ROTAN: A Rotation-based Temporal Attention Network for Time-Specific Next POI Recommendation

Shanshan Feng*
Centre for Frontier AI Research, A*STAR
Institute of High Performance Computing, A*STAR
Singapore, Singapore
victor_fengss@foxmail.com

Feiyu Meng*
University of Electronic Science and Technology of China
Chengdu, China
mengfeiyu52633@gmail.com

Lisi Chen
University of Electronic Science and Technology of China
Chengdu, China
lchen012@e.ntu.edu.sg

Shuo Shang†
University of Electronic Science and Technology of China
Chengdu, China
jedi.shang@gmail.com

Yew Soon Ong
Centre for Frontier AI Research, A*STAR
Nanyang Technological University
Singapore, Singapore
asysong@ntu.edu.sg

ABSTRACT

The next Point-of-interest recommendation has attracted extensive research interest recently, which predicts users' subsequent movements. The main challenge is how to effectively capture users' personalized sequential transitions in check-in trajectory, and various methods have been developed. However, most existing studies ignore the temporal information when conducting the next POI recommendation. To fill this gap, we investigate a time-specific next POI recommendation task, which additionally incorporates the target time information. We propose a brand new Time2Rotation technique to capture the temporal information. Different from conventional methods, we represent timeslots as rotation vectors and then perform the rotation operations. Based on the Time2Rotation technique, we propose a novel rotation-based temporal attention network, namely ROTAN, for the time-specific next POI recommendation task. The ROTAN begins by building a collaborative POI transition graph, capturing the asymmetric temporal influence in sequential transitions. After that, it incorporates temporal information into the modeling of individual check-in trajectories, extracting separate representations for user preference and POI influence to reflect their distinct temporal patterns. Lastly, the target time is integrated to generate recommendations. Extensive experiments are conducted on three real-world datasets, which demonstrates the advantages of the proposed Time2Rotation technique and ROTAN recommendation model.

*Both authors contributed equally to this research.
†Corresponding author.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671809

CCS CONCEPTS

• **Information systems → Social recommendation**; • **Computing methodologies → Learning latent representations**.

KEYWORDS

Next POI Recommendation, Rotations, Temporal Information

ACM Reference Format:
Shanshan Feng, Feiyu Meng, Lisi Chen, Shuo Shang, and Yew Soon Ong. 2024. ROTAN: A Rotation-based Temporal Attention Network for Time-Specific Next POI Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671809

1 INTRODUCTION

In recent years, location-based social networks (LBSNs), such as Foursquare and Google Maps, allow users to share their geographic locations through check-ins at points of interest (POI) on social networks. POIs typically represent specific geographical locations that users may find valuable or interesting, *e.g.*, shopping malls and restaurants. The increasing availability of online check-in data enables the POI recommendation problems [24, 33] that aim to suggest potential locations for users based on their preferences and context information. Among the variants of POI recommendation tasks, the next POI recommendation problem [3, 20, 22, 56] has attracted extensive research attention recently. Different from traditional POI recommendations, the next POI recommendation predicts users' next visits by incorporating sequential transitions and spatial-temporal context information.

A large variety of methods [8, 13, 28, 30, 38, 43–45, 47, 50] have been proposed for the next POI recommendation task. Conventionally, two kinds of methods have been commonly utilized: the graph neural networks [17, 43, 44, 47, 51] for learning diverse relations, and the recurrent neural networks or transformer based methods [8, 21, 26, 50, 55, 56] for modeling the trajectories. However,

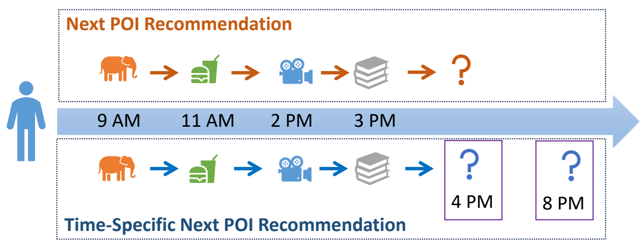


Figure 1: Conventional next POI recommendation task might overlook the importance of specific times in POI recommendations, whereas the time-specific next POI recommendation additionally considers the target time. For instance, the user's choice of next POIs at 4 PM and 8 PM could vary significantly.

most existing approaches focus solely on recommending subsequent locations without considering the target time information. For example, as shown in Figure 1, a user sequentially visits the zoo, snack bar, cinema, and library. Viewing the check-in trajectory as a sequence, the conventional next POI recommendation task primarily focuses on predicting the next location. However, recommendations at 4 PM and 8 PM should be different as user preference may vary with time. For example, users might prefer museums at 4 PM but favor bars at 8 PM. Additionally, POIs may have time-related characteristics, like opening hours. For instance, at 8 PM, recommending a museum that has already closed at 6 PM would be inappropriate. Hence, we introduce a novel research task, *i.e.,* time-specific next POI recommendation, which considers the target time information when generating the next POI recommendation.

The main distinction from the conventional next POI recommendation lies in the effective modeling of temporal information in user check-in trajectories. To investigate the temporal patterns of user sequential movements, we perform data observations and analyses on real-world check-in datasets (cf. Section 3). As shown in Figure 2, we observe that temporal information plays an important role in user consecutive check-ins, and asymmetric temporal influence is evident. In addition, as illustrated in Figure 3, users and POIs tend to exhibit different temporal patterns. These findings not only underscore the importance of temporal information but also present the following challenges of effectively integrating temporal information for time-specific next POI recommendations.

The first challenge is temporal information representation. Existing POI recommendation methods typically incorporate temporal information by representing time as an embedding vector. Then, the time embeddings are added or concatenated with POI and user embeddings [26, 44, 47, 50]. Those methods may increase the dimension and change their original spaces. In addition, this manner cannot naturally reflect the periodic temporal patterns that are essential in user behaviors. Hence, we need to design a novel approach to incorporate temporal information. The second challenge is time-aware sequential transitions. Existing methods often neglect temporal information when constructing transition graphs for the next POI recommendation. For instance, [50] simply constructs a user-agnostic POI-to-POI transfer graph, [30] considers relations between users and POIs without temporal aspects, and [44] develops an adaptive POI graph. However, careful consideration of temporal information is crucial, *e.g.,* how to capture the asymmetric temporal influence. The third challenge involves integrating various factors. Incorporating temporal information into various elements like user preference, POI sequential patterns, etc., proves to be a hard task. Data observations reveal that users and POIs display distinct temporal patterns. Current methods [20, 26, 50] often confound these factors, obscuring their contributions and hindering overall improvement.

To address these challenges, we propose a novel ROtation-based Temporal Attention Network (ROTAN). For integrating temporal information into embeddings, we introduce a new technique called Time2Rotation. Inspired by the success of Rotation methods in knowledge graph embeddings[1, 32, 36], Time2Rotation encodes the timeslot as a rotation. For a given POI or user, the temporal information of timeslot t is incorporated by applying the rotation operation. This approach offers two advantages: it naturally captures periodicities and does not alter the original embedding space. To account for asymmetric temporal relations in user trajectories, each timeslot can be assigned a source rotation and target rotation. This study is the first to employ rotation operations for incorporating temporal information in recommendation systems, distinguishing itself from conventional concatenation or addition operations.

Building on the Time2Rotation technique, we can generate time-specific POI recommendations by learning from collaborative POI transitions and individual check-in trajectories. Firstly, for collaborative POI sequential transitions, we create a time-aware POI transition graph where nodes represent POIs and relations convey temporal information. Each transition is modeled as a quadruplet comprising the source POI, source timeslot, target POI, and target timeslot. We utilize the Time2Rotation to model these sequential transitions. Secondly, we incorporate temporal information into learning user individual check-in trajectories. Recognizing that users and POIs exhibit distinct temporal patterns, we model their sequences separately. Source time rotation operations are applied to integrate temporal influence into the user and POI temporal sequence attention modules. This process extracts representations of user preference sequence and POI influence sequence for a given trajectory, capturing long-term sequential dependencies through attention mechanisms. Finally, Time2Rotation is utilized to integrate target time into these representations, which are combined to determine final scores for time-specific next POI recommendations.

Our contributions are summarized as follows:

- We investigate a time-specific next POI recommendation task that considers the target time when making subsequent POI recommendations, which is more challenging than the conventional next POI recommendation task.
- We introduce a Time2Rotation technique, encoding a given timeslot as rotations. This method naturally captures periodicity without altering the original space. To our knowledge, this is the first study to explore rotation operations to capture temporal information in user mobility behaviors.
- We propose a novel rotation-based temporal attention network. It utilizes Time2Rotation to consider temporal information in both global collaborative POI transitions and individual user check-in trajectories, and then incorporates the target time to generate recommendations.

- Extensive experiments on three real-world datasets demonstrate the superiority of the proposed ROTAN method. In particular, ROTAN outperforms existing baselines by achieving 22-26% higher Acc@1 and 12-22% higher MRR scores. Our implementation is available at https://github.com/ruiwenfan/ROTAN.

2 RELATED WORK

The POI recommendation task [24, 33] has been extensively studied, and various approaches have been proposed [15, 23, 25, 29, 39, 40, 42]. Several variants of the POI recommendation problem have emerged. In this section, we delve into two related variants, *i.e.,* the next POI recommendation and the time-aware POI recommendation. We also provide a literature review on rotation techniques.

2.1 Next POI Recommendation

Different from the conventional POI recommendation task, the key of next POI recommendation is effectively modeling the diverse interactions and sequential patterns in user check-in trajectories. Recently, a variety of methods have been developed. To capture the diverse relations in user check-in data, the *graph neural networks* [17, 30, 43, 44, 47, 51] are exploited. For instance, to harness collaborative signals from other users, [50] builds a POI-POI trajectory flow graph, and [44] develops an adaptive graph representation network. Yan *et al.* [47] present a hypergraph convolutional network that considers high-order transition information and incorporates trajectory similarities. Moreover, as the check-in trajectories can be regarded as sequences, the *recurrent neural networks* [21, 56] and *transformers* [8, 26, 50, 55] are leveraged to capture the complex dependencies in user's sequential movementes. For example, to model the check-in sequence, GetNext [50] develops a graph-enhanced Transformer model and AGRAN [44] designs a attention-based sequential model. CLSPRec [8] exploits a shared transformer to jointly learn long sequences and short sequences. However, existing studies focus on the conventional recommendation setting and ignore the target time of the recommendation, which may result in unreasonable recommendations and suboptimal performance.

Various factors have been considered for the POI recommendation, such as the categories [7], hierarchical structures [21], social relationships [17], etc. Among them, the geographical and temporal information are crucial factors that distinguish the next POI recommendations from other sequential recommendation tasks [9]. The geographical influence often be captured by spatial distances [44, 47] or region embeddings [20, 26], which are further incorporated into the RNN-based methods [22, 35, 56], transformer-based methods [27, 38], and graph-based methods [15, 23]. Similarly, the temporal information is commonly modeled as the temporal interval [38, 44, 47] or embedding of time [8, 26, 28, 50], and then incorporated into the trajectory prediction models.

In practice, almost all next POI recommendation methods have jointly considered spatial and temporal information when modeling the check-in sequences. For example, Zhao *et al.* [56] propose a Spatio-Temporal Gated Network (STGN) to capture personalized sequential patterns for users' long and short-term preferences. [27] develops a Spatio-Temporal Attention Network (STAN) that incorporates the impacts of non-adjacent locations and non-consecutive check-ins. In the recent methods, *e.g.,* [44, 47], the spatial distances and temporal interval information are aggregated into the attention mechanisms to capture spatial-temporal information. In another line of recent approaches, *e.g.,* [8, 26, 50], the spatial and temporal information are captured by the representation learning. This work also considers the spatial and temporal context for recommendations but focuses more on the temporal information. Different from existing methods that employ concatenation or addition operations, we leverage the rotation operations to incorporate temporal influence, which has not been studied before.

2.2 Time-aware POI Recommendation

Extensive studies have demonstrated the importance of temporal factors in user mobilities [24, 33]. The time-aware POI recommendation [14, 19, 53, 57] focuses on suggesting POIs for a user at a specified time of day. Yuan *et al.* [53] introduce this task and develop a collaborative filtering approach to integrate temporal and spatial information into the scoring calculation. Li *et al.* [19] employ a high-order tensor factorization method for time-aware POI recommendation. Overall, time-aware POI recommendation does not consider sequential movement information and has received much less research attention than the next POI recommendation.

The additional time information, *e.g.,* periodic patterns, has also been considered for the next POI recommendation [5, 10, 41, 52]. However, existing studies do not explicitly consider the target time when predicting the next locations. Very recently, Luo *et al.* [26] leverages timestamps as prompts in the location recommendation task, which is very close to our work. Specifically, it develops a temporal prompt-based and geography-aware (TPG) framework. The temporal information is encoded by 168 (24×7) embedding vectors, which will be used in the trajectory modeling and POI recommendation. The decoder of TPG takes a specific time as a temporal prompt to generate recommendations, by concatenating time embeddings with other embeddings. In contrast, we employ the rotation operation to integrate temporal influence, and extract different temporal patterns for user preference and POI influence.

2.3 Rotation Techniques

Sun *et al.* [36] introduce the rotation techniques for modeling multiple relations and develop a novel RotatE method for knowledge graph embedding, where each relation is described as a rotation from the source entity to the target entity in the complex vector space. Following this, other rotation-based knowledge graph embedding methods have been developed [1, 6, 32, 46]. Rotation techniques have found applications in various research domains. For example, [34] proposes a rotary positional encoding method for transformer-based language models, which is employed in the recent large language models (*e.g.,* LLaMA [37] and GLM-130B [54]). Feng *et al.* [11] combines the rotation operations with hyperbolic spaces to capture asymmetric proximity and hierarchical structures in directed graphs. Chen *et al.* [2] propose a learning-to-rotate attention module within the Quaternion transformer framework to capture the periodical patterns in time series forecasting tasks.

Recently, RotatE has been used in POI recommendation tasks. [51] applies the RotatE algorithm for learning the POI transition graph and multi-step dependency with RNN structures. [39] adopts the RotatE technique to learn entity representations from various types of graphs. These methods directly use RotatE to capture knowledge graphs related to POIs. Different from them, our approach explores rotation operations to model temporal information and considers asymmetrical sequential transitions.

3 PROBLEM STATEMENT AND ANALYSIS

3.1 Problem Definition

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ be the set of users and POIs, respectively. Each POI p is associated with a geographical coordinate (*i.e.,* longitude and latitude), which reflects its geospatial position. A check-in is represented by $c = (u, p, t)$, which denotes that user u visits POI p at timestamp t. In this work, we split the original timestamp into T timeslots in a day or week. Hence, t denotes a specific temporal timeslot.

A check-in trajectory is defined by a sequence of temporally ordered check-in records for a particular user. We use $T_{u_i} = \{c_1, c_2, \ldots, c_l\}$ to represent the trajectory sequence of user u_i. Following existing studies (*e.g.,* [50]), we partition all check-ins of a specified user into check-in trajectories by 24-hour intervals.

Definition 1: Next POI Recommendation (NPR). Given a user u_i and his recent check-in trajectory $T_{u_i} = \{c_1, c_2, \ldots, c_l\}$, it aims to predict the potential POIs that user u_i will most likely to visit subsequently.

Definition 2: Time-Specific Next POI Recommendation (TS-NPR). Given a user u_i, his recent check-in trajectory $T_{u_i} = \{c_1, c_2, \ldots, c_l\}$, and a specific target time t^{Tgt}, it aims to predict the POIs that user u_i will most likely to visit subsequently at the given timeslot t^{Tgt}.

Compared with the traditional NPR task, the TS-NPR additionally incorporates the target time t^{Tgt} when recommending the next POI, which is more challenging.

3.2 Data Observations and Analyses

To better solve the TS-NPR problem, we conduct data observations based on real-world datasets. Firstly, we examine the temporal characteristics of POI-POI sequential transitions. Using check-in trajectory data, we extract the temporal information associated with each POI-POI transition. The hour distribution of the NYC dataset is presented in Figure 2. Similar temporal patterns are observed in other datasets (Reported in the Appendix.). We have two findings: (1) Close hours exhibit relatively higher transition frequency (*e.g.,* the elements near the diagonal in the transition matrix.), underscoring the significance of temporal influence in user sequential movements. This finding is consistent with previous studies [12]. (2) Asymmetric temporal influence is evident in consecutive check-ins. For instance, there are more sequential transitions from 6 AM to 8 AM than the counts from 8 AM to 6 AM. These asymmetric patterns have not been extensively explored in previous studies.

Secondly, we investigate the temporal patterns of both users and POIs, as depicted in Figure 3. The X-axis represents the number of hours a user has check-ins (or a POI has been visited), while the Y-axis indicates the count of such users and POIs. We can learn that Users are active during more hours, as seen in their broader hour ranges. In contrast, the hour distribution of POIs is more concentrated, indicating that most POIs are accessible during limited hours. For instance, a museum may open from 9 AM to 6 PM, and restaurants may be visited during lunch or dinner times. The distinct temporal distributions of users and POIs necessitate separate modeling approaches.

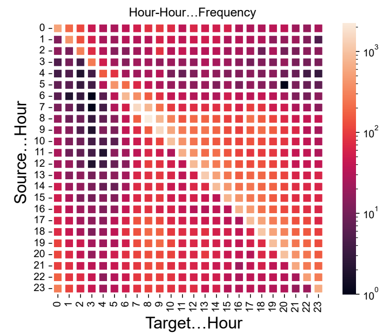
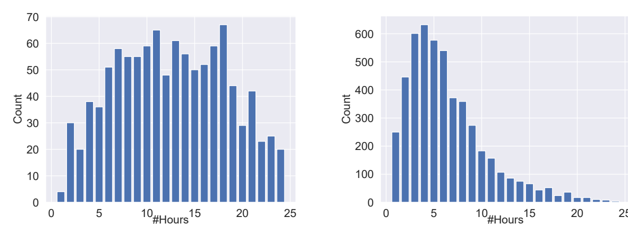


Figure 2: The temporal distribution of sequential transitions on the NYC dataset. The X-axis is the source hour and the Y-axis is the target hour. Each element denotes the frequency of such transition.

These observations demonstrate the inherent temporal patterns in user check-in data, which should be considered to address the TS-NPR problem.



(a) Users **(b) POIs**

Figure 3: The temporal distribution of users and POIs on the NYC dataset.

4 THE ROTAN METHOD

The overall framework of the ROTAN model is shown in Figure 4, which encompasses three major components. Firstly, to make full use of the POI sequential transitions of all users, we utilize a collaborative POI transition graph learning module to extract general temporal relations among POIs. Secondly, we develop a novel individual check-in trajectory learning component to capture both the long-term sequential dependencies and temporal information. This involves utilizing two distinct transformers for user temporal sequence and POI temporal sequence to reflect the different temporal patterns of users and POIs. Lastly, the learned trajectory representations and the input target time are combined to recommend the time-specific next POIs. Notably, the above three components exploit a Time2Rotation (T2R) technique to incorporate the temporal information. In the following sections, we will

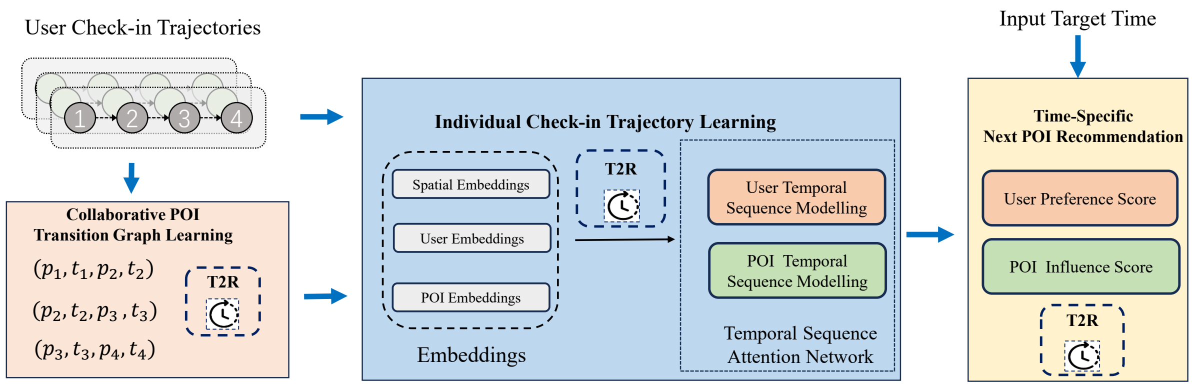


Figure 4: The overview of the rotation-based temporal attention network (ROTAN). It mainly consists of three components: the collaborative POI transition graph learning, the individual check-in trajectory modeling, and the time-specific next POI recommendation. All the above modules exploit the Time2Rotation (T2R) technique to incorporate temporal information.

first introduce the Time2Rotation approach and then present the three major components.

4.1 Time2Rotation

As an important factor in POI recommendation, temporal information has been widely investigated. Previous solutions either use the time interval (*e.g.,* the time difference between two consecutive check-ins) [38, 44] or timeslot's embedding [26, 50] (*e.g.,* regarding timeslots as entity nodes and learning their representations) to reflect the relative or absolute temporal influence in user check-in trajectories. Different from them, we explore the rotation technique for incorporating temporal information, which has not been studied before. This concept is inspired by circular patterns inherent in temporal information, such as daily and weekly periodicities. Human behaviors are closely tied to these periodic patterns, such as having breakfast in the morning and having dinner at night. This temporal information and periodicity intuitively correlate with rotation, similar to how time information can be symbolized by the rotation of clock hands.

To capture periodic patterns in user mobility, various cyclic functions, *e.g.,* sine function [18] and havercosine function [48], have been employed. However, the conventional methods of integrating temporal information, such as directly using it as temporal weights or concatenating timeslot embeddings, may not effectively model the temporal influence on other entities. For instance, concatenating time embeddings to users or POIs might increase the embedding dimension and alter their original vector spaces. Inspired by the RotatE approach [36], which has been proven effective for knowledge graph embedding, we propose to capture temporal influence by rotation operations.

Following the RotatE [36], rotations are modeled via unitary complex numbers (Alternative ways, *e.g.,* the block-diagonal transformation matrices [1], can also be used to model rotations.). As indicated by Euler's identity $e^{i\theta} = \cos\theta + i\sin\theta$, a unitary complex number can be naturally viewed as a rotation in the complex plane. The mathematical definition of Time2Rotation (T2R) is stated as follows. Given an embedding vector $\mathbf{h} \in \mathbb{C}^d$, the temporal information of timeslot $\mathbf{t}$ is incorporated by applying the rotation operation:

$$T2R(\mathbf{h}, \mathbf{r}_t) = \mathbf{h} \circ \mathbf{r}_t, \ where \mid \mathbf{r}_t \mid = 1. \tag{1}$$

Here, $\mathbf{r}_t \in \mathbb{C}^d$ denotes the rotation related to timeslot t, and "$\circ$" denotes the Hadamard (element-wise) product. Compared with the traditional strategies (*e.g.,* concatenation or addition), this rotation method does not modify the original space of embedding $\mathbf{h}$.

Additionally, the Time2Rotation technique further considers two important temporal factors. First, a given timeslot t would serve two roles: being influenced by activities in earlier timestamps or influencing future behaviors. Unlike existing solutions that assign only one embedding for each timeslot, we represent each timeslot with two rotations: one source rotation $\mathbf{r}_t^{Src}$ and one target rotation $\mathbf{r}_t^{Tgt}$ to capture asymmetric temporal relations. Second, in real-world scenarios, user mobility data exhibit multiple periodicities, *e.g.,* daily and weekly patterns. Following [50], we divide one day into 48 timeslots t_d (each representing 0.5 hours) to indicate daily periodicity. We also divide one week into 7 timeslots t_w (each representing 1 day) to represent weekly periodicity. These two periodicities can be combined to reflect the multi-periodic temporal patterns in user behaviors, as elaborated later.

4.2 Collaborative Transition Graph Learning

Since the check-in records of each user are sparse, we attempt to utilize the sequential transitions of all users. We first construct a collaborative POI-POI transition graph. Different from studies [44, 50, 51] that merely consider the sequential transition relations, we additionally incorporate the temporal information.

A user's current check-in was affected by the locations they have visited before and would influence the user's future movements. Hence, given a trajectory $T = \{c_1, c_2 ... c_l\}$, we can extract $l - 1$ transitions. Each transition is represented by a quadruplet:

$$quad_i = (p_i, t_i, p_{i+1}, t_{i+1}). \tag{2}$$

Each $quad_i$ means that a user first visits p_i at time t_i, and then moves to location p_{i+1} at time t_{i+1}. We define the temporal transition distance of p_i and p_{i+1} as

$$Dist_{quad_i} = \left\| T2R^{Src}(p_i, t_i) - T2R^{Tgt}(p_{i+1}, t_{i+1}) \right\|, \quad (3)$$

where $T2R^{Src}(p_i, t_i) = \mathbf{p}_i \circ \mathbf{r}_i^{Src}$ and $T2R^{Tgt}(p_{i+1}, t_{i+1}) = \mathbf{p}_{i+1} \circ \mathbf{r}_{i+1}^{Tgt}$ indicate the rotated POI embeddings, and $\|\cdot\|$ denotes the L1-norm.

Different from the RotatE [36] that considers the triplet of two entities and one relation, we leverage the T2R to address the time-aware quadruplet. By reducing the $Dist_{quad_i}$, we can learn temporal sequential patterns, where the timeslot's different roles are encoded by source rotation and target rotation, respectively.

Following [36], we can exploit the widely used negative sampling strategy to optimize the POI transition graph. This enables the extraction of POI embeddings and temporal rotations, forming the foundational elements for long-term trajectory modeling, as elaborated in the subsequent section.

4.3 Individual Check-In Trajectory Learning

Next, we will present how to learn the check-in trajectory by incorporating temporal information. The main challenge lies in jointly modeling various crucial factors, including user preference, influence, geographical influence, and temporal information.

Given a check-in trajectory sequence $T_{u_i} = \{c_1, c_2, \ldots, c_l\}$, we attempt to extract the trajectory representation. Alongside the mentioned POI embedding and time rotations, we also integrate User embedding and spatial embedding. Each user is assigned an embedding vector, which reflects the user's personalized interest. For the geographical influence, we use GeoEncoder [20] to get the spatial embeddings. Given the POI's coordinates (*i.e.,* longitude and latitude), a hierarchical map gridding is exploited to generate its spatial embedding. These embeddings are collectively used to encode the check-ins in trajectories.

There are two important elements: user preference and POI influence, both containing sequence information. Notably, users and POIs exhibit distinct temporal patterns, as illustrated in Figure 3. Therefore, choosing independent time encoding models for users and POIs becomes necessary. Motivated by this, we employ two transformers for user temporal sequence modeling and POI temporal sequence modeling, differing from prior studies that typically use the same sequence model for both.

4.3.1 User Temporal Sequence Modeling. Given each check-in $c = (u, p, t)$ in the observed trajectory, we employ source temporal rotations to capture its influence on subsequent check-ins. Acknowledging the existence of multiple periodicities in user behaviors, we consider daily and weekly patterns using two distinct types of rotations. Then, the user's temporal representations are extracted:

$$\mathbf{z}_\mathbf{d}^\mathbf{u} = T2R^{Src}(\mathbf{u}', \mathbf{t}_\mathbf{d}^\mathbf{u}), \quad \mathbf{z}_\mathbf{w}^\mathbf{u} = T2R^{Src}(\mathbf{u}', \mathbf{t}_\mathbf{w}^\mathbf{u}). \quad (4)$$

Here, $\mathbf{u}' = \mathbf{u} \parallel \mathbf{p}$ represents the user's preference for POIs, and $\parallel$ denotes the concatenation operation. $\mathbf{t}_\mathbf{d}^\mathbf{u}$ and $\mathbf{t}_\mathbf{w}^\mathbf{u}$ denote the daily and weekly periodic temporal rotations for user preferences. After that, these two representations can be fused to obtain $\mathbf{z}^\mathbf{u}$:

$$\mathbf{z}^\mathbf{u} = \alpha * \mathbf{z}_\mathbf{d}^\mathbf{u} + (1 - \alpha) * \mathbf{z}_\mathbf{w}^\mathbf{u}, \quad (5)$$

where $\alpha \in [0, 1]$ represents the weight for daily periodicity. In this manner, the user embedding $\mathbf{z}^\mathbf{u}$ is enriched with multi-periodic temporal information.

To model the long-term dependency in check-in trajectory, we use the attention mechanism to calculate the trajectory embeddings. Given the input z^u, we compute the linearly transformed features:

$$Q = z^u W_Q, \ K = z^u W_K, \ V = z^u W_V. \quad (6)$$

Here, W_Q, W_K and W_V are learnable parameters. We process the above features through a multi-head self-attention model:

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, \ldots, head_h) W^O, \\ head_i &= Attention(QW_i^Q, QW_i^K, VW_i^V), \\ Attention(Q, K, V) &= softmax\left(\frac{QK^T}{\sqrt{d}}\right)V. \end{aligned} \quad (7)$$

The outputs from different attention heads are combined using a linear transformation W^O to merge the information from various attention spaces.

Afterward, a fully connected neural network (FNN) is applied following the multi-head attention model. This process results in the user temporal trajectory representation:

$$\mathbf{g}^\mathbf{u} = FNN\big(MultiHead(Q, K, V)\big), \quad (8)$$

where $FNN()$ represents the commonly used feed-forward network. The $\mathbf{g}^\mathbf{u}$ preserves the historical user preference information.

4.3.2 POI Temporal Sequence Modeling. The POI sequence reflects the inherent sequential patterns of POIs, providing valuable information for predicting the subsequent locations. Similar to user preference, we incorporate temporal information into the POI sequence.

Firstly, the POI embeddings are rotated by daily and weekly rotations, defined as follows:

$$\mathbf{z}_\mathbf{d}^\mathbf{p} = T2R^{Src}(\mathbf{p}', \mathbf{t}_\mathbf{d}^\mathbf{p}), \quad \mathbf{z}_\mathbf{w}^\mathbf{p} = T2R^{Src}(\mathbf{p}', \mathbf{t}_\mathbf{w}^\mathbf{p}). \quad (9)$$

Here, $\mathbf{p}' = \mathbf{p} \parallel \mathbf{s}$, and $\mathbf{s}$ denotes the spatial embedding of the POI.

Then, these two rotated POI embeddings are fused to obtain multi-periodic temporal information:

$$\mathbf{z}^\mathbf{p} = \alpha * \mathbf{z}_\mathbf{d}^\mathbf{p} + (1 - \alpha) * \mathbf{z}_\mathbf{w}^\mathbf{p}. \quad (10)$$

Lastly, similar to the user temporal sequence modeling, the POI sequence representation $\mathbf{g}^\mathbf{p}$ is learned using a multi-head attention model followed by a fully connected neural network. The obtained representation $\mathbf{g}^\mathbf{p}$ captures the spatial and temporal information as well as complex sequential POI dependencies in the given trajectory.

4.4 Time-Specific Next-POI Recommendation

4.4.1 Incorporating the Target Time. In the TS-NPR problem, target time information is explicitly considered as the input. After the above check-in trajectory learning, we can obtain the two independent representation vectors: *i.e.,* user preference representation $\mathbf{g}^\mathbf{u}$ and POI influence representation $\mathbf{g}^\mathbf{p}$. Subsequently, Time2Rotation is employed to blend the target time into these vectors. Additionally, the multi-periodicity of temporal information is integrated.

The user preference at the target time is calculated as:

$$\begin{aligned} \mathbf{g}_\mathbf{d}^{\mathbf{u}'} = T2R^{Tgt}(\mathbf{g}^\mathbf{u}, \mathbf{t}_\mathbf{d}^\mathbf{u}), \ \mathbf{g}_\mathbf{w}^{\mathbf{u}'} &= T2R^{Tgt}(\mathbf{g}^\mathbf{u}, \mathbf{t}_\mathbf{w}^\mathbf{u}), \\ \mathbf{g}^{\mathbf{u}'} &= \alpha * \mathbf{g}_\mathbf{d}^{\mathbf{u}'} + (1 - \alpha) * \mathbf{g}_\mathbf{w}^{\mathbf{u}'}, \end{aligned} \quad (11)$$

where $\mathbf{t_d^u}$ and $\mathbf{t_w^u}$ denote the user's daily and weekly time rotations for the target time. Similarly, the POI preference at the target time is computed as :

$$\mathbf{g_d^{p'}} = T2R^{Tgt}(\mathbf{g^p}, \mathbf{t_d^p}),\ \mathbf{g_w^{p'}} = T2R^{Tgt}(\mathbf{g^p}, \mathbf{t_w^p}),$$
$$\mathbf{g^{p'}} = \alpha * \mathbf{g_d^{p'}} + (1-\alpha) * \mathbf{g_w^{p'}}, \quad (12)$$

where $\mathbf{t_d^p}$ and $\mathbf{t_w^p}$ represent the POI's daily and weekly time rotations for the target time.

The recent spatial information is also crucial in POI recommendation [12], which can provide geographical context information. Based on the latest checked-in POI in the given trajectory, we can get its spatial embedding $\mathbf{s}$ by the GeoEncoder [20]. Hence, we further incorporate the latest POI embedding $\mathbf{p_l}$ and spatial embedding $\mathbf{s_l}$ to output the final representations:

$$\mathbf{o^u} = \mathbf{g^{u'}} \parallel \mathbf{p_l},\ \ \mathbf{o^p} = \mathbf{g^{p'}} \parallel \mathbf{s_l}. \quad (13)$$

In this manner, we can get the time-specific user preference representation $\mathbf{o^u}$ and POI influence representation $\mathbf{o^p}$, which capture information about the current trajectory and the target time.

4.4.2 Predicting the Next POI. Based on the obtained two representations $\mathbf{o^u}$ and $\mathbf{o^p}$, we leverage the MLP as the decoder to compute POI recommendation scores.

First, we calculate the user preference score $\mathbf{y^u}$ and POI influence score $\mathbf{y^p}$ separately:

$$\mathbf{y^u} = Softmax(\mathbf{o^u} \cdot \mathbf{W_u} + \mathbf{b_u}), \quad (14)$$

$$\mathbf{y^p} = Softmax(\mathbf{o^p} \cdot \mathbf{W_p} + \mathbf{b_p}). \quad (15)$$

Here, $\mathbf{W_u} \in R^{d\times N}, \mathbf{W_p} \in R^{d\times N}, \mathbf{b_u} \in R^N$, and $\mathbf{b_p} \in R^N$ are learnable parameters. N denotes the number of POIs in the dataset.

After that, the final recommendation score $\mathbf{y}$ is calculated as

$$\mathbf{y} = \beta * \mathbf{y^u} + (1-\beta) * \mathbf{y^p}, \quad (16)$$

where $\beta \in [0, 1]$ represents the contribution of user preference and $(1-\beta)$ indicates the impact of POI influence. We utilize the commonly used cross-entropy function as the loss function and exploit the L2 regularization to avoid overfitting.

Table 1: The statistics of three real-world datasets.

Dataset	NYC	TKY	CA
Users	1,047	2,281	3,951
POIs	4,937	7,821	9,670
Check-ins	80,166	306,345	168,922

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 Datasets. We conduct extensive experiments on three real-world datasets: Foursquare-NYC [49], Foursquare-TKY[1] [49] and Gowalla-CA [4] [2]. We follow the pre-processing approach in [47]. The check-in records are chronologically sorted, with the first 80% assigned to the train set, the middle 10% to the validation set, and the last 10% to the test set. Dataset statistics are reported in Table 1.

[1]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[2]https://snap.stanford.edu/data/loc-gowalla.html

5.1.2 Baselines. We evaluate the following four types of methods: (1) Traditional sequential recommendation methods (**FPMC** [31] and **PRME** [12]) that utilize the conventional matrix factorization and metric embedding techniques; (2) RNN-based methods (**LSTM** [16], **STRNN** [22], **STGN** [56], and **PLSPL** [45]) that incorporate spatial and temporal information into the Recurrent neural networks; (3) Transformer-based methods (**STAN** [27], **GETNext** [50], **CLSPRec** [8], and **AGRAN** [44]) that leverage the attention mechanisms to model check-in sequences; (4) Time-aware next POI recommendation method (**TPG** [26]) that uses timestamps as prompts for geography-aware location recommendation.

GETNext, CLSPRec, and AGRAN are state-of-the-art methods for the NPR task, and we slightly adapt them to the TS-NPR problem. GETNext and AGRAN additionally utilize the GNN techniques for POI graphs. TPG is particularly relevant to ours, which considers the timestamp for the next location prediction.

5.1.3 Evaluation Metrics. Following [50], we utilize the Accuracy@k and Mean Reciprocal Rank (MRR), which are widely used in recommendation systems. Acc@k indicates whether the true POI appears in the top-k recommended POIs. MRR measures the index of the correctly recommended POI in the ordered result list.

More details of the datasets, baselines, evaluation metrics, and implementations can be found in the Appendix.

5.2 Main Results

Table 2 displays the experimental results[3] of time-specific next POI recommendation across three datasets. Based on the results, the following observations can be made: (1) The recent Transformer-based methods outperform early-stage methods (FPMC and PRME) and RNN-based methods (LSTM, STRNN, STGN, and PLSPL), suggesting that the transformer is suitable for modeling the user check-in trajectories. (2) The GNN-enhanced transformer-based methods (*e.g.*, GETNext and AGRAN) achieve better scores than the transformer-only methods (STAN and CLSPRec). This indicates that the extracted or adaptively learned POI graphs provide additional global sequential transition information, improving the recommendation performance. (3) Generally, the TPG model is able to obtain better results compared to the latest next POI recommendation approaches. This is because TPG utilizes timestamps as prompts to predict future movements, showing that target time information helps make more accurate recommendations. (4) The proposed ROTAN significantly outperforms all baselines across the three datasets. For instance, ROTAN can surpass the strong baselines by achieving 22-26% higher Acc@1 and 12-22% higher MRR scores. Notably, ROTAN significantly outperforms the TPG, which is the state-of-the-art time-specific next POI recommendation method. In summary, the empirical results demonstrate the effectiveness of the ROTAN method, which leverages Time2Rotation technique for both POI transition graph and check-in trajectories.

5.3 Ablation Studies

To investigate the effectiveness of the proposed modules in ROTAN, we conduct the following ablation studies. (1) w/o Graph: We only utilize the check-in Trajectory learning module, removing

[3]The experimental results of the first seven methods are taken from [50].

Table 2: Experimental results on three datasets. Best results are in boldface, and second best results are underlined.

Methods	NYC				TKY				CA			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
FPMC	0.1003	0.2126	0.2970	0.1701	0.0814	0.2045	0.2746	0.1344	0.0383	0.0702	0.1159	0.0911
PRME	0.1159	0.2236	0.3105	0.1712	0.1052	0.2278	0.2944	0.1786	0.0521	0.1034	0.1425	0.1002
LSTM	0.1305	0.2719	0.3283	0.1857	0.1335	0.2728	0.3277	0.1834	0.0665	0.1306	0.1784	0.1201
STRNN	0.1483	0.2923	0.3622	0.2198	0.1409	0.3022	0.3577	0.2212	0.0799	0.1423	0.1940	0.1429
STGN	0.1716	0.3381	0.4122	0.2598	0.1689	0.3391	0.3848	0.2422	0.0810	0.1842	0.2579	0.1675
PLSPL	0.1917	0.3678	0.4523	0.2806	0.1889	0.3523	0.4150	0.2542	0.1072	0.2278	0.2995	0.1847
STAN	0.2231	0.4582	0.5734	0.3253	<u>0.1963</u>	0.3798	0.4464	0.2852	0.1104	0.2348	0.3018	0.1869
GETNext	0.2406	0.4815	0.5811	0.3528	0.1829	<u>0.4045</u>	<u>0.4961</u>	<u>0.2853</u>	0.1526	0.3278	0.3946	0.2364
CLSPRec	0.1784	0.3830	0.4591	0.2691	0.1453	0.3394	0.4106	0.2340	0.0891	0.1815	0.2013	0.1302
AGRAN	0.2121	0.4519	0.5529	0.3179	0.1428	0.3737	0.4605	0.2471	0.1199	0.3148	<u>0.4017</u>	0.2140
TPG	<u>0.2555</u>	<u>0.5005</u>	<u>0.5932</u>	<u>0.3669</u>	0.1420	0.3631	0.4492	0.2436	<u>0.1749</u>	<u>0.3285</u>	0.3860	<u>0.2479</u>
ROTAN	**0.3106**	**0.5281**	**0.6131**	**0.4104**	**0.2458**	**0.4626**	**0.5392**	**0.3475**	**0.2199**	**0.3718**	**0.4334**	**0.2931**
Improvement	21.57%	5.51%	3.35%	11.86%	25.22%	14.36%	8.69%	21.80%	25.73%	13.18%	7.89%	18.23%

the collaborative POI transition graph. (2) w/o T2R: We remove the Time2Rotation technique and incorporate the temporal information through concatenation, which is the same as TPG [26]. (3) w/o Tgt: We do not consider the target time. This is the same setting as the conventional next POI recommendation task. (4) w/o User: We only keep the POI temporal sequence modeling module, and remove the representation of user preference. (5) w/o POI: We only keep the user temporal sequence modeling, and remove the representation of POI influence. (6) U&P: We do not separately consider the temporal influence on user preference and POI sequence representations. Instead, we merge them and learn trajectory simultaneously, aligning with the GETNext [50].

Table 3 presents the ablation study results on the NYC and CA dataset. We can draw the following observations. (1) Without the collaborative POI transition graph, the performance of ROTAN will decrease. This suggests that global sequential relations can alleviate the data sparsity, thereby improving the personalized check-in sequence prediction. (2) We observe that the results are much worse if Time2Rotation is removed, showing the superiority of Time2Rotation over the straightforward concatenation approach widely used in previous methods. (3) Removing target time information leads to a notable decrease in performance. It underscores the crucial role of target time in the next POI recommendation, justifying the significance of the TS-NPR task. (4) Performance notably declines when the user temporal sequence modeling module is removed, indicating the importance of user preference in POI recommendation tasks. (5) Removing the POI temporal sequence module results in a slight performance decline, indicating the lesser significance of the POI sequence compared to user preference. Hence, solely considering check-in trajectories as POI sequences may not be effective, highlighting the need to consider interactions between users and POIs. (6) Modeling temporal information for user preference and POI sequence with a single transformer is less effective than separately incorporating their temporal patterns. This is because users and POIs may exhibit distinct temporal patterns.

Table 3: The results of ablation studies.

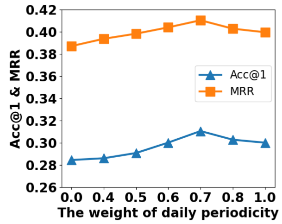
Methods	NYC		CA	
	Acc@1	MRR	Acc@1	MRR
Original	**0.3106**	**0.4104**	**0.2199**	**0.2931**
(1) w/o Graph	0.2909	0.3907	0.2084	0.2799
(2) w/o T2R	0.2562	0.3642	0.1854	0.2696
(3) w/o Tgt	0.2499	0.3634	0.1685	0.2535
(4) w/o User	0.2240	0.3151	0.1557	0.2231
(5) w/o POI	0.2885	0.3831	0.1953	0.2613
(6) U&P	0.2678	0.3782	0.1810	0.2740

5.4 Sensitivity Analyses

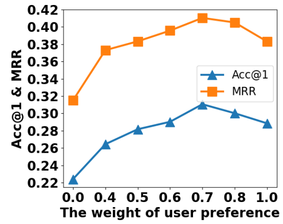
5.4.1 The weight α for periodicities . We explore the impact of the weight parameter α in combining daily and weekly periodicities (*i.e.,* α in Equations 5, 10, 11, and 12). By varying α, we observe its influence on performance, as shown in Figure 5a. The results indicate that combining both daily and weekly temporal patterns is effective. Using only daily periodicity (*i.e.,* $\alpha = 1.0$) or weekly periodicity alone (*i.e.,* $\alpha = 0.0$) is less effective. The best results are achieved with a weight of 0.7 for daily periodicity and 0.3 for weekly periodicity. This suggests that daily patterns play a more significant role than weekly patterns, aligning with our common knowledge that daily patterns are stronger than weekly patterns.

5.4.2 The weight β. User preference and POI influence are two crucial factors in our model, combined to calculate the final recommendation scores. Figure 5b shows the impact of different weights, *i.e.,* varying β values in Equation 16. Observations indicate that combining user preference and POI sequence is more effective than relying solely on user preference ($\beta = 1.0$) or POI sequence pattern ($\beta = 0.0$). Notably, the best results are obtained when the β is around 0.7. This suggests that user preference plays a more significant role, which is consistent with our findings in the ablation studies.

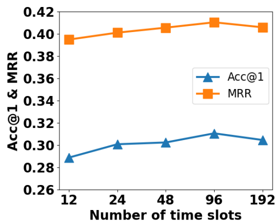
5.4.3 Granularity of time slots. We examine how the number of timeslots per day, which represents the time granularity, affects our model's performance, as depicted in Figure 5c. Although there's a



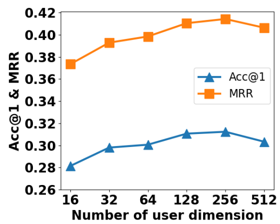
(a) The impact of α



(b) The impact of β



(c) Granularity of time slots



(d) Number of dimensions

Figure 5: The results of hyperparameter sensitivity analyses.

slight performance improvement with finer time granularity, overall our model's performance remains stable. In summary, ROTAN is not significantly affected by the number of time slots.

5.4.4 The number of dimensions. We study the impact of varying numbers of dimensions. Figure 5d indicates that performance tends to improve with the increase of dimensions. However, performance starts to drop if the number of dimensions exceeds 256. Empirically, the best performance is achieved when the number of dimensions is 256. It is important to note that, for a fair comparison with other baselines, the default value is set to 128 in this paper.

6 CONCLUSION

In this work, we study a new time-specific next POI recommendation problem that aims to provide accurate next recommendations at the given timestamp. We introduce an innovative Time2Rotation technique to effectively capture temporal information, which encodes timeslots as rotations and employs rotation operations to incorporate temporal influence. Building upon this technique, we propose ROTAN, a rotation-based temporal attention network for the time-specific next POI recommendation task. The ROTAN utilizes collaborative POI transition graph learning and individual check-in trajectory learning to learn representations of user temporal sequence and POI temporal sequence, which are combined with the input target information to produce time-specific recommendations. The extensive experiments and ablation studies validate the effectiveness of the proposed Time2Rotation technique and the ROTAN recommendation model.

ACKNOWLEDGMENT

This work was supported by the NSFC (No. U22B2037, U2001212, U21B2046, 62032001, and 61932004). This research was partly supported by the Distributed Smart Value Chain programme which is funded in part by the Singapore RIE2025 Manufacturing, Trade and Connectivity (MTC) Industry Alignment Fund-Pre-Positioning (Award No: M23L4a0001). This research was also partly supported by the Center for Frontier AI Research, Institute of High Performance Computing, A*STAR and the College of Computing and Data Science, Nanyang Technological University.

REFERENCES

[1] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *ACL*.

[2] Weiqi Chen, Wenwei Wang, Bingqing Peng, Qingsong Wen, Tian Zhou, and Liang Sun. 2022. Learning to rotate: Quaternion transformer for complicated periodical time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 146–156.

[3] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.

[4] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks *(KDD '11)*. Association for Computing Machinery, New York, NY, USA, 1082–1090. https://doi.org/10.1145/2020408.2020579

[5] Qiang Cui, Chenrui Zhang, Yafeng Zhang, Jinpeng Wang, and Mingchen Cai. 2021. ST-PIL: Spatial-Temporal Periodic Interest Learning for Next Point-of-Interest Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*. 2960–2964.

[6] Yao Dong, Lei Wang, Ji Xiang, Xiaobo Guo, and Yuqiang Xie. 2022. RotateCT: Knowledge Graph Embedding by Rotation and Coordinate Transformation in Complex Space. In *COLING*.

[7] Zheng Dong, Xiangwu Meng, and Yujie Zhang. 2021. Exploiting category-level multiple characteristics for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[8] Chenghua Duan, Wei Fan, Wei Zhou, Hu Liu, and Junhao Wen. 2023. CLSPRec: Contrastive Learning of Long and Short-term Preferences for Next POI Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 473–482.

[9] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.

[10] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. 1459–1468.

[11] Shanshan Feng, Lisi Chen, Kaiqi Zhao, Wei Wei, Xuemeng Song, Shuo Shang, Panos Kalnis, and Ling Shao. 2022. ROLE: Rotated Lorentzian Graph Embedding Model for Asymmetric Proximity. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[12] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 2069–2075.

[13] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. 2020. Hme: A hyperbolic metric embedding approach for next-poi recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on research and development in information retrieval*. 1429–1438.

[14] Lu Gao, Yuhua Li, Ruixuan Li, Zhenlong Zhu, Xiwu Gu, and Olivier Habimana. 2019. St-rnet: a time-aware point-of-interest recommendation method based on neural network. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[15] Haoyu Han, Mengdi Zhang, Min Hou, Fuzheng Zhang, Zhongyuan Wang, Enhong Chen, Hongwei Wang, Jianhui Ma, and Qi Liu. 2020. STGCN: A Spatial-Temporal Aware Graph Learning Method for POI Recommendation. In *2020 IEEE International Conference on Data Mining (ICDM)*. 1052–1057.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[17] Zheng Huang, Jing Ma, Yushun Dong, Natasha Zhang Foutz, and Jundong Li. 2022. Empowering next poi recommendation with multi-relational modeling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2034–2038.

[18] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. 2019. Time2Vec: Learning a Vector Representation of Time. arXiv:1907.05321 [cs.LG]

[19] Xin Li, Mingming Jiang, Huiting Hong, and Lejian Liao. 2017. A time-aware personalized point-of-interest recommendation via high-order tensor factorization. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 1–23.

[20] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-Aware Sequential Location Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 2009–2019.

[21] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Yong Liang Goh, Renrong Weng, and Rui Tan. 2022. Hierarchical multi-task graph recurrent network for next poi recommendation. In *Proceedings of the 45th international ACM SIGIR conference on Research and development in Information Retrieval*. 1133–1143.

[22] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[23] Shaohua Liu, Yu Qi, Gen Li, Mingjian Chen, Teng Zhang, Jia Cheng, and Jun Lei. 2023. STGIN: Spatial-Temporal Graph Interaction Network for Large-Scale POI Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4120–4124.

[24] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment* 10, 10 (2017), 1010–1021.

[25] Jing Long, Tong Chen, Quoc Viet Hung Nguyen, Guandong Xu, Kai Zheng, and Hongzhi Yin. 2023. Model-Agnostic Decentralized Collaborative Learning for On-Device POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 423–432.

[26] Yan Luo, Haoyi Duan, Ye Liu, and Fu lai Chung. 2023. Timestamps as Prompts for Geography-Aware Location Recommendation.

[27] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *Proceedings of the Web Conference 2021 (WWW '21)*. 2177–2185.

[28] Junjie Ou, Haiming Jin, Xiaocheng Wang, Hao Jiang, Xinbing Wang, and Chenghu Zhou. 2023. STA-TCN: Spatial-temporal Attention over Temporal Convolutional Network for Next Point-of-interest Recommendation. *ACM Transactions on Knowledge Discovery from Data* 17, 9 (2023), 1–19.

[29] Yifang Qin, Yifan Wang, Fang Sun, Wei Ju, Xuyang Hou, Zhe Wang, Jia Cheng, Jun Lei, and Ming Zhang. 2023. DisenPOI: Disentangling Sequential and Geographical Influence for Point-of-Interest Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 508–516.

[30] Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. 2022. Graph-Flashback Network for Next Location Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1463–1471.

[31] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[32] Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. In *AAAI*.

[33] Pablo Sánchez and Alejandro Bellogín. 2022. Point-of-interest recommender systems based on location-based social networks: a survey from an experimental perspective. *ACM Computing Surveys (CSUR)* 54, 11s (2022), 1–37.

[34] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063.

[35] K. Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation. In *AAAI Conference on Artificial Intelligence*.

[36] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

[37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[38] En Wang, Yiheng Jiang, Yuanbo Xu, Liang Wang, and Yongjian Yang. 2022. Spatial-Temporal Interval Aware Sequential POI Recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*.

[39] En Wang, Yuanbo Xu, Yongjian Yang, Yiheng Jiang, Fukang Yang, and Jie Wu. 2023. Zone-Enhanced Spatio-Temporal Representation Learning for Urban POI Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[40] Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, Jiyi Li, and Dongjin Yu. 2023. EEDN: Enhanced Encoder-Decoder Network with Local and Global Context Learning for POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 383–392.

[41] Xin Wang, Xiao Liu, Li Li, Xiao Chen, Jin Liu, and Hao Wu. 2021. Time-aware User Modeling with Check-in Time Prediction for Next POI Recommendation. In *2021 IEEE International Conference on Web Services (ICWS)*. 125–134.

[42] Xiaolin Wang, Guohao Sun, Xiu Fang, Jian Yang, and Shoujin Wang. 2022. Modeling Spatio-temporal Neighbourhood for Personalized Point-of-interest Recommendation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3530–3536.

[43] Zhaobo Wang, Yanmin Zhu, Haobing Liu, and Chunyang Wang. 2022. Learning Graph-Based Disentangled Representations for Next POI Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 1154–1163.

[44] Zhaobo Wang, Yanmin Zhu, Chunyang Wang, Wenze Ma, Bo Li, and Jiadi Yu. 2023. Adaptive Graph Representation Learning for Next POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 393–402.

[45] Yuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. 2022. Personalized Long- and Short-term Preference Learning for Next POI Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2022), 1944–1957. https://doi.org/10.1109/TKDE.2020.3002531

[46] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. TeRo: A Time-aware Knowledge Graph Embedding via Temporal Rotation. In *Proceedings of the 28th International Conference on Computational Linguistics*. 1583–1593.

[47] Xiaodong Yan, Tengwei Song, Yifeng Jiao, Jianshan He, Jiaotuan Wang, Ruopeng Li, and Wei Chu. 2023. Spatio-Temporal Hypergraph Learning for Next POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 403–412.

[48] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2020. Location Prediction over Sparse User Mobility Traces Using RNNs: Flashback in Hidden States!. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. 2184–2190.

[49] Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.

[50] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: Trajectory Flow Map Enhanced Transformer for Next POI Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 1144–1153.

[51] Feiyu Yin, Yong Liu, Zhiqi Shen, Lisi Chen, Shuo Shang, and Peng Han. 2023. Next POI recommendation with dynamic graph and explicit dependency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4827–4834.

[52] Haochao Ying, Jian Wu, Guandong Xu, Yanchi Liu, Tingting Liang, Xiao Zhang, and Hui Xiong. 2019. Time-aware metric embedding with asymmetric projection for successive POI recommendation. *World Wide Web* 22 (2019), 2209–2224.

[53] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 363–372.

[54] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: An Open Bilingual Pre-trained Model. arXiv:2210.02414 [cs.CL]

[55] Lu Zhang, Zhu Sun, Ziqing Wu, Jie Zhang, Yew Soon Ong, and Xinghua Qu. 2022. Next point-of-interest recommendation with inferring multi-step future preferences. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*. 3751–3757.

[56] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S. Sheng, and Xiaofang Zhou. 2022. Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2022), 2512–2524.

[57] Hai-Tao Zheng, Yingmin Zhou, Nan Liang, Xi Xiao, Arun Kumar Sangaiah, and Congzhi Zhao. 2017. Exploiting user mobility for time-aware POI recommendation in social networks. *IEEE Access* (2017).

A APPENDIX

A.1 Data Observations

We present the observation results on TKY and CA datasets in Figure 6, Figure 7, Figure 8, and Figure 9.

A.2 Experimental Setup

A.2.1 Dataset Descriptions and Statistics. Foursquare-NYC and Foursquare-TKY datasets contain check-ins in NYC and Tokyo (from 12 April 2012 to 16 February 2013). The Gowalla-CA dataset

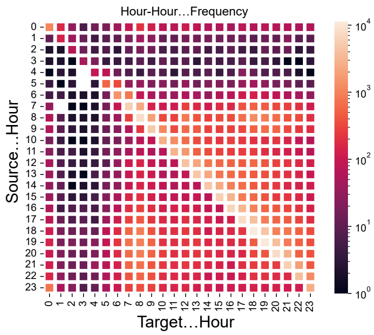
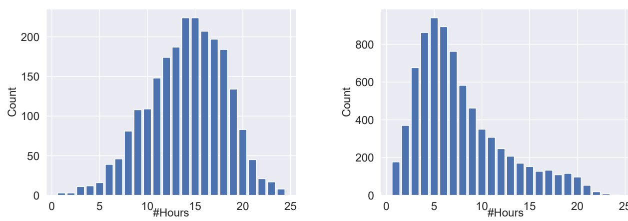


Figure 6: The temporal distribution of sequential transitions on the TKY dataset. The X-axis is the source hour and the Y-axis is the target hour. Each element denotes the frequency of such transition.



(a) Users **(b) POIs**

Figure 7: The temporal distribution of users and POIs on the TKY dataset.

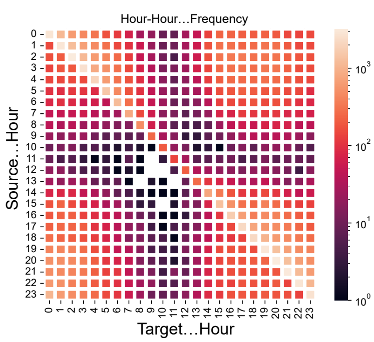
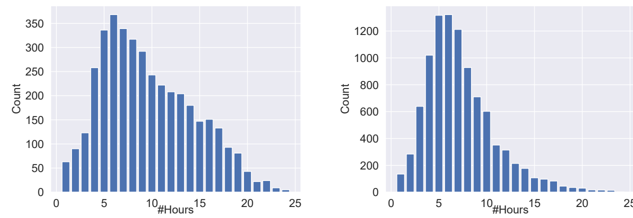


Figure 8: The temporal distribution of sequential transitions on the CA dataset. The X-axis is the source hour and the Y-axis is the target hour. Each element denotes the frequency of such transition.

is collected from the Gowalla platform, covering a broader geographical area across California and Nevada. Each check-in consists of user, POI, latitude, longitude, and timestamp. We follow the pre-processing setting in [47]: 1) filter out POIs with fewer than 10 visit records in history, 2) filter out users with fewer than 10



(a) Users **(b) POIs**

Figure 9: The temporal distribution of users and POIs on the CA dataset.

visit records in history, 3) divide user check-in records into several trajectories.

A.2.2 Baseline Descriptions. We evaluate the following methods:

- FPMC [31]: A personalized Markov chain framework, which learns both user preference and poi sequential pattern.
- PRME [12]: A metric embedding-based method, which jointly incorporates sequential transition, user preference, and geographical influence.
- LSTM [16]: A variant of the RNN model, which handles long sequence dependency and Vanishing gradient problems.
- STRNN [22]: A RNN-based method, considering the temporal and spatial interval information.
- STGN [56]: A RNN-based method, which adds two time gates and two distance gates to LSTM.
- PLSPL [45]: A RNN-based method, which learns the user's long-term pattern by using attention mechanism and short-term preference. and short-term preference with LSTM
- STAN [27]: A transformer-based method, which modifies the attention coefficients using temporal and spatial interval information.
- GETNext [50]: A transformer and GNN-based method, which employs a user-agnostic global trajectory flow map and a novel Graph Enhanced Transformer model.
- CLSPRec [8]: A transformer and RNN-based method, which uses the attention mechanism to obtain long-term preference and RNN to get short-term preference.
- AGRAN [44]: A transformer and GNN-based method, which uses an adaptive graph to avoid manually building a graph.
- TPG [26]: A transformer-based method, which explicitly uses target times as prompts for the geography-aware location recommendation. Its experimental setup is similar to ours.

A.2.3 Evaluation Metric Descriptions. In this work, we utilize two widely used measure metrics for evaluating the recommendation performance. Given a test set with m samples (trajectories), the metrics (Accuracy@k and MRR@k) are calculated as:

$$\begin{aligned} &Acc@k = \frac{1}{m}\sum_{1}^{m} hits(i), \quad MRR = \frac{1}{m}\sum_{1}^{m}\frac{1}{rank_i} \\ &hits(i) = \begin{cases} 1 & \text{if } rank_i <= k \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{17}$$

Here, $rank_i$ represents the rank of the ground truth next POI in the recommended list. In general, for all these metrics, a higher value indicates better performance.

A.2.4 Implementation Details. We use the PyTorch framework to develop and train our model. We conduct experiments on the following hardware (CPU: Intel Xeon Silver 4214R, GPU: NVIDIA GeForce RTX 3090 24GB). The hyperparameters for our experiments are as follows: the embedding dimensions for POI and user are set to 128, while the rotation dimensions for time are 64. We employ Adam as the model optimizer with a learning rate of 1e-3 and weight decay of 8e-4. The learning rate scheduler is set as CosineLRScheduler, with a minimum learning rate of 1e-5, warm-up steps at 10, and warm-up learning rate initialized at 1e-4. Additionally, the hyperparameters for the Transformer are configured as follows: transformer-head as 2, transformer-hidden as 1024, transformer-layer as 4, and dropout as 0.4. We also incorporate the GeoEncoder [20] in our embedding layer, setting the embedding dimensions for geography at 128, the quadkey length at 25, and the number of grams at 6. Training is performed on three datasets for 60 epochs with a batch size of 128.