travel planning more challenging. Thus, point-of-interest (POI) recommendation systems have received increasing attention, particularly those leveraging users' historical check-in records. Despite notable advancements, existing approaches face several key limitations. First, most models concentrate on individual user trajectories, overlooking complex inter-POI relationships. Second, although temporal dynamics are considered, prior work emphasizes short-term variations while neglecting broader periodic patterns. Third, spatial dependencies are frequently oversimplified, with insufficient attention to specific geographic zones. Fourth, while deep learning methods have improved predictive performance, they still struggle to capture cyclical temporal trends. To address these challenges, this study proposes a next-POI recommendation system that integrates graph embeddings to model complex and interactive relationships among POIs. In addition, the system encodes user activities, considering both short-term and long-term temporal behavioral patterns. Moreover, we enhance spatial representation by employing a UTM (Universal Transverse Mercator) grid-based method, which more effectively captures geographic dependencies than traditional GPS coordinates. Finally, a transformer-based approach with a self-attention mechanism analyzes user check-in sequences, capturing complex behavioral patterns for more accurate and personalized recommendations. Experimental results demonstrate that incorporating graph and grid-based spatial features substantially improves prediction accuracy. Additionally, the proposed system outperforms both sequential and non-sequential baselines across multiple evaluation metrics, confirming the benefits of the proposed next-POI recommendation system.

Keywords— POI Recommendation Systems, Graph Embedding, Temporal Features, Grid-based Spatial Encoding, Transformer.

I. INTRODUCTION

With the growing popularity of travel, tourists increasingly rely on digital platforms to explore information about destinations. However, the rapid expansion of the internet and social media has led to an overwhelming influx of information, making it difficult for users to identify attractions that align with their interests. This phenomenon of information overload underscores the need for effective filtering mechanisms. In response, the next point-of-interest (POI) recommendation

systems have emerged as promising solutions that aim to capture contextual patterns within users' historical behavior to deliver personalized suggestions. Despite their potential, these systems face substantial challenges in extracting meaningful features from complex and dynamic user behavior data.

First, many current models, for instance, emphasize direct user-POI interactions without fully leveraging the underlying graph structures connecting different POIs [1]. As a result, they often fail to capture latent associations that may emerge through indirect relationships—for example, when two POIs share frequent co-visitation patterns despite lacking direct user overlap. Second, temporal modeling in existing systems typically segments daily activity into discrete time slots to capture variations in user preferences [2-4]. While such approaches effectively capture short-term patterns, they often overlook longer-term periodic behaviors, such as seasonal trends or monthly preferences, which can be equally influential in shaping user choices. Similarly, spatial modeling also faces challenges. Many systems rely on simple distance-based metrics to assess spatial proximity [5,6]. However, these methods often neglect the relative positioning and density of POIs within specific geographic regions. As a result, spatial dependencies may be oversimplified, leading to suboptimal location-based recommendations.

Early recommendation approaches primarily relied on traditional techniques such as matrix factorization and Markov chains. However, the increasing complexity of user data has prompted a shift toward deep learning models, which automatically learn hierarchical feature representations and reduce the need for manual feature engineering [7,8]. Deep learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks have been adopted to address spatial and temporal complexities. CNNs are adept at learning spatial hierarchies, while RNNs and LSTMs excel at capturing sequential patterns [9]. Nevertheless, these models also present limitations. CNNs are not inherently designed to manage sequential dependencies, and RNNs and LSTMs often struggle with long-term dependencies and complex feature interactions [10].

To overcome these limitations, this study proposes a novel next POI recommendation system that builds on users' historical check-in sequences. The system begins by leveraging

graph embedding techniques to model interactive relationships among POIs based on co-visitation patterns. We construct a POI interaction relationship graph and utilize the DeepWalk method to derive a more comprehensive representation of POI connectivity from the graph. Temporal dynamics are modeled across multiple time scales to capture both short-term behaviors and long-term trends. This multi-level temporal modeling enables the system to detect variations in user preferences over time. To enhance spatial judgement, this study adopts a grid-based spatial representation. POI coordinates are converted from GPS to the UTM coordinate system, which is then segmented into fine-grained grid units. Spatial features are generated by calculating the distances between POIs and the centers of their respective grid cells, thereby capturing both relative position and local density within specific regions. Finally, a transformer-based approach with a self-attention mechanism analyzes user check-in sequences, capturing complex behavioral patterns for more accurate and personalized recommendations. This approach addresses the shortcomings of CNNs, RNNs, and LSTMs by offering a unified and flexible framework for modeling user trajectories and preferences.

II. LITERATURE REVIEW

A. Temporal and Spatial Dynamics in Recommendation Systems

Recent advancements in POI recommendation systems have highlighted the importance of integrating temporal and spatial information to more accurately reflect dynamic user behaviors. Traditional models often treat user preferences as static, failing to account for temporal variability and geographical context. In contrast, time- and location-aware frameworks adapt to users' evolving interests and mobility patterns, enabling more personalized and context-sensitive recommendations.

Temporal models embed time as a key feature, capturing behavioral patterns such as daily routines, seasonal trends, and shifting interests. Methods range from time-aware collaborative filtering [11] and continuous-time network embeddings [12] to neural architectures such as CNNs [13], RNNs with attention [14], and time-aware GNNs [15]. These models improve recommendation accuracy by recognizing short- and long-term temporal dependencies [16,17].

In parallel, spatial recommendation systems leverage user location data to improve relevance. Early approaches incorporated social and geographic proximity [18], while more recent models employ probabilistic embeddings [19], autoencoders [20], and trajectory prediction networks [21]. Neural embeddings of GPS coordinates [22,23] further enhance spatial representation, while hybrid models [24] dynamically integrate spatial and temporal contexts.

Together, these studies demonstrate that modeling temporal and spatial dimensions is essential for effective POI recommendation. Building on this foundation, our study introduces a transformer-based model that unifies graph embeddings, multi-scale temporal patterns, and grid-based spatial representations to advance next POI prediction.

B. Deep Learning-Based Recommendation Systems

Deep learning has revolutionized recommendation systems by enabling models to automatically learn complex, non-linear relationships from high-dimensional data, surpassing traditional approaches that rely on handcrafted features [8]. The introduction of the transformer architecture by [25], with its self-attention mechanism and parallel training capabilities, marked a significant advancement in modeling long-range dependencies. Subsequent research has extended attention-based architectures to recommendation tasks. [26] improved factorization machines using attention to highlight key feature interactions, while [27] proposed STAMP to capture short-term user intent in session-based recommendations. [1] introduced SASRec, a sequential model that leverages self-attention to balance long-term and short-term preferences, outperforming RNN- and Markov-based models.

Transformer variants have further advanced recommendations in conversational and spatio-temporal contexts. Models such as the conversational recommender [28] and GETNext [29] demonstrate improved adaptability by integrating trajectory flows, user intent, and contextual information. [30] further enriched POI prediction with a multi-modal transformer that captures deep sequence and cross-modal dependencies. To reduce computational cost, [31] proposed feature-based POI grouping with transformer encoding, enhancing efficiency for large-scale systems. Addressing explainability, [32] introduced a self-explainable Transformer-based framework that utilizes a variational information bottleneck, thereby enabling intrinsic interpretability and fostering user trust. More related studies can be found in [33-36]. Overall, these studies highlight the capacity of deep learning to integrate temporal, spatial, and semantic signals within a unified predictive framework, thereby advancing both accuracy and interpretability in modern recommendation systems.

III. METHODOLOGY

The framework of the proposed POI recommendation system is visually illustrated in Figure 1. First, user check-in sequences and POI information are used as the input to the system. In the representation layer, we derive the latent representations of POIs. Four essential features, including the interaction relationship feature, temporal feature, category feature, and spatial feature, are considered when building a POI latent representation. In the embedding layer, the latent representations of POIs are mapped to dense POI embeddings. Next, the user check-in sequences and POI embeddings will serve as inputs to the Transformer encoder, which utilizes a self-attention mechanism to capture complex behavioral patterns for more accurate and personalized recommendations. The transformer encoder features a primary structure comprising multiple self-attention blocks, each consisting of one self-attention mechanism and two feedforward networks. After processing several self-attention blocks, the next POI with the highest probability will be returned in the prediction layer.

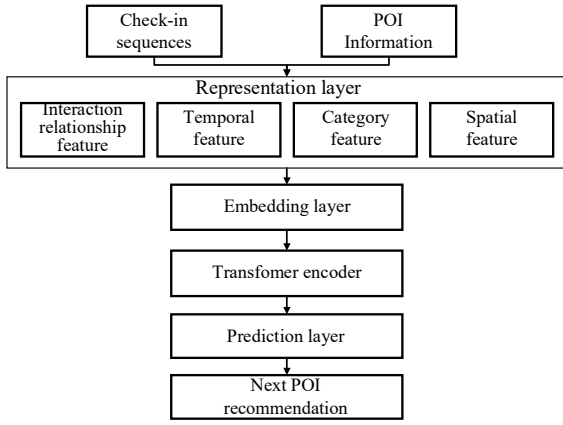


Figure 1. The framework of the proposed POI recommendation system

A. Data description

A check-in record can be represented as $r = \langle u, poi, ct \rangle$ where $u \in U = \{u_1, u_2, \dots, u_{|U|}\}$ is the set of users, $poi \in POI = \langle poi_1, poi_2, \dots, poi_{|POI|} \rangle$ is the set of POIs, and ct represents the check-in time. A POI can be represented as $poi = \langle pid, ca, la, lo \rangle$ where pid is the POI ID, ca is the category for the POI, la and lo are the latitude and longitude of the POI. A check-in sequence can be represented as $S = \langle r_1, r_2, \dots, r_n \rangle$ where r_n indicates n -th check-in record, and $r_1.ct \leq r_2.ct \leq \dots \leq r_n.ct$.

B. Representation layer

1) Interactive relationship feature

To better understand the interactive relationships among POIs, we first establish a POI interactive relationship graph. Based on check-in sequences of all users, connections between two POIs can be established if they are accessed consecutively within a sequence. Weights are assigned to edges based on the frequency of transitions between POIs in user check-in sequences. This weight reflects how often poi_i transitions to poi_j across all users' check-in sequences. Consequently, the resulting POI relationship graph captures the similarity between different POIs.

After obtaining the POI relationship graph, the DeepWalk method [37] is employed to extract similarities among various nodes (POIs). Inspired by the ideas of Word2Vec, DeepWalk transforms each POI in the POI relationship graph to a low-dimensional vector space, capturing the contextual similarity between POIs in feature vectors. These feature vectors can be utilized to analyze relationships between different POIs, identify user behavior patterns, and perform recommendation tasks. Let the POI graph be $G = (V, E)$ where V represents the set of nodes and E represents the set of edges. The objective of graph embedding is to learn a low-dimensional mapping function $\Phi: V \rightarrow \mathbb{R}^d$, which means each node in V is represented as a d -dimensional vector, and the dimensionality $d \ll |V|$.

The DeepWalk generates new sequences through a random walk, which starts from a particular initial node, randomly selects a neighboring node in the graph to move to, and repeats

this process until a specified number of steps is reached. The probability of a random walk of choosing another node is defined as [21]:

$$P(v_j|v_i) = \begin{cases} \frac{M_{ij}}{\sum_{j \in N^+(v_i)} M_{ij}}, & v_j \in N^+(v_i) \\ 0, & e_{ij} \notin E \end{cases} \quad (1)$$

where v_i represents the current node, v_j denotes a neighboring node of v_i , M_{ij} represents the weight of the connection from node v_i to node v_j , $N^+(v_i)$ represents the set of connected neighbours of v_i , e_{ij} represents the edge between nodes v_i and v_j , E denotes the set of edges in the graph $G = (V, E)$.

Next, the DeepWalk treats each random walk sequence as a sentence and trains it using the Skip-gram method. By maximizing the probability of two nodes co-occurring, the use of this method leads to the following optimization problem [21]:

$$\text{minimize}_{\Phi} -\log Pr(\{v_{i-w}, \dots, v_{i+w}\} \setminus v_i | \Phi(v_i)) \quad (2)$$

where Φ represents the parameters to be optimized. $\setminus$ indicates the exclusion of v_i from the context nodes. The objective is to minimize the negative log probability of observing the context sequence $\{v_{i-w}, \dots, v_{i+w}\}$ given the central node v_i conditioned on the representation $\Phi(v_i)$. The context sequence consists of nodes within a window size w centered around the node v_i . After completing the above process, the interactive relationship feature of poi_i is represented as $IRf_i \in \mathbb{R}^n$.

2) Temporal feature

Three temporal levels, day, week, and month, are included in encoding the temporal characteristics of a POI. First, we divide one day into eight intervals, each lasting three hours. Therefore, the temporal trend in terms of a day for poi_i can be represented as $Td_i = \langle td_{i1}, td_{i2}, \dots, td_{i8} \rangle$ where td_{ij} is the visit count in time interval j for poi_i . Second, to encode the temporal trend in terms of the day of the week for poi_i , we have $Tw_i = \langle tw_{i1}, tw_{i2}, \dots, tw_{i7} \rangle$ where tw_{i1} to tw_{i7} correspond to the visit count from Monday to Sunday. Third, the temporal trend in terms of months for poi_i can be represented as $Tm_i = \langle tm_{i1}, tm_{i2}, \dots, tm_{i12} \rangle$ where tm_{i1} to tm_{i12} correspond to the visit count for each month of a year. Finally, the spare temporal vector of poi_i can be represented as $Tf_i = \langle Td_i, Tw_i, Tm_i \rangle$.

3) Category feature

Typically, a POI might belong to specific categories. A set of categories is defined first to utilize the category information. Since a POI might belong to multiple categories, the spare category feature for poi_i can be represented as $Cf_i = \langle ca_{i1}, ca_{i2}, \dots, ca_{in} \rangle$ where ca_{ij} is set to 1 if poi_i belongs to category j , otherwise, it is set to 0. The example categories include food, restaurants, shopping, interior design, nightlife, hair salons, beauty & spas, event planning & services, and so on.

4) Spatial feature

Precise geographical location representation for POIs is essential for POI recommendation. This study examines a grid-based method that utilizes a high-dimensional vector representation to capture geographic location information. First, the latitude and longitude coordinates of POIs are transformed into UTM (Universal Transverse Mercator) coordinates [38]. The UTM system divides the Earth's surface into 60 longitude zones and 20 latitude bands [39].

Next, the UTM regions containing the target POI are identified. For each identified UTM region, the region will be divided into $m \times m$ rectangular grids where each grid has a central point. We can use the identified UTM region's four corner points to determine each rectangular grid's boundaries. Then we calculate the UTM coordinates of the center point of each grid based on these boundaries. Instead of using all 60×20 UTM regions to represent POIs, this study uses only the identified UTM regions. Thus, the computational cost is reduced.

Third, the temporal feature of a POI can be calculated for all grid center points. Let $Z = \{g_{ij} | i, j = 1, 2, \dots, m\}$ be the grid and $c_{ij} = (x_{ij}, y_{ij})$ be the UTM coordinates of the center point for grid g_{ij} , and the UTM coordinates of a POI be $l = (l_u, l_v)$. Then, the feature of the POI can be calculated as (Yin et al. 2019):

$$e_{ij}^l = \exp\left(-\frac{|l - c_{ij}|^2}{\sigma_{ij}}\right) \quad (3)$$

where $|l - c_{ij}|$ represents the Euclidean distance between UTM coordinates l and c_{ij} , σ_{ij} denotes a decay coefficient for each grid g_{ij} . The spatial feature of a POI can be represented as $E^l = (e_{11}^l, e_{12}^l, \dots, e_{1m}^l, \dots, e_{m1}^l, \dots, e_{mm}^l)$ where the length of E^l is $m \times m$. To make the following discussion easier, the spatial feature of poi_i is represented as $Sf_i \in R^{m \times m}$.

5) Integrated POI feature

To enhance the accuracy of personalized POI recommendations, this research incorporates all the above features, including the interactive relationship feature, temporal feature, category feature, and spatial feature, as the integrated POI feature. That is, the integrated POI feature for poi_v is:

$$If_v = \text{Concat}(IRf_v, Tf_v, Cf_v, Sf_v) \quad (4)$$

where IRf_v represents the interactive relationship feature of poi_v , Tf_v represents the temporal feature of poi_v , Cf_v represents the category feature of poi_v , Sf_v represents the spatial feature of poi_v .

C. Embedding layer

For each $poi \in POI$, the POI embedding layer maps the integrated POI feature If_i in Eq. (4) to a dense vector representation $H_i \in \mathbb{R}^d$. The formula of POI embedding is $H_i = W \cdot If_i$ where W denotes the learnable weight trained together with the model.

D. Transformer encoder

Next, the user check-in sequences and POI embeddings will serve as inputs to the Transformer encoder, which utilizes a self-attention mechanism to capture complex behavioral

patterns for more accurate and personalized recommendations [1]. First, the POI embedding H_v is updated by adding the position embedding as $\hat{H}_v$. The updated embedding is represented as follows:

$$\hat{H}_v = \begin{bmatrix} H_{v1} + P_1 \\ H_{v2} + P_2 \\ \dots \\ H_{vn} + P_n \end{bmatrix} \quad (5)$$

where H_v represents POI embeddings, P_n represents the positional embeddings. In this study, sine and cosine functions are used to generate these position embeddings based on the position pos in the sequence and the index i of the embedding dimension. The i -th dimension of the embedding vector at position pos can be represented as:

$$P_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \quad (6)$$

$$P_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \quad (7)$$

where pos represents the position index of the embedding vector, i represents the dimension index of the embedding vector, and d represents the dimension of the embedding vector.

The input of the self-attention layer is $\hat{H}_p$ where $\hat{H}_p$ will be projected using linear projection into a matrix and then input into the attention layer, which is defined as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (8)$$

where Q , K , and V represent queries, keys, and values. $\sqrt{d}$ serves as a scaling factor to prevent overly large dot products in high-dimensional spaces.

For multi-head self-attention, multiple attention heads are used in parallel to capture different aspects of dependencies. The output of the multi-head attention mechanism is defined as follows [25]:

$$S = SA(\hat{H}_v) = \text{MultiHead}(Q, K, V) \\ = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O \quad (9)$$

where each attention head is computed as:

$$\text{head}_i = \text{Attention}(\hat{H}_v W^Q, \hat{H}_v W^K, \hat{H}_v W^V) \quad (10)$$

where W^Q , W^K , $W^V \in \mathbb{R}^{d \times d_h}$ are learned projection matrices for each head, and $W^O \in \mathbb{R}^{hd_h \times d}$ is the output projection matrix. d_h is the dimensionality of each head, and h is the number of heads.

Since the self-attention model operates linearly, consideration must be given to potential interactions across different dimensions. Consequently, a two-layer feedforward neural network is assigned to all [1]:

$$F_i = \text{FFN}(S_i) = \text{ReLU}(S_i W^{(1)} + b^{(1)})W^{(2)} + b^{(2)} \quad (11)$$

where $W^{(1)}$ and $W^{(2)}$ are $d \times d$ weight matrices, $b^{(1)}$ and $b^{(2)}$ are bias vectors, each of which is d -dimensional. S_i is the i th output from formula (8).

After establishing the first self-attention block, F_i aggregates embeddings of all items, and stacking more self-attention blocks can help the model to learn more complex item embeddings. The b th self-attention block is defined as:

$$S^{(b)} = SA(F^{(b-1)}) \quad (12)$$

$$F_i^{(b)} = FFN(S_i^{(b)}), \forall i \in \{1, 2, \dots, n\} \quad (13)$$

where the first self-attention block is defined as:

$$S^{(1)} = S$$

$$F^{(1)} = F$$

As the number of network layers increases, several issues gradually arise. Increased model capacity may lead to overfitting, resulting in unstable training processes and longer training times. Therefore, it's necessary to employ the following operations to address these issues:

$$g'(x) = x + \text{Dropout}(g(\text{LayerNorm}(x))) \quad (14)$$

where $g(x)$ represents the self-attention layer or feedforward network. Before passing x to g , it is necessary to normalize the input x with layer normalization, apply dropout to the output of g , and finally add the input x to obtain the final output.

Layer normalization is defined as:

$$\text{LayerNorm}(x) = \alpha \odot \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (15)$$

where $\odot$ denotes element-wise multiplication, μ and σ are the mean and variance of x , α and β are the learned scaling factor and bias term. ϵ represents a tiny constant to prevent division by zero errors during computations.

E. Prediction layer

After processing adaptive and hierarchically extracted historical information through self-attention blocks, the next POI is predicted based on $F_t^{(b)}$, given the previous t POIs. The relationship of the i th POI is predicted using the following operation:

$$r_{i,t} = F_t^{(b)} N_i^T \quad (16)$$

where $r_{i,t}$ represents the relationship of the i th POI becoming the next destination given the previous t POIs, and $N \in \mathbb{R}^{|\mathcal{L}| \times d}$ is a POI embedding matrix. A higher interaction score $r_{i,t}$ indicates a stronger relevance of the POI, and recommendations are generated for users based on the ranking of interaction scores.

IV. IMPLEMENTATION AND EXPERIMENT RESULTS

A. Dataset

In this study, we utilize the Yelp dataset as the foundation for our experiments. This dataset, which is 4GB in size, is derived from real-world user activity data, including detailed business information such as names, categories, addresses, geolocations, operating hours, and average ratings. The dataset spans from December 30, 2009, to January 19, 2022, comprising 13,356,875 check-in records. With its extensive content, the dataset includes over 150,000 businesses. To demonstrate the benefits of the proposed recommendation system, we focus on the records from Pennsylvania and filter POIs to include only those belonging to the top ten most frequent categories. Furthermore, for the user check-in sequences, each sequence contains at least two POI visit records. After this filtering process, the final dataset includes 273,104 records, covering 16,074 unique users and 823 POIs. Table 1 shows example check-in records in the dataset.

TABLE 1. EXAMPLES OF USER CHECK-IN RECORDS IN THE DATASET

UserID	POIID	Category	Latitude	Longitude	Check-in Time
U0001	P0238	Restaurants	39.96036	-75.6066	2014/9/18 15:56
U0002	P0051	Restaurants	40.23217	-75.2276	2015/9/25 15:36
U0003	P0382	Interior Design, Shopping	40.29746	-75.0823	2017/11/3 21:16
U0004	P0638	Hair Salons, Beauty & Spas	39.94663	-75.1748	2016/3/12 17:51
U0005	P0516	Nightlife, Bars	40.02501	-75.2231	2013/1/6 04:11

The experiment was conducted in a Windows 10 environment with a 2.60 GHz Intel i5-11400F CPU, an 8 GB Nvidia GeForce RTX 3060 Ti GPU, and 32 GB of RAM.

B. Evaluation metrics

In the domain of POI recommendation systems, it is common to use multiple evaluation metrics to measure the performance of a model. In this study, we use Accuracy (ACC@K), Hit Rate (HR@K), Mean Reciprocal Rank (MRR@K), and Normalized Discounted Cumulative Gain (NDCG@K) to evaluate a model's effectiveness [1]. Each metric captures a different aspect of recommendation performance.

C. Comparison with baseline methods

To demonstrate the effectiveness of the proposed model, we compare its recommendation results with those of other well-known models. The comparison includes two categories: traditional non-sequential methods (e.g., PopRec, ItemKNN, BPR) and sequential methods (e.g., FPMC, STAMP, GRU4Rec, SASRec). The parameter settings for these baseline models are chosen based on commonly used configurations in the literature on recommendation systems. The sequence length of 20 is selected to effectively capture short-term sequential dependencies in POI recommendation tasks, aligning with previous studies. Similarly, Adam is chosen as the optimizer due to its robustness in handling gradient updates, and dropout rates are set at 0.2 to mitigate overfitting, a standard practice in neural recommendation models.

Tables 2 and 3 present the performance of all traditional non-sequential methods and sequential recommendation models at various K values. Traditional methods consistently lag behind sequential models across the four evaluation metrics at various K values. PopRec performs the worst, indicating that recommendations solely based on item popularity fail to address users' specific needs or sequential behavior patterns. In contrast, ItemKNN and BPR significantly improve compared to PopRec, especially at larger K values. However, all non-sequential models fail to consider temporal dependencies in user behavior, resulting in performance that still falls short of that of sequential models.

Sequential models demonstrate significant advantages across all metrics, highlighting their effectiveness in capturing temporal and sequential patterns. Among them, FPMC, which combines Markov chains with matrix factorization, captures short-term sequential dependencies and thus outperforms traditional non-sequential methods. STAMP combines long-term memory and short-term attention mechanisms. It dynamically captures users' interests and is sensitive to short-

term behavior patterns. This enables it to deliver stable performance at lower K values. GRU4Rec is based on Gated Recurrent Units (GRU) and effectively models sequential dependencies in user behavior. However, it tends to focus excessively on local relationships within sequences. As a result, its performance is comparable to STAMP but slightly inferior to SASRec. SASRec leverages a self-attention mechanism to focus on critical items in user behavior sequences. It excels in contexts where sequences are long or behavior gaps are irregular. As a result, it achieves the best performance among baseline sequential models, particularly in short-term recommendation scenarios. However, as K increases, our proposed model gradually surpasses all baseline models, especially at K = 10 and K = 20, demonstrating superior accuracy and relevance in larger candidate pools. This improvement can be attributed to the additional consideration of the POI relationship graph and spatiotemporal factors, in addition to the multi-head SASRec model, which enables our model to capture diverse user preferences more effectively. As a result, it slightly exceeds other baseline models in handling larger recommendation sets, showing its exceptional capability in generating accurate recommendations as K increases.

TABLE 2. PERFORMANCE OF RECOMMENDATION MODELS AT K=10

Type	Method	ACC	HR	MRR	NDCG
Traditional non-sequential methods	PopRec	0.0012	0.0122	0.0046	0.0061
	ItemKNN	0.0176	0.1755	0.0349	0.0663
	BPR	0.0220	0.2195	0.1359	0.1556
Sequential model	FPMC	0.0520	0.5199	0.3125	0.3634
	STAMP	0.0532	0.5317	0.3166	0.3706
	GRU4Rec	0.0525	0.5250	0.3134	0.3653
	SASRec	0.0532	0.5315	0.3177	0.3701
	Proposed model	0.0537	0.5373	0.3199	0.3736

TABLE 3. PERFORMANCE OF RECOMMENDATION MODELS AT K=20

Type	Method	ACC	HR	MRR	NDCG
Traditional non-sequential methods	PopRec	0.0009	0.0181	0.0049	0.0075
	ItemKNN	0.0115	0.2291	0.0396	0.0810
	BPR	0.0136	0.2709	0.1395	0.1686
Sequential model	FPMC	0.0286	0.5725	0.3160	0.3766
	STAMP	0.0289	0.5783	0.3196	0.3821
	GRU4Rec	0.0288	0.5764	0.3170	0.3783
	SASRec	0.0284	0.5681	0.3211	0.3827
	Proposed model	0.0289	0.5789	0.3225	0.3838

D. Effects of different dataset sizes

To understand how dataset sizes might impact the proposed system, we tested three additional datasets of varying sizes: Florida (medium size), Tennessee (small size with fewer POIs), and Indiana (small size). Table 4 presents the statistics of these datasets.

TABLE 4. THREE DATASET STATISTICS

	Florida	Tennessee	Indiana
Users	3641	2009	1762
POIs	463	217	301
Check-in records	122692	90847	83715

Based on Figures 2 to 3, we can observe that, despite differences in dataset sizes and POI distributions, the performance trends of traditional non-sequential methods and sequential recommendation models remain consistent across the four datasets: Pennsylvania (largest), Florida (medium size), Tennessee (small size with fewer POIs), and Indiana (small size). Sequential models demonstrate significant advantages in all datasets and evaluation metrics, highlighting their effectiveness in capturing temporal and sequential patterns. Moreover, our model surpasses SASRec across all datasets and evaluation metrics, particularly at K=10 and K=20. This demonstrates that the proposed model significantly improves recommendation accuracy in larger candidate pools.

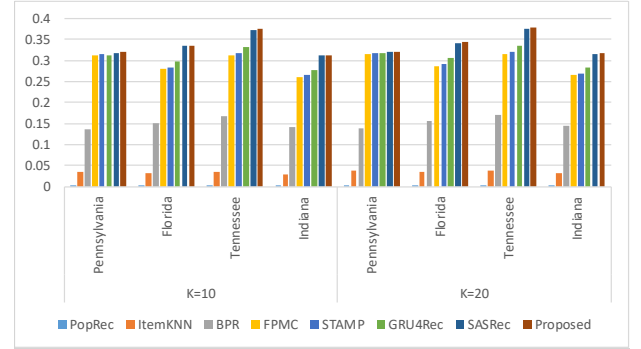


Figure 2. Model performance comparison using MRR@K

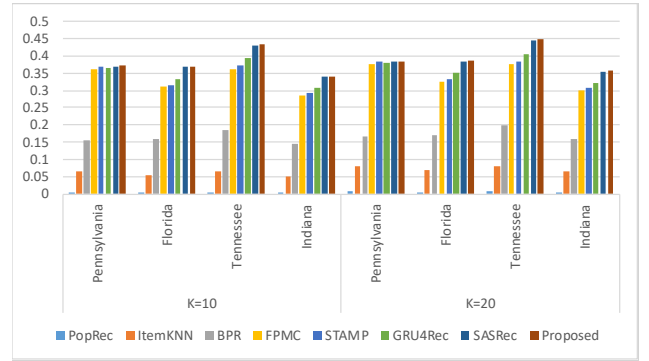


Figure 3. Model performance comparison using NDCG@K

E. Benefits of the grid-based spatial encoding

Next, we compare the performance of grid-based encoding and GPS coordinate encoding to observe their impact on the results. This study uses Min-Max Scaling as the GPS coordinate encoding (Han et al., 2022). This technique normalizes latitude and longitude values into a fixed range, ensuring the input features remain within a controlled numerical scale. Table 5 presents the experimental results for ACC@K, HR@K, MRR@K, and NDCG@K with K=10 and 20. These results show that using grid-based encoding consistently improves recommendation performance across all evaluation metrics compared to GPS coordinate encoding. The grid-based approach provides a more effective spatial feature representation, leading to more accurate and relevant recommendations.

TABLE 5. IMPACT OF GRID-BASED AND GPS COORDINATE ENCODINGS

		Grid-based encoding	GPS coordinate encoding
K=10	ACC	0.0537	0.0532
	HR	0.5373	0.5316
	MRR	0.3199	0.3113
	NDCG	0.3736	0.3656
K=20	ACC	0.0289	0.0287
	HR	0.5789	0.5743
	MRR	0.3225	0.3141
	NDCG	0.3838	0.3763

F. Impact of the number of attention heads

This section examines how varying the number of attention heads affects the performance of the recommendation model. Table 6 presents the results for ACC@K, HR@K, MRR@K, and NDCG@K when the number of attention heads is set to 2, 4, and 8. From the results, we observe that increasing the attention heads from 2 to 4 generally improves performance across all metrics. These results suggest that increasing the number of attention heads enables the model to capture more complex interactions, resulting in enhanced recommendation accuracy. However, increasing the attention heads from 4 to 8 does not provide further improvements.

In some cases, performance slightly declines. This suggests that while adding more attention heads can enhance feature extraction, too many heads may introduce redundancy or excessive noise, leading to marginal or negative performance effects. Overall, the results suggest that using four attention heads offers the optimal balance between capturing meaningful relationships and minimizing unnecessary complexity. While increasing attention beyond this point does not significantly degrade performance, it also does not provide substantial benefits.

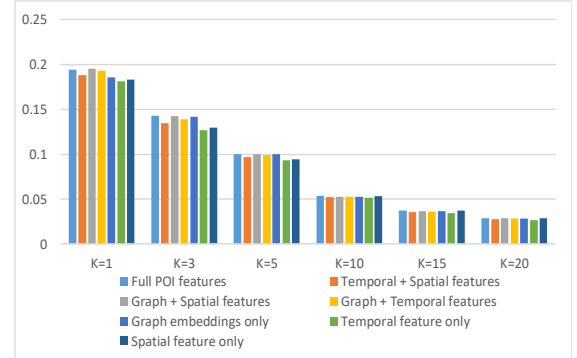
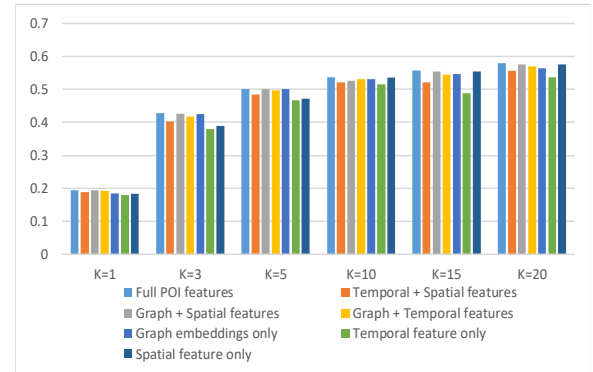
TABLE 6. IMPACT OF THE NUMBER OF ATTENTION HEADS ON RECOMMENDATION METRICS

		Attention heads = 2	Attention heads = 4	Attention heads = 8
K=10	ACC	0.0524	0.0537	0.0533
	HR	0.5240	0.5373	0.5328
	MRR	0.3107	0.3199	0.3157
	NDCG	0.3633	0.3736	0.3692
K=20	ACC	0.0289	0.0289	0.0285
	HR	0.5772	0.5789	0.5709
	MRR	0.3143	0.3225	0.3183
	NDCG	0.3767	0.3838	0.3788

G. Ablation study

This section examines the impact of various feature combinations on the performance of the proposed system. Figures 4 and 5 present the results for ACC@K and HR@K at K = 1, 3, 5, 10, 15, and 20 across various settings, including full POI features and multiple feature combinations. The results show that using full POI features achieves the best performance. For example, at K = 20, the HR reaches 0.5789, the highest value across all feature configurations. This suggests that integrating all available features yields the most comprehensive representation of POIs, resulting in improved recommendation accuracy.

The combination of graph and spatial features also shows strong performance among the partial feature settings. At K = 1, the ACC is 0.1950, the highest among all configurations. At K = 3, the HR reaches 0.4261, close to the full POI embedding performance. This suggests that graph-based embedding effectively captures the relational dependencies between POIs, and incorporating spatial information further enhances location-based recommendations. In contrast, using only temporal or spatial features results in noticeably lower performance. Overall, the findings suggest that while graph-based embedding is essential for improving recommendations, combining it with temporal and spatial features further enhances the model's ability to provide accurate predictions.

**Figure 4.** Impact of feature combination using ACC@K**Figure 5.** Impact of feature combination using HR@K

V. CONCLUSION AND FUTURE WORK

With the increasing demand for personalized travel experiences, next POI recommendation systems have become essential for guiding users toward relevant attractions. However, accurately capturing and utilizing the complex relationships among spatial, temporal, and behavioral features remains a significant challenge. To address these challenges, this study proposed a next POI recommendation system that integrates graph embedding, temporal information, grid-based spatial encoding, and transformer-based architectures to enhance recommendation accuracy. The key contributions of this research are as follows. First, graph embedding techniques to capture complex structural relationships between POIs lead to more accurate POI recommendations. Second, integrating different time levels of temporal features enhances the model's ability to reflect evolving user preferences over various periods.

Third, adopting a grid-based spatial encoding improves the representation of POI distributions, capturing spatial dependencies more precisely. Ultimately, implementing a transformer-based model effectively captures the complex interactions between features, surpassing conventional deep learning models in terms of recommendation accuracy.

Future research can extend this study in several directions. First, while the proposed system was evaluated using the Yelp dataset, testing on additional datasets would help assess its generalizability across diverse real-world scenarios. Second, incorporating richer contextual information—such as textual descriptions or user reviews—may enhance the semantic quality of POI embeddings and improve the accuracy of recommendations. Third, the current dataset covers a limited geographic area; applying the model to datasets with broader spatial coverage could evaluate its scalability. Finally, integrating special holiday features into temporal embeddings could better capture behavioral shifts during such periods, further refining prediction performance.

REFERENCES

- [1] W. C. Kang and J. McAuley, "Self-attentive sequential recommendation," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2018, pp. 197–206.
- [2] K. Sun et al., "Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation," in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 1, pp. 214–221, 2020.
- [3] Y. Wu, K. Li, G. Zhao, and X. Qian, "Personalized long- and short-term preference learning for next POI recommendation," IEEE Trans. Knowl. Data Eng., vol. 34, no. 4, pp. 1944–1957, 2022.
- [4] Y. Luo, Q. Liu, and Z. Liu, "Stan: Spatio-temporal attention network for next location recommendation," in Proc. Web Conf. (WWW), 2021, pp. 2177–2185.
- [5] C. Cheng, H. Yang, I. King, and M. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in Proc. AAAI Conf. Artif. Intell., vol. 26, no. 1, pp. 17–23, 2012.
- [6] H. Gao, J. Tang, and H. Liu, "gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks," in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2012, pp. 1582–1586.
- [7] X. He et al., "Neural collaborative filtering," in Proc. 26th Int. Conf. World Wide Web (WWW), 2017, pp. 173–182.
- [8] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," ACM Comput. Surv., vol. 52, no. 1, pp. 1–38, 2019.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
- [10] G. Lai, W. C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in Proc. 41st Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2018, pp. 95–104.
- [11] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in Proc. 36th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2013, pp. 363–372.
- [12] G. H. Nguyen et al., "Continuous-time dynamic network embeddings," in Proc. Web Conf. Companion, 2018, pp. 969–976.
- [13] D. Yang, J. Zhang, S. Wang, and X. Zhang, "A time-aware CNN-based personalized recommender system," Complexity, vol. 2019, Art. no. 9476981, 2019.
- [14] D. Wang, D. Xu, D. Yu, and G. Xu, "Time-aware sequence model for next-item recommendation," Appl. Intell., vol. 51, pp. 906–920, 2021.
- [15] Y. Guo, Y. Ling, and H. Chen, "A time-aware graph neural network for session-based recommendation," IEEE Access, vol. 8, pp. 167371–167382, 2020.
- [16] M. Rostami, M. Oussalah, and V. Farrahi, "A novel time-aware food recommender system based on deep learning and graph clustering," IEEE Access, vol. 10, pp. 52508–52524, 2022.
- [17] L. Huang et al., "Position-enhanced and time-aware graph convolutional network for sequential recommendations," ACM Trans. Inf. Syst., vol. 41, no. 1, pp. 1–32, 2023.
- [18] M. Ye, P. Yin, and W. C. Lee, "Location recommendation for location-based social networks," in Proc. 18th SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst., 2010, pp. 458–461.
- [19] L. Dos Santos, B. Piwowarski, and P. Gallinari, "Gaussian embeddings for collaborative filtering," in Proc. 40th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2017, pp. 1065–1068.
- [20] C. Ma, Y. Zhang, Q. Wang, and X. Liu, "Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence," in Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2018, pp. 697–706.
- [21] H. Wang, H. Shen, W. Ouyang, and X. Cheng, "Exploiting POI-specific geographical influence for point-of-interest recommendation," in Proc. Int. Joint Conf. Artif. Intell. (IJCAI), 2018, pp. 3877–3883.
- [22] Y. Yin et al., "GPS2Vec: Towards generating worldwide GPS embeddings," in Proc. 27th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst., 2019, pp. 416–419.
- [23] Y. Yin et al., "GPS2Vec: Pre-trained semantic embeddings for worldwide GPS coordinates," IEEE Trans. Multimedia, vol. 24, pp. 890–903, 2021.
- [24] M. Li, W. Zheng, Y. Xiao, K. Zhu, and W. Huang, "Exploring temporal and spatial features for next POI recommendation in LBSNs," IEEE Access, vol. 9, pp. 35997–36007, 2021.
- [25] A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017.
- [26] J. Xiao et al., "Attentional factorization machines: Learning the weight of feature interactions via attention networks," arXiv preprint arXiv:1708.04617, 2017.
- [27] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "STAMP: Short-term attention/memory priority model for session-based recommendation," in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2018, pp. 1831–1839.
- [28] J. Zou et al., "Improving conversational recommender systems via transformer-based sequential modelling," in Proc. 45th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2022, pp. 2319–2324.
- [29] S. Yang, J. Liu, and K. Zhao, "GETNext: Trajectory flow map enhanced transformer for next POI recommendation," in Proc. 45th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2022, pp. 1144–1153.
- [30] Y. Qin, Y. Fang, H. Luo, F. Zhao, and C. Wang, "Next point-of-interest recommendation with auto-correlation enhanced multi-modal transformer network," in Proc. 45th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2022, pp. 2612–2616.
- [31] Y. He, W. Zhou, F. Luo, M. Gao, and J. Wen, "Feature-based POI grouping with transformer for next point of interest recommendation," Appl. Soft Comput., vol. 147, Art. no. 110754, 2023.
- [32] K. Yang et al., "Self-explainable next POI recommendation," in Proc. 47th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2024, pp. 2619–2623.
- [33] C.-Y. Tsai and J.-H. Wang, "A personalized itinerary recommender system: considering sequential pattern mining," Electronics, vol. 14, no. 10, p. 2077, 2025.
- [34] C.-W. Chang, C.-Y. Tsai, L. Yao, R. J. Kuo, and C.-Y. Tsai, "Personalized travel itinerary recommendation enhancing by user interests and point-of-interest characteristics," Inf. Technol. Tour., Apr. 2025.
- [35] C.-Y. Tsai, K.-W. Chuang, H.-Y. Jen, and H. Huang, "A tour recommendation system considering implicit and dynamic information," Appl. Sci., vol. 14, no. 20, p. 9271, 2024.
- [36] C.-Y. Tsai, Y.-J. Chen, A. S. Peña, and G. Paniagua, "A visiting sequence recommendation framework: Enhanced by dynamic landmark and stay time," Expert Syst. Appl., vol. 230, p. 120662, 2023.
- [37] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2014, pp. 701–710.
- [38] "Universal Transverse Mercator coordinate system," Wikipedia, [Online]. Available: <https://commons.wikimedia.org/wiki/File:Utm-zones.jpg>. [Accessed: Jul. 24, 2024].
- [39] C. Lloyd, Spatial Data Analysis: An Introduction for GIS Users. New York, NY, USA: Oxford Univ. Press, 2010.