Spatial-Temporal Interval Aware Individual Future Trajectory Prediction

Yiheng Jiang, Yongjian Yang, Yuanbo Xu†, En Wang

Abstract—The past flourishing years of sequential location-based services began with the introduction of the Self-Attention Network (SAN), which quickly superseded CNN or RNN as the state-of-the-art backbone. Recent works utilize modified attention mechanisms or neural network layers to process spatial-temporal factors to realize fine-grained individual behavior pattern modeling. However, we argue these methods can be further improved due to the significant increase in the model's parameter scale or computational burden. In this paper, we first exploit two lightweight approaches, Rotary Time Aware Position Encoder (RoTAPE) and multi-head Interval Aware Attention Block (IAAB), to impel SAN by efficiently and effectively capturing spatial-temporal intervals among the user's visited locations, which require neither extra parameters nor a high computational cost. On the one hand, RoTAPE encodes the day- and hour-level timestamps into sequence representation simultaneously via a sinusoidal encoding matrix, and the corresponding time intervals can be explicitly captured by SAN. Specifically, the multi-level temporal differences are mutually independent to reflect the periodical pattern and jointly complete to measure the absolute time interval. On the other hand, IAAB, point-wise injecting the historical spatial-temporal intervals into the attention map, can promote SAN attaching importance to the spatial relations under the constraints of time conditions. Then, we design a novel MLP-based module, Spatial-Temporal Relation Memory (STR Memory), implemented with fully connected linear layers and matrix transpose operations. STR Memory, endowing the interactions inside historical intervals along different directions, can convert the historical intervals into spatial-temporal relations in future trajectories for accurate predictions. To this end, we propose an end-to-end mobility trajectory prediction framework, namely STiSAN$^+$, employing RoTAPE, stacking multiple layers of IAAB-based encoder-decoder architecture, and coupling with STR Memory. We conducted numerous experiments on six public LBSN datasets to evaluate our proposed algorithm. From Next Location Recommendation to Multi-location Future Trajectory Prediction, our STiSAN$^+$ gains average 15.05% and 18.35% improvements against several state-of-the-art sequential models, respectively. Ablation studies demonstrate the effectiveness of RoTAPE, IAAB, and STR Memory under our framework. Moreover, we separately validate the extensibility and interpretability of RoTAPE and IAAB through non-sampled metric evaluation and visualization.

Index Terms—Spatial-temporal trajectory, self-attention network, sequential location-based recommendation.

1 Introduction

Credited to the rapid development of information technology, Location-Based Social Networks (LBSNs) have become the hot-spot for both industry and academia in recent years [1]. As two typical application scenarios in LBSN, Next Location Recommendation and Multi-location Future Trajectory Prediction aim to conjecture users' preferences or behavior patterns to provide pleasant suggestions or accurate predictions.

The ***spatial*** and ***temporal information*** are two pivotal and complementary factors in either location recommendation [2], [3] or trajectory prediction [4], [5]. Spatial information, e.g., the geography interval Δd in Fig. 1, can describe the physical proximity between locations [6], especially when individual mobility history [7] usually exhibits the spatial clustering phenomenon [8], [9], [10]. Temporal information, e.g., time interval Δt in Fig. 1, can reflect the relative temporal proximity among locations, which contributes to more personalized individual behavior modeling [11]. As the example in Fig. 1, user 1, 2 shared the same historical

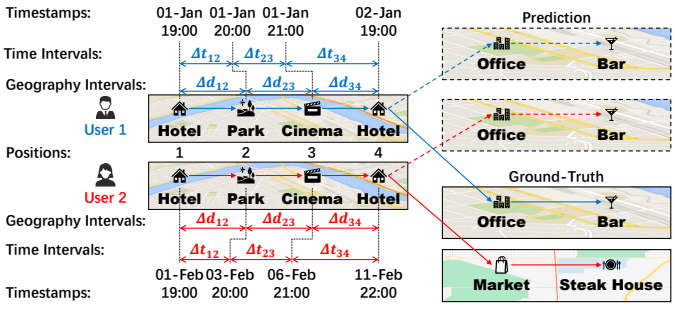


Fig. 1. The influence of spatial-temporal factors on behavior patterns modeling. The same historical trajectory with different time intervals results in different future trajectories of user 1 and user 2.

trajectory "$1: Hotel \rightarrow 2: Park \rightarrow 3: Cinema \rightarrow 4: Hotel$", and then they separately visited "$Office \rightarrow Bar$", "$Market \rightarrow SteakHouse$" in future. If a sequential model only considers the locations and spatial information, the representations for these two trajectories would be highly similar. The predictions might be biased from the ground truth (as the right half of Fig. 1). Fortunately, their time intervals Δt reveal the following distinctiveness of these two historical trajectories: *(1) Absolute Temporal Difference.* Along with the example in Fig. 1, user 1 visited "$Hotel$" and "$Park$" in 1 hour, while the situation of user 2 is around two days. The difference can measure the correlation among locations in such similar trajectories from the

• *Y. Jiang, Y. Yang, Y. Xu†(corresponding author) and E. Wang are with the Department of Computer Science and Technology, Jilin University, Changchun, 130012, China and Key Laboratory of Symbolic Computation and Knowledge Engineering for the Ministry of Education, Jilin University, Changchun, 130012, China. E-mail: jiangyh22@mails.jlu.edu.cn, yyj, yuanbox, wangen@jlu.edu.cn.*

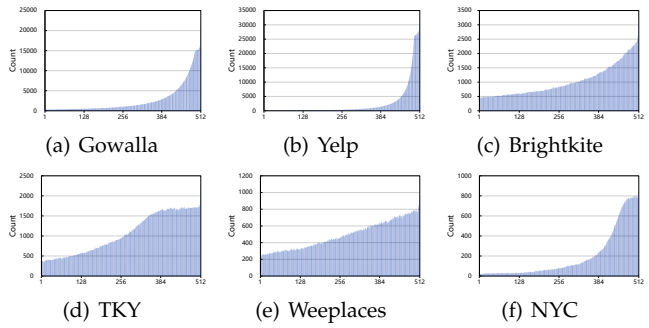


Fig. 2. The distribution of spatial correlated locations among users' historical trajectories in six datasets. The horizontal axis denotes positions, and the vertical axis counts the number of locations that are physically proximity to the corresponding target (less than 10 km).

temporal perspective. *(2) Relative Periodical Pattern.* Along the example, user 1 visited "$Hotel$" at 19:00 of 01-Jan and 02-Jan, where the day- and hour-level time intervals are 1 and 0, respectively. Such a phenomenon is common in the law of human mobility, which reflects the individual tidal preference. Thus, modeling temporal factors would be conducive to realizing fine-grained trajectory representations.

Various approaches have been attempted to explore the influence brought by temporal factors. For example, CFPRec [12] directly embeds the timestamps to incorporate temporal information, and CTLE [13] encodes them via a sinusoidal function with learnable parameters. In the separate line, [11], [14] integrate temporal information by carrying on modified attention mechanism. However, as the premise of all these advanced methods, mapping temporal information into high-dimension space significantly increases either parameter scale or computational cost.

Towards this issue, we propose the Rotary Time Aware Position Encoder (RoTAPE) in this paper, requiring neither extra parameters nor high computational burden, i.e., ***lightweight***, to comprehensively consider the temporal information in trajectories. It is sparked by the rotary positional encoding [15], which tackles the 1D positional information. Our RoTAPE extends it as the 2D version to encode the day simultaneously- and hour-level timestamps. Specifically, RoTAPE generates the time-aware positional matrix via the fixed sinusoidal function and injects it into trajectory representation in a multiplicative manner. Correspondingly, the attention mechanism can explicitly capture the multi-level temporal differences, which are jointly complete to measure the absolute time span and mutually independent to reflect the periodical pattern.

Taking a step further, the global weighted averaging in SAN inhibits the local relations [16], [17], i.e., SAN is an expert in learning the dependency from the whole trajectory while weakening the spatial correlation among neighboring locations. However, locations with strong spatial correlations are critical for prediction because the smaller geography distance usually leads to a higher visiting probability [18]. Fig. 2 visualizes the distribution of these critical locations, which strongly correlated with the target[1]. We can find that these locations distribute variously among historical trajectories (e.g., the last 128 visits in `Yelp`, the last 256 visits in `Gowalla`, `NYC`, and the whole trajectory in `Brightkite`, `TKY`, `Weeplaces`). Unfortunately, the inherent drawback limits SAN from considering such local relations meticulously, especially when processing longer trajectories.

To avoid this issue, we are inspired by [14], [17] and propose another lightweight module, the multi-head Interval Aware Attention Block (IAAB). Specifically, we first construct a spatial-temporal relation-based matrix on the geography and time intervals (Δd, Δt) among historical trajectories to reflect the spatial correlations under the constraints of temporal conditions. Then, IAAB introduces the relation matrix, as the inductive bias, into the attention map by point-wise addition. In this way, IAAB impels SAN, attaching importance to the spatial correlation in the whole trajectory, which relieves the insufficient local attention issue. Since IAAB utilizes the spatial-temporal information in an explicit manner, the interpretability is improved.

Moreover, since the prediction might be sub-optimal if we directly match them with candidate locations [6], we design a novel MLP-based Spatial-Temporal Relation Memory, namely STR Memory. It takes the historical spatial-temporal relation matrix as input and endows the interactions inside the relation matrix along different directions by linear layers and matrix transposition. With STR Memory, future spatial-temporal relations can be individually modeled without information leakage, which contributes to decoding the attentive results toward the future trajectory.

To this end, we integrate RoTAPE, IAAB, and STR Memory into an end-to-end mobility trajectory prediction framework called STiSAN$^+$. Our contributions can be summarized as follows:

- We exploit two lightweight approaches, Rotary Time Aware Position Encoder (RoTAPE) and multi-head Interval Aware Attention Block (IAAB), to consider the spatial-temporal information comprehensively, where requiring neither extra parameters nor a significant computational burden.
- We design a novel MLP-based module, Spatial-Temporal Relation Memory (STR Memory), for personalized converting historical relation matrix into future spatial-temporal relations, implemented solely with fully connected linear layers.
- We propose an end-to-end mobility trajectory prediction framework STiSAN$^+$ employing RoTAPE, stacking multiple IAAB-based encoder-decoder structures, and coupling with STR Memory.
- We conduct numerous experiments on six public LBSN datasets to evaluate the proposed algorithm under two scenarios. Ranging from Next Location Recommendation to Multi-location Future Trajectory Prediction, our STiSAN$^+$ gains average 15.05% and 18.35% improvements against several state-of-the-art sequential models. Besides demonstrating the effectiveness of our approaches through an ablation study, we also validate the efficiency, extensibility, and interpretability of RoTAPE and IAAB.

1. We define the maximum geography interval as 10km [18] and set the last visited location in trajectory as the target.

2 PRELIMINARIES

2.1 Basic Definitions

We utilize $\mathcal{U} = \{u_1, u_2, \cdots, u_{|U|}\}$ and $\mathcal{L} = \{l_1, l_2, \cdots, l_{|L|}\}$ to denote the user set and location set separately. Note that each location l_i in L correlates with a specific GPS coordinate $g_i = \langle lat_i, lon_i \rangle$ in the GPS set $\mathcal{G} = \{g_1, g_2, \cdots, g_{|L|}\}$. Accordingly, we have the following definitions to describe user-location interactions.

Definition 1. **Check-in.** A user u's single check-in behavior is denoted as a quadruple $c^u = \langle u, l, g, t \rangle$, which indicates that the user u visited location l at time t and the exact GPS coordinate is g. Note that the time t can be decomposed into the day-level t^d and hour-level t^h, which separately stand for the date index and hour index.

Definition 2. **Historical Trajectory.** A user u's historical trajectory $S^u = \left\{c_1^u \to c_2^u \to \cdots \to c_{|S^u|}^u\right\}$ records his or her $|S^u|$ check-ins by chronological order. Each check-in's position in the sequence is the subscript of the symbol.

Definition 3. **Historical Spatial-Temporal Relation Matrix.** For a user u's historical trajectory, the corresponding spatial-temporal relation matrix $\boldsymbol{R}_{his}^u \in \mathbb{R}^{|S^u| \times |S^u|}$ records the pair-wise spatial-temporal relations among check-ins. Specifically, the relation between a user u's i th and j th check-in r_{ij}^u is calculated by the time interval Δt_{ij}^u and geographical distance interval Δd_{ij}^u.

2.2 Problem Statement

Problem 1. **Future Trajectory Prediction** Given a user u's historical trajectory $S^u = \left\{c_1^u \to c_2^u \to \cdots \to c_{|S^u|}^u\right\}$, mining the dependency from visited locations, modeling the user's behavior patterns-based on the dependency and spatial-temporal information and predicting the user's future trajectory of length k. Specifically, it can be described as,

$$\mathcal{F}\left(S^u\right) \to \hat{S}^u, \tag{1}$$

where $\hat{S}^u = \left\{c_{|S^u|+k}^u\right\}_{k=1}^{K}$ is the predicted future trajectory and K is the number of locations contained in the predicted future trajectory. $\mathcal{F}(\cdot)$ is an abstract function representation symbol to signify trajectory predictor.

Note that the future trajectory problem can be classified into two specific scenarios according to the different settings of K. When $K = 1$, i.e., predicting the next location after the historical trajectory, the problem transformed as **Next Location Recommendation**. When $K > 1$, i.e., predicting K locations correlated to the future K time steps after the historical trajectory, the problem performs as **Multi-location Trajectory Prediction**.

3 METHODOLOGY

This section gives a general overview of the proposed STiSAN$^+$ architecture, including the macro-structure, inputs, and outputs. Then, we elaborate on the details of each component in STiSAN$^+$. Finally, we introduce the training process and parameter optimization strategy. We will omit some superscripts in the following statements to avoid redundancy, i.e., the specific user u or layer N.

3.1 Framework Overview

Figure 3 depicts the STiSAN$^+$ architecture. Macroscopically, it employs Multi-Modal Embedding (consists of ID Embedding, GPS Encoder, and Rotary Time Aware Position Encoder), stacks N multi-head Interval Aware Attention Block-based Encoder-Decoder architectures, and couples with N Spatial-Temporal Relation Memories.

The Encoder, Decoder, and Spatial-Temporal Relation Memory in STiSAN$^+$ take three types of data as input, respectively. During the training process, the inputs are historical trajectory $S_{enc} = \left\{c_1 \to c_2 \to \cdots \to c_{|S|}\right\}$, future trajectory $S_{dec} = \left\{c_{|S|+1} \to c_{|S|+2} \to \cdots \to c_{|S|+K}\right\}$ and the historical spatial-temporal relation matrix $\boldsymbol{R}_{his}$, and STiSAN$^+$ matches the output with negative samples for loss calculating. During the evaluating process, the input trajectory for Decoder is slightly different, which consists of candidate locations, and STiSAN$^+$ predicts the future trajectory-based on the descending order of matching scores.

3.2 Multi-Modal Embedding

As shown in Fig. 3(b), the Multi-Modal Embedding aims at converting the information in trajectories into high-dimension representations. For the historical trajectory S_{enc}, we separately process the location IDs, GPS coordinates, and timestamps with ID Embedding[2], GPS Encoder[3] and Rotary Time Aware Position Encoder (short for RoTAPE, whose design will be elaborated later). Take a check-in record $c = \langle u, l, g, t \rangle$ for instance, we first embed location ID l as $\boldsymbol{l} \in \mathbb{R}^{1 \times d/2}$ and then encode GPS g into $\boldsymbol{g} \in \mathbb{R}^{1 \times d/2}$. Meanwhile, we employ RoTAPE to transform timestamp $t = \left(t^d, t^h\right)$ into the time aware positional encoding matrix $\boldsymbol{P}_t \in \mathbb{R}^{d \times d}$. Then, the check-in c's representation $\boldsymbol{e} \in \mathbb{R}^{1 \times d}$ is generated as follows,

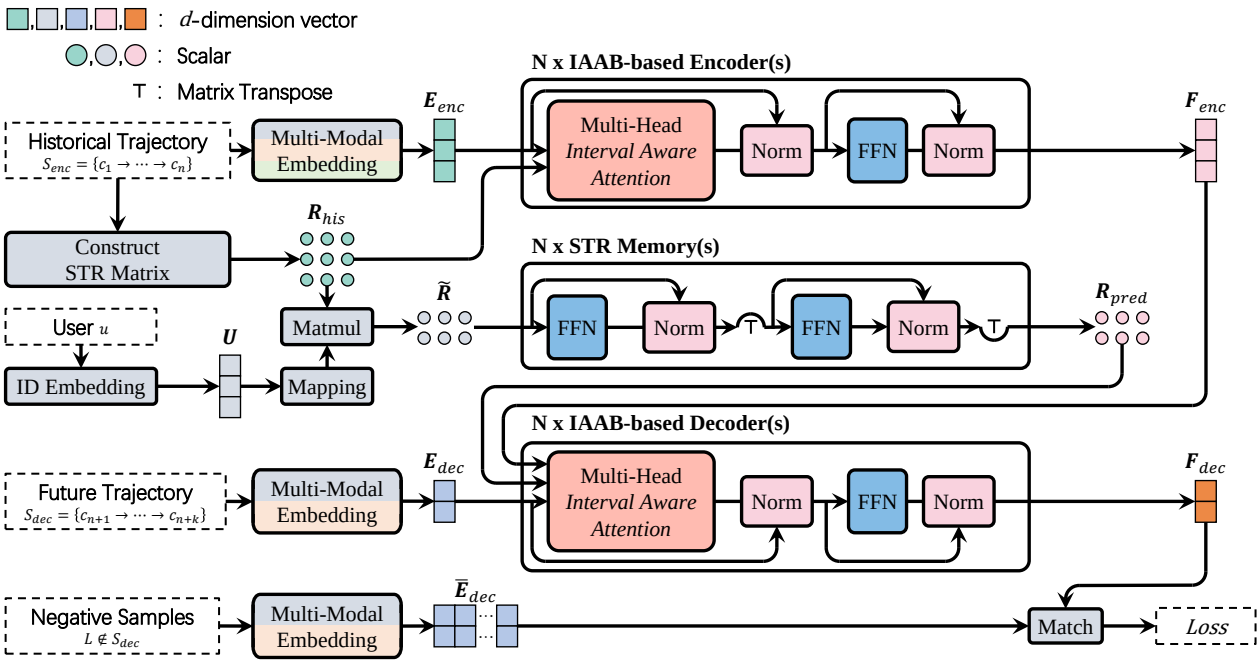
$$\boldsymbol{e} = \text{Concat}\left(\boldsymbol{l}, \boldsymbol{g}\right) \cdot \boldsymbol{P}_t. \tag{2}$$

For the future trajectory S_{dec} or negative samples $L \notin S_{dec}$, since the locations to be predicted do not possess temporal attributes, we only process the location IDs and GPS coordinates with the identical ID embedding and GPS encoder. As follows, the representation $\boldsymbol{e}$ of an item $c = \langle l, g \rangle$ in S_{dec} or $L \notin S_{dec}$ is the concatenation of $\boldsymbol{l}$ and $\boldsymbol{g}$,

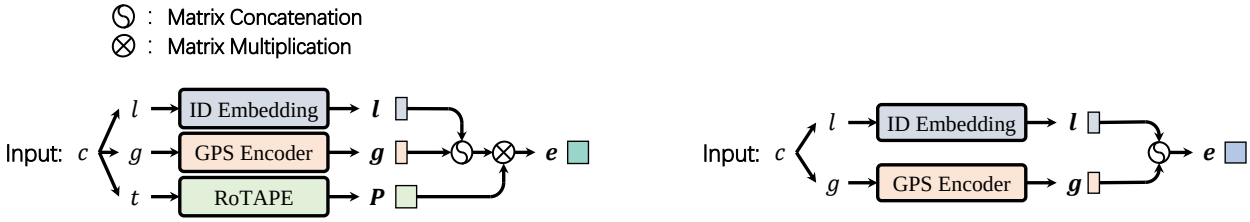
$$\boldsymbol{e} = \text{Concat}\left(\boldsymbol{l}, \boldsymbol{g}\right). \tag{3}$$

To unify various lengths of different historical trajectories, we set the maximum length n. The longer trajectories are split into several sub-sequences with "slide-window" of size n, and shorter sequences are repeatedly added with the "`padding`" term until their length grows to n. Thus, the historical trajectory S_{enc} can be multi-modal embedded into $\boldsymbol{E}_{enc} \in \mathbb{R}^{n \times d}$. Similarly, the future trajectory S_{dec} and negative samples are converted into $\boldsymbol{E}_{dec} \in \mathbb{R}^{k \times d}$ and $\overline{\boldsymbol{E}}_{dec} \in \mathbb{R}^{(k \cdot neg) \times d}$, separately, where k is the length of future trajectory and neg is the number of negative samples for each target location.

2. `torch.nn.Embedding()`.
3. The same as GeoSAN's geography encoder [6].



(a) STiSAN$^+$ employs Multi-Modal Embedding, stacks N multi-head Interval Aware Attention Block-based "Encoder-Decoder" architectures and couples N Spatial-Temporal Relation Memories along with residual connection and layer normalization.



(b) Multi-Modal Embedding consists of ID Embedding, GPS Encoder, and RoTAPE, where separately processing the location IDs, GPS coordinates, and timestamps in historical trajectory (left), future trajectory (right) and negative samples (right).

Fig. 3. The training process of STiSAN$^+$. Fig. 3(a) depicts the macro-structure and Fig. 3(b) shows the details of Multi-Modal Embedding.

3.3 Rotary Time Aware Position Encoder

Rotary Time Aware Position Encoder (RoTAPE) aims at promoting the attention mechanism to explicitly capture absolute time span and relative periodical pattern via injecting multi-level temporal information. In the following subsections, we first give the general form of RoTAPE, then explain how the attention mechanism can explicitly capture the encoded temporal information and discuss the computational complexity.

3.3.1 General Form

Given an embedding vector $\boldsymbol{e} \in \mathbb{R}^{1\times d}$ where d can exactly divisible by 4 and the multi-level timestamp t, RoTAPE injects the temporal information as follows,

$$\mathcal{R}(\boldsymbol{e}, t) = \boldsymbol{e} \cdot \boldsymbol{P}_t, \tag{4}$$

where $\mathcal{R}(\cdot)$ denotes RoTAPE. The time aware positional encoding matrix $\boldsymbol{P}_t \in \mathbb{R}^{d\times d}$ is

$$\boldsymbol{P}_t = \begin{bmatrix} \boldsymbol{P}_t^1 & \boldsymbol{P}_t^2 & \cdots & \boldsymbol{P}_t^{d/4} \end{bmatrix}, \tag{5}$$

where each sub-matrix $\boldsymbol{P}_t^i \in \mathbb{R}^{4\times 4}$ is

$$\boldsymbol{P}_t^i = \begin{bmatrix} \cos\left(t^d\theta_i\right) & -\sin\left(t^d\theta_i\right) & 0 & 0 \\ \sin\left(t^d\theta_i\right) & \cos\left(t^d\theta_i\right) & 0 & 0 \\ 0 & 0 & \cos\left(t^h\theta_i\right) & -\sin\left(t^h\theta_i\right) \\ 0 & 0 & \sin\left(t^h\theta_i\right) & \cos\left(t^h\theta_i\right) \end{bmatrix}, \tag{6}$$

and $\left\{\theta_i = 10000^{-4(i-1)/d}, i = 1, 2, \cdots, d/4\right\}$ are the predefined parameters. Next, we explain how the temporal information influences the self-attention mechanism.

3.3.2 An Illustrative Example in 4D Case

In this section, we first explore the question of whether or not the self-attention mechanism can explicitly capture the encoded temporal information.

The core idea behind the self-attention mechanism [19] is the inner product, which dynamically calculates the attention scores (or relevance) between queries and keys. Thus, the above question can be transformed as proving the following equation

$$\left\langle \mathcal{R}\left(\boldsymbol{q}_m, t_m\right), \mathcal{R}\left(\boldsymbol{k}_n, t_n\right) \right\rangle = g\left(\boldsymbol{q}_m, \boldsymbol{k}_n, \Delta t_{m-n}^d, \Delta t_{m-n}^h\right), \tag{7}$$

Algorithm 1 PyTorch-like Pseudo-code of RoTAPE

```
class RoTAPE(nn.Module):
  # d is the dimension which should be exact divisible by 4
  def __init__(self, d)
      super().__init__()
      # pre-defined theta in the positonal matrix P
      self.theta = torch.exp(0, d, 4) * (math.log(10000.0) / d)

  def get_p(self, t_d, t_h):
      # t_d and t_h are the day-level and hour-level timestamp separately
      cos_t_d, sin_t_d = torch.cos(t_d / self.theta), torch.sin(t_d / self.theta)
      cos_t_h, sin_t_h = torch.cos(t_h / self.theta), torch.sin(t_h / self.theta)
      return cos_t_d, sin_t_d, cos_t_h, sin_t_h

  def forward(self, x, t_d, t_h):
      # x is the input vector of dimension d
      cos_t_d, sin_t_d, cos_t_h, sin_t_h = self.get_p(t_d, t_h)
      x_1, x_2, x_3, x_4 = x[0::4], x[1::4], x[2::4], x[3::4]
      x_d_1 = x_1 * cos_t_d + x_2 * sin_t_d
      x_d_2 = x_2 * cos_t_d - x_1 * sin_t_d
      x_h_1 = x_3 * cos_t_h + x_4 * sin_t_h
      x_h_2 = x_4 * cos_t_h - x_3 * sin_t_h
      x = torch.cat([x_d_1, x_d_2, x_h_1, x_h_2], dim=-1)
      return x
```

where $\langle \cdot, \cdot \rangle$ is the inner product between two vectors, $\boldsymbol{q}_m, \boldsymbol{k}_n \in \mathbb{R}^{1\times d}$ are the query, key vectors of the m th, n th check-in record in a trajectory, and $t_{\{m,n\}}$ are the timestamps. In other words, we hope the inner product is equivalent to a function $g\{\cdot\}$ which only takes as input $\boldsymbol{q}_m$, $\boldsymbol{k}_n$ and multi-level temporal difference $\Delta t_{m-n}^{\{d,h\}}$.

Here, we present the simple case $d = 4$. Specifically, RoTAPE encodes the temporal information as follows,

$$\begin{aligned}
&\mathcal{R}\left(\boldsymbol{q}_m, t_m\right) = \boldsymbol{q}_m \cdot \boldsymbol{P}_{t_m} \\
&= \begin{bmatrix} q_m^1 q_m^2 q_m^3 q_m^4 \end{bmatrix} \begin{bmatrix} \cos\left(t_m^d\theta\right) & -\sin\left(t_m^d\theta\right) & 0 & 0 \\ \sin\left(t_m^d\theta\right) & \cos\left(t_m^d\theta\right) & 0 & 0 \\ 0 & 0 & \cos\left(t_m^h\theta\right) & -\sin\left(t_m^h\theta\right) \\ 0 & 0 & \sin\left(t_m^h\theta\right) & \cos\left(t_m^h\theta\right) \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{q}_m^1 \boldsymbol{q}_m^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_{t_m^d} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{t_m^h} \end{bmatrix}, \\
&\mathcal{R}\left(\boldsymbol{k}_n, t_n\right) = \boldsymbol{k}_n \cdot \boldsymbol{P}_{t_n} = \begin{bmatrix} \boldsymbol{k}_n^1 \boldsymbol{k}_n^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_{t_n^d} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{t_m^h} \end{bmatrix},
\end{aligned} \tag{8}$$

where q_m^i, k_n^i represents the element on the i th dimension. For the concise, we chunk the vectors and matrices along the dashed lines, where $\boldsymbol{q}_m^{\{1,2\}}, \boldsymbol{k}_n^{\{1,2\}} \in \mathbb{R}^{1\times 2}$, $\boldsymbol{P}_{t_{\{m,n\}}^{\{d,h\}}} \in \mathbb{R}^{2\times 2}$.

The inner product in Eq. 7 is

$$\begin{aligned}
&\left\langle \mathcal{R}\left(\boldsymbol{q}_m, t_m\right), \mathcal{R}\left(\boldsymbol{k}_n, t_n\right) \right\rangle \\
=&\boldsymbol{q}_m^1 \begin{bmatrix} \cos(\Delta t_{m-n}^d\theta) & -\sin(\Delta t_{m-n}^d\theta) \\ \sin(\Delta t_{m-n}^d\theta) & \cos(\Delta t_{m-n}^d\theta) \end{bmatrix} \boldsymbol{k}_n^{1\top} + \\
&\boldsymbol{q}_m^2 \begin{bmatrix} \cos(\Delta t_{m-n}^h\theta) & -\sin(\Delta t_{m-n}^h\theta) \\ \sin(\Delta t_{m-n}^h\theta) & \cos(\Delta t_{m-n}^h\theta) \end{bmatrix} \boldsymbol{k}_n^{2\top} \\
=&g\left(\boldsymbol{q}_m, \boldsymbol{k}_n, \Delta t_{m-n}^d, \Delta t_{m-n}^h\right),
\end{aligned} \tag{9}$$

where the relative temporal difference $\Delta t_{m-n}^{\{d,h\}}$ are explicitly encoded into the inner product with sinusoidal functions. The goal in Eq. 7 is achieved.

According in Eq. 9, we can see that $\Delta t_{m-n}^{\{d,h\}}$ are jointly encoded to represent the absolute time interval. Notably, when $\Delta t_{m-n}^d \neq 0$ and $\Delta t_{m-n}^h = 0$, i.e., the m th and n th check-in took place at the same time of different days, the hour-level difference part loses efficacy by degrading as identity matrix. The product focuses on mining the day-level periodical pattern among check-ins with Δt_{m-n}^d.

3.3.3 Complexity Discussion

Recall the sparaity in $\boldsymbol{P}_t$, we derive an equivalent and more efficient form of Eq. 4 as Eq. 10. The PyTorch-like pseudo-code of RoTAPE is presented in Algorithm 1.

$$\mathcal{R}\left(\boldsymbol{e}, t\right) = \begin{bmatrix} e^1 \\ e^2 \\ e^3 \\ e^4 \\ \vdots \\ e^{d-3} \\ x^{d-2} \\ e^{d-1} \\ x^d \end{bmatrix} \times \begin{bmatrix} \cos(t^d\theta_1) \\ \cos(t^d\theta_1) \\ \cos(t^h\theta_1) \\ \cos(t^h\theta_1) \\ \vdots \\ \cos(t^d\theta_{d/4}) \\ \cos(t^d\theta_{d/4}) \\ \cos(t^h\theta_{d/4}) \\ \cos(t^h\theta_{d/4}) \end{bmatrix} + \begin{bmatrix} e^2 \\ -e^1 \\ e^4 \\ -e^3 \\ \vdots \\ e^{d-2} \\ -e^{d-1} \\ e^d \\ -e^{d-3} \end{bmatrix} \times \begin{bmatrix} \sin(t^d\theta_1) \\ \sin(t^d\theta_1) \\ \sin(t^h\theta_1) \\ \sin(t^h\theta_1) \\ \vdots \\ \sin(t^d\theta_{d/4}) \\ \sin(t^d\theta_{d/4}) \\ \sin(t^h\theta_{d/4}) \\ \sin(t^h\theta_{d/4}) \end{bmatrix}. \tag{10}$$

Along the efficient implementation, RoTAPE only requires nd MACs to (Multiply-Accumulate Operations) inject the temporal information which is negligible to subsequent attention operator. Moreover, RoTAPE utilizes the fixed and pre-defined θ to avoid increasing the parameter scale. Thus, RoTAPE conforms to the lightweight property.

3.4 Construct Spatial-Temporal Relation Matrix

Before devoting into the details of encoder part, we first construct the historical spatial-temporal relation matrix $\boldsymbol{R}_{his} \in \mathbb{R}^{n\times n}$ according to the timestamps and GPS coordinates in S_{enc} for recording the relative spatial-temporal proximity,

$$\boldsymbol{R}_{enc} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix},$$

where the spatial-temporal relation between the i th and j th check-in record is denoted as r_{ij}.

To achieve r_{ij}, we first define $\hat{r}_{ij} = \Delta t_{ij} + \Delta d_{ij}$ consists of the corresponding time and geography intervals. Then, we consider that the precise intervals might not be useful beyond a certain threshold [11], and clip Δt_{ij}, Δd_{ij} by maximum time span k_t and distance k_d, respectively:

$$\begin{aligned}
\Delta t_{ij} &= \min\left(k_t, \mid t_i - t_j \mid\right), \\
\Delta d_{ij} &= \min\left(k_d, \text{Haversine}\left(g_i, g_j\right)\right),
\end{aligned} \tag{11}$$

where $\text{Haversine}\left(\cdot\right)$ calculates the physical distance between two GPS coordinates. Moreover, we argue that the relations should be inverse to their intervals and implement the point by $r_{ij} = \hat{r}_{max} - \hat{r}_{ij}$ where $\hat{r}_{max}$ is the max value among $\hat{r}_{ij}$.

3.5 IAAB-based Encoder

To impel the model by attaching important spatial information among local locations and providing more explainable prediction, we introduce the historical relation matrix $\boldsymbol{R}_{his}$ into the attention mechanism as the inductive bias. As shown in Fig. 3(a), the multi-head Interval Aware Attention Block (IAAB)-based Encoder alternates a multi-head interval aware attention layer and a feed-forward network along with the residual connection and layer normalization.

Algorithm 2 PyTorch-like Pseudo-code of IAAB

```
def interval_attention(q, k, v, str_mat):
    # q, k, v are the query, key, value representations
    # str_mat is the spatial-temporal relation matrix
    scores = torch.matmul(q, k.transpose(0, 1)) / math.sqrt(d)
    scores = scores + str_mat
    probs = F.softmax(scores, dim=-1)
    return torch.matmul(probs, v)

class IAAB(nn.Module):
    def __init__(self, d, h):
        super().__init__()
        # d is the latent dimension
        # h is the number of attention head
        self.d_h = d // h # the dimension for each head
        self.h = h
        self.map_q, self.map_k, self.map_v = nn.Linear(d, d), nn.Linear(d, d), nn.
            Linear(d, d)
        self.merge = nn.Linear(d, d)
    def forward(self, x, str_mat):
        # x is the sequence representation
        # str_mat is the spatial temporal relation matrix
        q, k, v = self.map_q(x), self.map_k(x), self.map_v(x)
        q, k, v = q.view(self.h, -1, self.d_h), k.view(self.h, -1, self.d_h), v.
            view(self.h, -1, self.d_h)
        str_mat = F.softmax(str_mat, dim=-1)
        x = interval_attention(q, k, v, str_mat)
        x = x.transpose(1, 2).contiguous().view(-1, self.h * self.d_h)
        x = self.merge(x)
        return x
```

Algorithm 3 PyTorch-like Pseudo-code of STR Memory

```
class FFN(nn.Module):
    def __init__(self, d, e)
        super().__init__()
        # d is the dimension of input
        # e is expansion factor to decide the intermediate dimension between two
            linear layers
        self.w_1 = nn.Linear(d, d * e)
        self.w_2 = nn.Linear(d * e, d)
        self.act = nn.ReLU()
    def forward(self, x):
        x = self.w_2(self.act(self.w_1(x)))
        return x

class STRMem(nn.Module):
    def __init__(self, n, k):
        super().__init__()
        # n is the length of historical sequence
        # k is the length of predicted trajectory
        self.FFN_h = FFN(n, k)
        self.FFN_v = FFN(K, n)
    def forward(self, p_str_mat):
        # p_str_mat is the personalized historical spatial-temporal relation matrix
        p_str_mat = self.FFN_h(p_str_mat)
        p_str_mat = self.FFN_v(p_str_mat.tranpose(0, 1)).transpose(0, 1)
        return p_str_mat
```

3.5.1 Multi-head Interval Aware Attention Layer

The multi-head interval aware attention layer takes as input historical trajectory representation $\boldsymbol{E}_{enc}$, corresponding relation matrix $\boldsymbol{R}_{his}$ and outputs the attentive results $\boldsymbol{A}_{enc}$. It can be formulated as Equation (12),

$$\boldsymbol{A}_{enc} = \mathrm{IAAB}\left(\boldsymbol{E}_{enc}, \boldsymbol{R}_{his}\right). \tag{12}$$

Specifically, $\mathrm{IAAB}\left(\cdot\right)$ first converts $\boldsymbol{E}_{enc}$ into query, key, and value matrices through three distinct matrices,

$$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} = \boldsymbol{E}_{enc}\boldsymbol{W}_Q, \boldsymbol{E}_{enc}\boldsymbol{W}_K, \boldsymbol{E}_{enc}\boldsymbol{W}_V \tag{13}$$

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{W}_{\{Q,K,V\}} \in \mathbb{R}^{d\times d}$. Then, it chunks query, key, value into h heads along with the last dimension d, i.e., $\{\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\}_i \in \mathbb{R}^{h\times n\times d/h}, i \in [1, h]$. For each head, the interval aware attention $\mathrm{IA}\left(\cdot\right)$ explicitly combines the attention map (i.e., sequential dependency) with the relation matrix by point-wise addition. As follows,

$$\begin{aligned}\boldsymbol{A}_{enc_i} &= \mathrm{IA}\left(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i, \boldsymbol{R}_{his}\right) \\ &= \mathrm{Softmax}(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^\top}{\sqrt{d/h}} + \boldsymbol{R}_{his})\boldsymbol{V}_i,\end{aligned} \tag{14}$$

where $\boldsymbol{A}_{enc_i} \in \mathbb{R}^{n\times d}$ is the attentive result of head i. The results of all heads are merged with concatenation and linear layers. The pseudo-code of IAAB is shown in Algorithm 2. Note that we scale $\boldsymbol{R}_{his}$ with $\mathrm{Softmax}$ before the addition for normalization. In this way, our multi-head interval aware attention layer utilizes the spatial-temporal relation to provide the attention map with positive revisions, which strengthens the model's ability to consider the relative spatial proximity among local locations. Rather than embedding the relation into high-dimension space, the explicit combination improves the models' interpretability.

3.5.2 Complexity Dicussion

IAAB utilizes the relation matrix $\boldsymbol{R}_{his}$ to record the spatial-temporal information, which is calculated by the timestamps and gps coordinates in the historical sequence. Thus, there are no additional learnable parameters in IAAB. Compared to the vanilla self-attention mechanism with h heads, IAAB only requires extra $h \cdot \frac{n^2}{2}$ MACs to inject the relation via the element-wise addition, which is negligible to scaled-dot product ($h \cdot \frac{n^2}{2} \cdot d$ MACs). Thus, IAAB conforms the requirement of lightweight.

3.5.3 Feed-Forward Network

We employ a 2-layer point-wise feed-forward network to encode the interactions between different dimensions and endow the attentive results with non-linearity [19]. It consists of two distinct linear layers and the activation function ReLU. It can be expressed as the following equation,

$$\boldsymbol{F}_{enc} = \mathrm{FFN}\left(\boldsymbol{A}_{enc}\right) = \max\left(0, \boldsymbol{A}_{enc}\boldsymbol{W}_1 + b_1\right)\boldsymbol{W}_2 + b_2, \tag{15}$$

where $\boldsymbol{F}_{enc} \in \mathbb{R}^{n\times d}$, $\boldsymbol{W}_1 \in \mathbb{R}^{d\times 4d}$, $\boldsymbol{W}_2 \in \mathbb{R}^{4d\times d}$ and $b_1, b_2 \in \mathbb{R}^{1\times d}$ are the learned bias terms. The implemented details are revealed as `FFN()` in Algorithm 3

3.6 Spatial-Temporal Relation Memory

The idea behind Spatial-Temporal Relation Memory (STR Memory) is endowing interactions inside the historical relation matrix for mining individual behavior patterns and predicting future spatial-temporal relations.

Specifically, each STR Memory alternates two types of FFNs in different directions. Before feeding the historical matrix into STR Memory, STiSAN$^+$ first personalized maps $\boldsymbol{R}_{his}$ into $\tilde{\boldsymbol{R}} \in \mathbb{R}^{k\times n}$. It can be formulated as,

$$\tilde{\boldsymbol{R}} = (\boldsymbol{U}\boldsymbol{W}_{map})^\top \boldsymbol{R}_{his}, \tag{16}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times d}$ is user embedding and $\boldsymbol{W}_{map} \in \mathbb{R}^{d\times k}$. Then, STR Memory employs horizontal and vertical fully connected linear layers, denoted as FFN_h and FFN_v separately, to predict future spatial-temporal relations,

$$\begin{aligned}\boldsymbol{R}_h &= \mathrm{FFN}_h(\tilde{\boldsymbol{R}}) = \max\left(0, \tilde{\boldsymbol{R}}\boldsymbol{W}_3 + b_3\right)\boldsymbol{W}_4 + b_4, \\ \boldsymbol{R}_v &= \mathrm{FFN}_v(\boldsymbol{R}_h^\top) = \max\left(0, \boldsymbol{R}_h^\top\boldsymbol{W}_5 + b_5\right)\boldsymbol{W}_6 + b_6,\end{aligned} \tag{17}$$

where $\boldsymbol{R}_h \in \mathbb{R}^{k\times n}$ and $\boldsymbol{R}_v \in \mathbb{R}^{n\times k}$. After transposing the output of FFN_v, the predicted spatial-temporal relation of future trajectory is denoted as $\boldsymbol{R}_{pred} \in \mathbb{R}^{k\times n}$. Specifically, we set the original dimension in FFN_h as n and scale it by k times while setting the original dimension in FFN_v as k and scale it by n times. The pseudo-code of Spatial-Temporal Relation Memory is shown in Algorithm 3.

3.7 IAAB-based Decoder

As reported in [14], the prediction might be sub-optimal if we directly match the output of Encoder $\boldsymbol{F}_{enc}$ with candidate locations. Therefore, we employ the IAAB-based Decoder, whose structure is identical to the Encoder, to introduce the predicted spatial-temporal relations and improve the representations of candidates. It can be described as the following formulaic expression,

$$\boldsymbol{A}_{dec} = \mathrm{IAAB}\left(\boldsymbol{E}_{dec}, \boldsymbol{F}_{enc}, \boldsymbol{R}_{pred}\right), \tag{18}$$

$$\boldsymbol{F}_{dec} = \mathrm{FFN}\left(\boldsymbol{A}_{dec}\right), \tag{19}$$

where $\boldsymbol{A}_{dec} \in \mathbb{R}^{k\times n}$ is the output of attention layer and $\boldsymbol{F}_{dec} \in \mathbb{R}^{k\times d}$ is the predicted trajectory representations. Notably, the slight difference lies in Decoder is that the query matrix is mapped from $\boldsymbol{E}_{dec}$, while key and value matrices are mapped from $\boldsymbol{F}_{enc}$. The interval-aware attention layer in Decoder can be viewed as re-weighting each check-in record in historical trajectory according to the sequential dependency and predicted spatial-temporal relations when generating future trajectory representations.

3.8 Matching and Ranking

Recall that the output of Decoder is denoted as $\boldsymbol{F}_{dec}$, each row $\boldsymbol{F}_{dec_i} \in \mathbb{R}^{1\times d}$ stands for the representation of i th location in future trajectory. We calculate the matching score $y_{i,j}$ over the candidate location j with the following function,

$$y_{i,j} = f\left(\boldsymbol{F}_{dec_i}, \boldsymbol{L}_j\right), \tag{20}$$

where $\boldsymbol{L}_j \in \mathbb{R}^{1\times d}$ is the representation of location j and $f(\cdot)$ is the inner production. As shown in the right half of Fig. 3(b), $\boldsymbol{L}_j$ is the concatenation of location ID embedding and GPS coordinates encoding. After matching all k output vectors with candidate locations, the model predicts a user u's future trajectory, which consists of k locations, according to the descending order of matching scores.

3.9 Model Training

The binary cross-entropy loss function is widely used for optimizing sequential models [11], [20]. However, for the sake of efficient training, only one negative sample is randomly picked from all un-visited locations, which cannot make fully effective use of the large number of negative samples [6]. Thus, for each future visited location l_i, we utilize K Nearest Neighbour sampler [6] to retrieve the K nearest locations $\overline{l_{ij}}$ around it as negative samples, and we introduce the following binary cross-entropy loss function,

$$Loss = -\sum_{S^u\in S}\sum_{i=1}^{k}(\log\sigma(y_{i,l_i}) + \sum_{j=1}^{K}\log(1-\sigma(y_{i,\overline{l_{ij}}}))), \tag{21}$$

where S is the set of all training sequences.

4 Experiments

In this section, we first introduce the datasets, baselines, and evaluation metrics and implement details of STiSAN$^+$. Then, we analyze the overall performances under two scenarios. Besides validating the effectiveness of our approaches via the ablation study, we validate the extensibility, interpretability, and efficiency of RoTAPE and IAAB.

TABLE 1
The Statistics of Six Datasets (After Pre-processed)

Dataset	Gowalla	Yelp	Brightkite
# Users	31,708	55,067	5,247
# Locations	131,329	80,128	48,181
# Check-ins	2,963,373	2,830,574	1,699,579
Sparsity	93.93%	99.94%	99.33%
Avg. Seq. Len.	93.46	51.41	323.91
Dataset	TKY	Weeplaces	NYC
# Users	2,279	1,362	1,010
# Locations	15,177	18,364	5,135
# Check-ins	985,364	650,690	140,229
Sparsity	97.15%	97.40%	97.30%
Avg. Seq. Len.	432.37	477.75	138.80

Moreover, we explore the model's sensitivity with respect to hyper-parameter settings. In summary, we conduct a large number of experiments to answer the following questions:

- **RQ1** How is the performance of STiSAN$^+$ in the scenarios of Next Location Recommendation and Multi-location Trajectory Prediction?
- **RQ2** How is the effectiveness of RoTAPE, IAAB, and STR Memory under the STiSAN$^+$ framework?
- **RQ3** How is the influence brought by RoTAPE and IAAB when extending them to the vanilla self-attention network? Can the information contained in RoTAPE and IAAB be effectively captured by SAN without extra significant computational cost?
- **RQ4** How is STiSAN$^+$'s sensitivity with respect to different hyper-parameter settings, i.e., maximum time and spatial interval thresholds (k_t, k_d)?

4.1 Datasets

We choose six public LBSN datasets: Gowalla[4], Brightkite[5], Weeplaces[6], Yelp[7], NYC[8] and TKY[9] to evaluate our proposed model. In order to ensure the quality of datasets, we filter out inactive users and un-popular locations during the pre-processing. Specifically, we remove the users who visit less than 20 locations and the locations that have interacted with fewer than 10 times, as shown in Table 1.

4.2 Baselines

To evaluate the effectiveness of our proposed model, we compare it with various existing methods. For better understanding, we now briefly introduce our competitors.

GRU4Rec [21] is a basic GRU-based model for sequential recommendation. **Caser** [22] is employs convolution filters to capture sequential dependency from local and global perspectives. **STGN** [23] designs special gates for capturing the spatial-temporal correlations between successive locations. **Flashback** [24] enhances RNN by reweighting the hidden states according to the spatial-temporal contexts. **SASRec**

4. https://snap.stanford.edu/data/loc-gowalla.html
5. https://snap.stanford.edu/data/loc-brightkite.html
6. https://www.yongliu.org/datasets.html
7. https://www.yelp.com/dataset
8. https://www.kaggle.com/chetanism/foursquare-nyc-and-tokyo-checkin-dataset
9. https://www.kaggle.com/chetanism/foursquare-nyc-and-tokyo-checkin-dataset

TABLE 2
Next Location Recommendation Performance Comparison (the best scores are boldfaced, and the second scores are underlined)

Dataset	Gowalla				Yelp				Brightkite			
Metric	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Caser	0.2327	0.1876	0.3688	0.2049	0.2530	0.2017	0.3350	0.2176	0.3164	0.2123	0.4302	0.2745
GRU4Rec	0.3150	0.2302	0.4334	0.2683	0.3392	0.2481	0.4761	0.2812	0.3889	0.2991	0.5012	0.3356
STGN	0.1655	0.1171	0.2915	0.1603	0.2574	0.1674	0.3475	0.2121	0.2721	0.1892	0.3614	0.2375
Flashback	0.3391	0.2421	0.4497	0.2771	0.3425	0.2649	0.4886	0.2874	0.4312	0.3270	0.5469	0.3667
SASRec	0.3288	0.2401	0.4505	0.2794	0.3296	0.2414	0.4621	0.2734	0.4151	0.3193	0.5325	0.3573
Bert4Rec	0.3317	0.2440	0.4652	0.2853	0.3047	0.2181	0.3995	0.2734	0.3950	0.3051	0.5036	0.3424
TiSASRec	0.3426	0.2525	0.4581	0.2897	0.3425	0.2521	0.4797	0.2851	0.4488	0.3493	0.5548	0.3834
LSPSL	0.3692	0.2711	0.4728	0.2943	0.3507	0.2591	0.4926	0.2865	0.4671	0.3422	0.5587	0.3846
CTLE	0.3389	0.2347	0.4593	0.2988	0.3264	0.2318	0.4533	0.2470	0.4510	0.3533	0.5614	0.3711
CFPRec	0.3576	0.2491	0.4632	0.3104	0.3489	0.2418	0.4679	0.2535	0.4625	0.3690	0.5642	0.3849
GeoSAN	0.3838	0.2837	0.5120	0.3249	0.3789	0.2804	0.5152	0.3128	0.4591	0.3504	0.5593	0.3921
STAN	0.4369	0.3544	0.5384	0.3864	0.3564	0.2713	0.4535	0.2874	0.4736	0.3819	0.5670	0.4263
STiSAN	0.4617	0.3721	0.5679	0.4053	0.3972	0.2824	0.5420	0.3290	0.5310	0.4339	0.6512	0.4727
STiSAN$^+$	**0.4877**	**0.3815**	**0.5843**	**0.4217**	**0.4276**	**0.3018**	**0.5689**	**0.3410**	**0.5596**	**0.4649**	**0.6753**	**0.4804**
Improv.	11.63%	7.65%	8.53%	9.14%	12.85%	7.63%	10.42%	9.02%	18.16%	16.73%	19.20%	12.69%
Dataset	TKY				Weeplaces				NYC			
Metric	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Caser	0.2251	0.1429	0.3085	0.1876	0.2735	0.1964	0.3712	0.2403	0.2214	0.1321	0.3156	0.1883
GRU4Rec	0.2522	0.1882	0.3483	0.2101	0.3084	0.2161	0.4156	0.2511	0.2426	0.1689	0.3475	0.2027
STGN	0.1962	0.1245	0.2515	0.1584	0.1864	0.1325	0.3481	0.1657	0.1833	0.1032	0.2047	0.1277
Flashback	0.2725	0.1732	0.3694	0.2188	0.3369	0.2293	0.4312	0.2638	0.2766	0.1863	0.3693	0.2148
SASRec	0.2643	0.1874	0.3686	0.2119	0.3157	0.2323	0.4361	0.2711	0.2574	0.1801	0.3792	0.2193
Bert4Rec	0.2609	0.1729	0.3644	0.2012	0.2902	0.2105	0.3997	0.2614	0.1564	0.0909	0.2248	0.1527
TiSASRec	0.2851	0.2154	0.3858	0.2379	0.3363	0.2465	0.4545	0.2850	0.2446	0.1741	0.3604	0.2114
LSPSL	0.2899	0.2025	0.3971	0.2238	0.3396	0.2471	0.4681	0.2884	0.2615	0.1886	0.3841	0.2376
CTLE	0.2736	0.1967	0.3546	0.2056	0.3128	0.2106	0.4359	0.2431	0.2537	0.1830	0.3749	0.2305
CFPRec	0.2918	0.2115	0.3792	0.2371	0.3254	0.2310	0.4461	0.2698	0.2736	0.1919	0.3942	0.2411
GeoSAN	0.2831	0.2083	0.4025	0.2370	0.3480	0.2677	0.4699	0.3069	0.2624	0.1771	0.4109	0.2254
STAN	0.3132	0.2215	0.4270	0.2589	0.3276	0.2341	0.4349	0.2830	0.2802	0.1953	0.4216	0.2431
STiSAN	0.3234	0.2359	0.4379	0.2722	0.4332	0.3437	0.5558	0.3833	0.2921	0.2092	0.4337	0.2551
STiSAN$^+$	**0.3659**	**0.2600**	**0.4783**	**0.2964**	**0.4451**	**0.3543**	**0.5762**	**0.4036**	**0.3108**	**0.2149**	**0.4526**	**0.2774**
Improv.	16.83%	17.38%	12.01%	14.48%	27.90%	32.35%	22.62%	31.51%	10.92%	10.04%	7.35%	14.11%

[20] first applies the self-attention mechanism for sequential recommendation. **Bert4Rec** [25] models sequential dependency with the bi-directional self-attention mechanism. **TiSASRec** [11] proposes the time-aware self-attention layer to integrate temporal information. **CFPRec** [12] models the multi-step future preferences and mimics the activity planning before predicting the next visit. **GeoSAN** [6] exploits a novel self-attention-based geography encoder that shows the state-of-the-art performance in modeling exact locations. **STAN** [14] is a state-of-the-art sequential location recommender that explicitly models the relative spatial-temporal with the proposed bi-layer attention architecture. **STiSAN** [26] is a state-of-the-art sequential recommender that first employs Time Aware Position Encoder and Interval Aware Attention Block. **LSPSL** [27] introduces two self-supervised optimization objectives to improve the long- and short-term preference modeling. **CTLE** [13] is a bi-directional attention pre-trained location embedding model which incorporates the spatial-temporal context in trajectories.

4.3 Metrics

Towards the Next Location Recommendation, we adopt two widely-used metrics, Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [28], to measure how well the target locations in the test are ranked. We report the HR and NDCG at $n = \{5, 10\}$. For the Multi-location Trajectory Prediction, we follow [29], [30], [31] and adopt F1 Score [28] ($n = \{3, 5\}$) to balance the precision and recall of the ground-truth locations in the predicted trajectory For all metrics, the larger values indicate better performance.

4.4 Settings

For data partition, we take each user's most recent k locations in the whole trajectory for evaluation and all the locations prior to targets for training. Specifically, we utilize a "slide window" of size $n+k$ to generate training instances. Longer trajectories will be clipped into sub-sequences of length n, and shorter trajectories will be repeatedly added "padding" location in the head until their lengths grow to n. We set the maximum trajectory length as 100.

The implementation details of our STiSAN$^+$ are listed as follows. For the latent representations, we set the dimensions of location embedding and GPS coordinates encoding to 128, and the sequence dimension d is concatenated to 256. For the preference modeling, we stack $N = 2$ IAAB-based Encoder-Decoder structure of $h = 2$ heads and $N = 2$ STR Memories. We conduct all experiments on a server with 64GB RAM, 12-core AMD 9 Ryzen 5900X CPU and Nvidia RTX 3090 GPU. The code is available at https://github.com/jiangyiheng1/STiSAN_v2.pytorch.

4.5 Validations and Discussions

4.5.1 Next Location Recommendation Performance (RQ1)

For the sake of efficient evaluation, we choose the 100 nearest neighbors around the target location as negative candidates, and the metrics are calculated according to the rank of these 101 locations. Table 2 summarizes the overall performance of the next location recommendation. Specifically, we have the following observations:

Attention-based methods generally have more stable and better performance than CNN or RNN-based models. It

TABLE 3
Multi-location Trajectory Prediction Accuracy Comparison (the best scores are boldfaced, and the second scores are underlined)

Dataset	Gowalla		Yelp		Brightkite		TKY		Weeplaces		NYC	
Metric	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5
GRU4Rec	0.1819	0.2020	0.1460	0.1462	0.3270	0.3506	0.4344	0.4008	0.2020	0.2237	0.3101	0.3105
STGN	0.1343	0.1504	0.1017	0.1251	0.3113	0.3442	0.3740	0.3511	0.1570	0.1977	0.2786	0.2866
Flashback	0.1968	0.2155	0.1624	0.1539	0.3480	0.3824	0.4236	0.3871	0.2193	0.2249	0.3356	0.3241
SASRec	0.1887	0.1182	0.1894	0.1987	0.1872	0.2683	0.4501	0.3928	0.2237	0.2665	0.3195	0.3310
Bert4Rec	0.1806	0.1993	0.1434	0.1446	0.3385	0.3619	0.3624	0.3500	0.2364	0.2622	0.2968	0.3116
TiSASRec	0.1818	0.2016	0.1378	0.1403	0.3322	0.3730	0.4329	0.3619	0.2437	0.2829	0.3134	0.3249
LSPSL	0.1960	0.1583	0.1733	0.1666	0.3257	0.3578	0.4271	0.3901	0.2596	0.2716	0.2833	0.2674
CTLE	0.2159	0.2287	0.1538	0.1787	0.3687	0.3856	0.4257	0.3588	0.2676	0.2974	0.2786	0.3425
CFPRec	0.2276	0.2387	0.1633	0.1678	0.3824	0.3976	0.4472	0.3599	0.2843	0.3076	0.2977	0.3249
GeoSAN	0.1871	0.2183	0.1422	0.1454	0.3585	0.4022	0.4134	0.3465	0.2715	0.3110	0.2872	0.3339
STAN	0.2389	0.2471	<u>0.1743</u>	<u>0.1714</u>	<u>0.4120</u>	<u>0.4381</u>	0.4549	<u>0.3673</u>	0.3055	0.3359	0.3281	0.3526
STiSAN	<u>0.2530</u>	<u>0.2570</u>	0.1692	0.1675	0.3718	0.4046	<u>0.4628</u>	0.3570	<u>0.3180</u>	<u>0.3387</u>	<u>0.3372</u>	<u>0.3588</u>
STiSAN$^+$	**0.2753**	**0.2869**	**0.2076**	**0.2033**	**0.5638**	**0.5591**	**0.5235**	**0.4417**	**0.3981**	**0.3710**	**0.4027**	**0.3998**
Improv.	8.81%	11.63%	19.10%	18.61%	36.84%	27.62%	13.12%	20.26%	25.19%	9.54%	18.03%	11.43%

TABLE 4
Ablation Study of STiSAN$^+$ in Multi-location Trajectory Prediction (the best scores are boldfaced)

Dataset	Gowalla		Yelp		Brightkite		TKY		Weeplaces		NYC	
Metric	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5
Original	**0.2753**	**0.2869**	0.2076	0.2033	**0.5638**	**0.5591**	**0.5235**	**0.4417**	**0.3981**	**0.3710**	**0.4027**	**0.3998**
I. -RoTAPE	0.2256	0.2347	0.1830	0.1895	0.4843	0.4739	0.4418	0.4072	0.3419	0.3671	0.3412	0.3704
II. -IAAB	0.2431	0.2497	**0.2230**	**0.2428**	0.3252	0.4382	0.4472	0.4190	0.3142	0.3472	0.3621	0.3724
III. -STR Memory	0.2369	0.2547	0.1834	0.1947	0.4441	0.5230	0.4665	0.4353	0.3395	0.3562	0.3590	0.3855

proves that the self-attention mechanism can better capture sequential dependency. However, we notice that Bert4Rec performs poorly over all datasets. The possible reason is that Bert-based methods' auto-encoding training manner and masking strategy are originally designed for the Sequence-to-Sequence task. However, in the Next Location Recommendation task, the training objective changes to Sequence-to-One, which might lead to the biased mask representation and further carry on the negative impact during the prediction stage. Similar phenomenons are also reported in [32].

Among the methods that focus on temporal information, the pre-trained CTLE is inferior to TiSASRec and CFPRec. The possible reason is that it splits the trajectory as short sessions in different days, which attaches more importance to the short-term preferences. Credited to the tile-map-based geographical encoding, LSPSL has better performance than CFPRec on most datasets.

STAN is the strongest competitor. Credited to the embedded spatial-temporal intervals and modified attention mechanism, STAN gains higher accuracy on most datasets. However, the geographical modeling and importance-based negative sampling help GeoSAN make up for the lack of temporal information on Yelp and Weeplaces.

Notably, our previous STiSAN consistently outperforms all baselines with a large margin on all six datasets, and STiSAN$^+$ further improves the recommendation performance. It is mainly due to the new RoTAPE, which can explicit model the multi-level time intervals to reflect absolute time span and relative periodical behavior patterns. At the same time, TAPE is only sensitive to the absolute one.

4.5.2 Multi-location Trajectory Prediction Accuracy (RQ1)

Since all of the compared methods are designed for the next location recommendation task (single target), we add an extra head to convert their outputs as multiple targets and retrain them under our framework to their best performance. For each target location in the future trajectory of length k, we choose 50 nearest neighbors around it as candidates. We set $k = 3$, $k = 5$ in this experiment, and the metrics are calculated according to the rank of these 153, 255 locations. The experimental results are summarized in Table 3.

According to the last row in Table 3, our STiSAN$^+$ outperforms all baselines remarkably over all datasets. Compared to STiSAN, the significant improvements demonstrate that the STR Memory and IAAB-based Decoder enhance STiSAN$^+$ to simultaneously possess the ability to predict the next location and multi-location future trajectory. It reveals that our STiSAN$^+$ possesses the potential to perform like a foundation framework towards both Sequence-to-One and Sequence-to-Sequence tasks. Moreover, although CFPRec does not introduce geographical information, they achieve competitive performances. This is because the future multi-step preference modeling benefits the multi-location trajectory prediction.

4.5.3 Ablation Study (RQ2)

We choose STiSAN$^+$ as the base model (Original) and consider the following variants to verify the effectiveness of various components under the scenario of Multi-Location Trajectory Prediction:

- I. -RoTAPE: We replace the Rotary Time Aware Position Encoder with the positional encoding [19].
- II. -IAAB: We replace the interval aware attention layer with the multi-head self-attention [19].
- III. -STR Memory: We remove the Spatial-Temporal Relation Memory and only convert the historical relation matrix with Equation (16).

TABLE 5
Comparison among Various Positional Representation Methods. The best and second scores are boldfaced and underlined, respectively.

Dataset	Brightkite						
Method	FixedPE	LearnedPE	RoPE	CTLE	CFPRec	TAPE	RoTAPE
Metric	HR@5	HR@5	HR@5	HR@5	HR@5	HR@5	HR@5
$n = 32$	0.02535	0.04174	0.04283	0.04441	<u>0.04479</u>	0.04117	**0.04583**
$n = 64$	0.01982	<u>0.02611</u>	<u>0.02611</u>	0.02516	0.02516	0.02420	**0.02763**
$n = 128$	0.05279	0.05351	0.05203	0.05364	0.05489	0.05279	**0.05508**
$n = 256$	0.05641	0.05698	0.05603	0.05603	<u>0.05712</u>	0.05538	**0.05775**
$n = 512$	0.05413	0.05679	0.05413	<u>0.05737</u>	0.05165	0.05698	**0.05984**

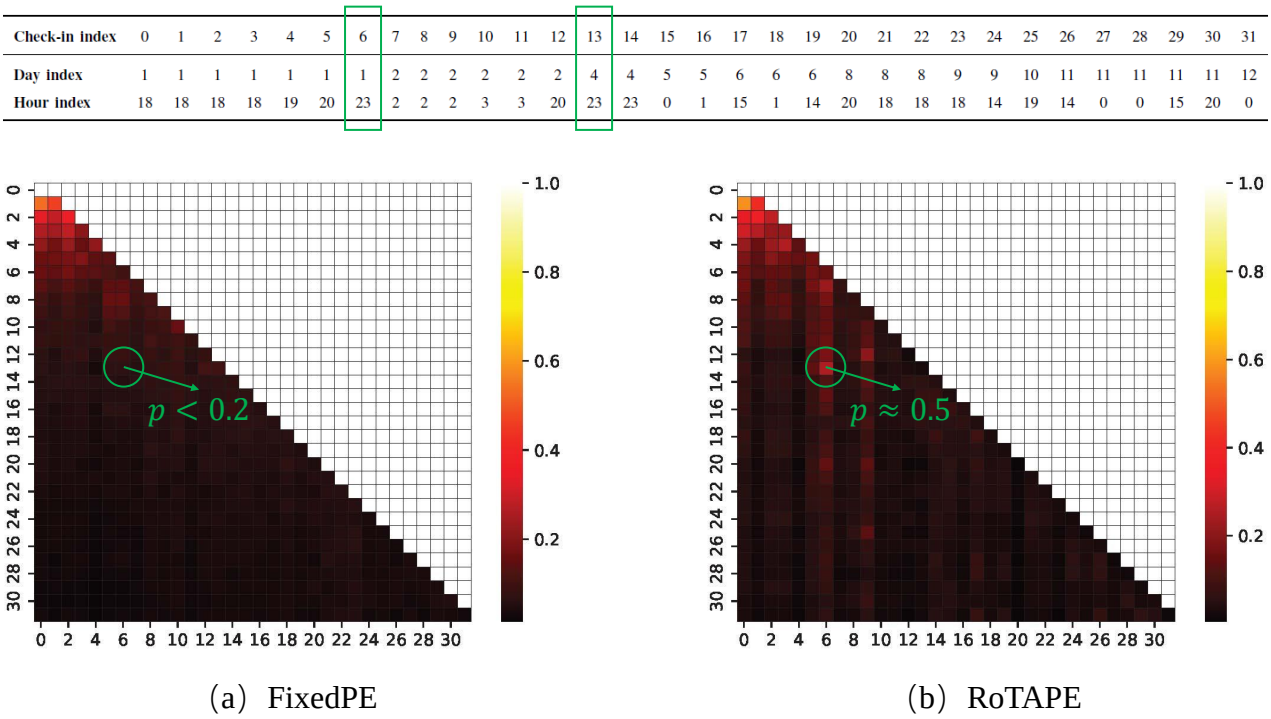


Fig. 4. Visualization of average weight in FixedPE and RoTAPE. The upper part lists the check-in index day- and hour-level time index. The attention score of the 13 th check-in towards the 6 th check-in is marked with green circle.

The results are summarized in Table 4. This table shows that RoTAPE, IAAB, and STR Memory are pivotal and critical in STiSAN$^+$ for predicting multi-location trajectory. Separately, removing RoTAPE and IAAB will decrease an average of 12.35% and 14.00% performance. Moreover, comparing Variant III with Original, removing STR Memory decreases 9.80% performance, demonstrating the effectiveness of STR Memory in forecasting the future spatial-temporal relations, which can be captured by the IAAB-based Decoder for prediction.

4.5.4 *Extensibility and Interpretability of RoTAPE (RQ3)*

We set two experiments under the scenario of Next Location Recommendation to answer this question from the angles of metric evaluation and principle.

Firstly, we choose a 2-layer self-attention network as the backbone to verify the compatibility of RoTAPE. Moreover, we compare RoTAPE with FixedPE [19], LearnedPE [33], RoPE [15], CTLE [13], CFPRec [12] and TAPE [26]. The experimental results on the dataset of Brightkite are reported in Table 5. Credited to the proper temporal encoding (hour- and weekday-level), CFPRec achieves comparable performances when $n = \{32, 128, 256, 512\}$. Note that our RoTAPE consistently shows superiority with historical sequences of various lengths n. It proves the effectiveness of simultaneously encoding multi-level temporal information.

Take a step further, towards the question "*Why RoTAPE ?*", we conduct the following visualization experiment. Specifically, we randomly choose a sequence of length 32 in Brightkite and compare the average attention weights of vanilla positional encoding [19] (denoted as FixedPE) and RoTAPE in Fig. 4. The upper part of Fig. 4 lists the day- and hour-level timestamps of the 32 check-ins. We take the 6 th and 13 th check-in for the detailed comparison, where the positional, day- and hour-level differences are 7, 3, and 0, separately. In FixedPE, the weight (marked with green circles in Fig. 4 (a)) is smaller than 0.2, which only considers the positional difference. In RoTAPE, the weight is around 0.5. It indicates that RoTAPE is sensitive to such special temporal differences and endows the higher weight, which might contain the periodical patterns.

4.5.5 *Extensibility and Interpretability of IAAB (RQ3)*

We also set two experiments in the Next Location recommendation task to answer this question from the view

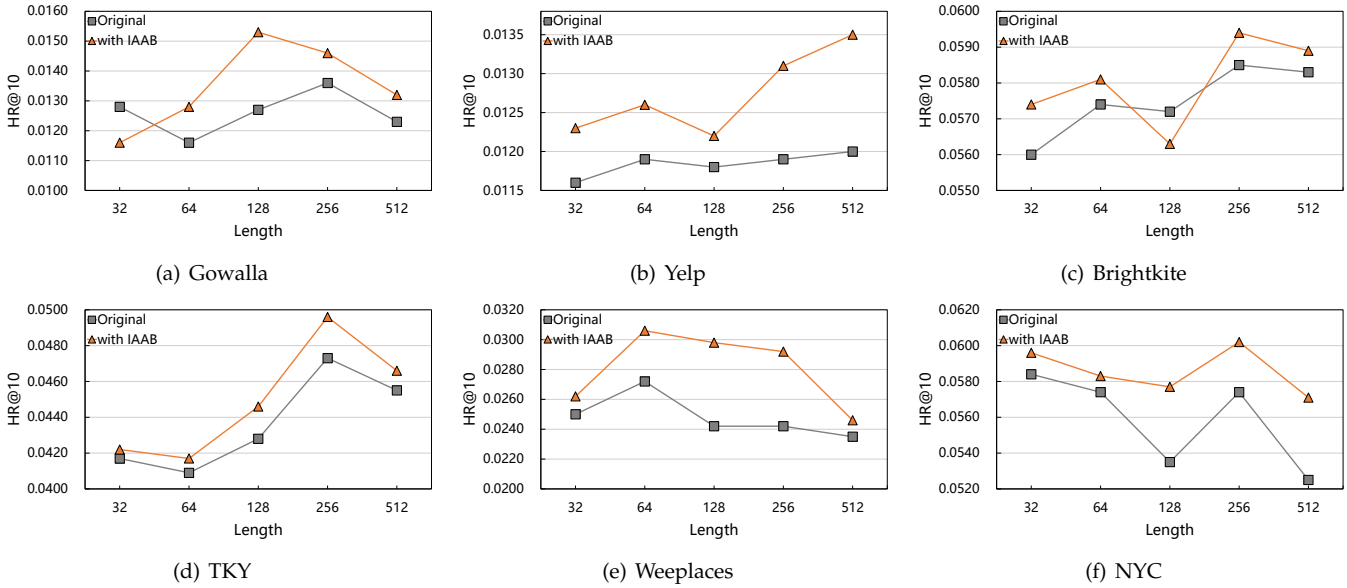


Fig. 5. Extensibility of Interval Aware Attention Block (evaluated on non-sampled metric).

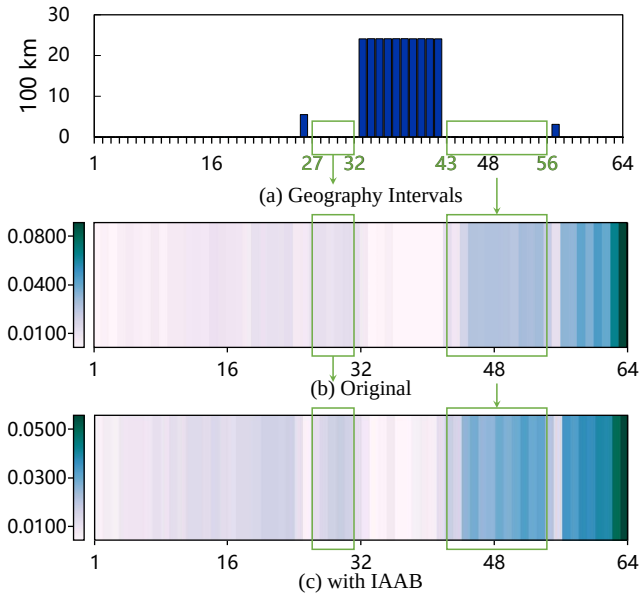


Fig. 6. Interpretability of Interval Aware Attention Block. The locations with strong spatial correlations are marked with the green box.

of metric evaluation and principle separately. Firstly, we replace the self-attention mechanism (denoted as Original) with our IAAB to explore the influence of various sequence lengths. As shown in Fig. 5, longer sequences improve the preference modeling. However, due to the insufficient attention of local locations, Original's performance decreases dramatically (e.g., 128 to 512 in Fig.5(a) and 64 to 128 in Fig. 5(b)). Our IAAB effectively relieves this issue and helps the model achieve superior recommendation accuracy. It proves that IAAB can be a lightweight alternative for SAN to consider spatial factors.

Moreover, we randomly pick a user in Weeplaces who has visited 64 locations. Fig. 6 (a) shows the geography intervals between historical and target locations, and the strong spatial correlated locations are marked with the green box (e.g., location $27 \sim 32$ and location $43 \sim 56$). Comparing the corresponding weights of Original and IAAB (as shown in Fig. 6 (b) and Fig. (c)), our IAAB can pay significant attention to these vital locations. The experimental results also demonstrate that our method can provide explainable recommendations.

TABLE 6
Run Time Comparison on the Dataset of Gowalla.

Dataset	Gowalla			
Method	SAN	w. Ro	w. IA	w. Ro & IA
Run Time	7.45 s	7.84 s	7.62 s	7.98 s

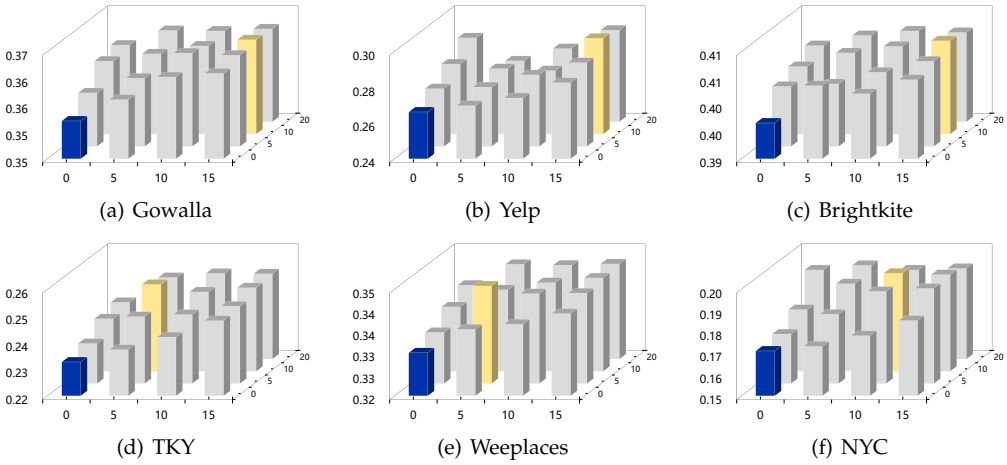
4.5.6 *Efficiency of RoTAPE and IAAB (RQ3)*

We choose a 2-layer self-attention network with vanilla positional encoding as baseline (denoted as SAN), and equip it with RoTAPE (Ro), IAAB (IA) to compare the average time cost of finishing 10 rounds recommendation. The results are reported in TABLE 6. According to the results, the incremental time cost is marginal.

4.5.7 *Hyper-parameters Analysis*

We explore STiSAN$^+$'s sensitivity with respect to the thresholds k_t and k_d, which control the maximum time interval and geography interval in the spatial-temporal relation matrix. We set $k_t = \{0, 5, 10, 20\}$ days. Correspondingly, we set $k_d = \{0, 5, 10, 15\}$ kilometers.

As the blue columns in Fig. 7, when setting both k_t and k_d to zero, the recommendation accuracy reaches the lowest on all datasets which actually disables the IAAB. For Gowalla, Yelp and Brightkite, the most suitable sets are $k_t = 10, k_d = 15$. The model achieves the best performance at $k_t = 5, k_d = 10$ on TKY, $k_t = 5, k_d = 5$ on Weeplaces, $k_t = 10, k_d = 10$ on NYC.



(a) Gowalla (b) Yelp (c) Brightkite

(d) TKY (e) Weeplaces (f) NYC

Fig. 7. Sensitivity w.r.t different hyper-parameter settings (NDCG@5). The lowest and highest scores are marked with blue and yellow separately.

5 Related Works

5.1 Attention Mechanism

Attention mechanism has been proved to be effective in various tasks ranging from computer vision and natural language processing to sequential recommender systems [20]. The core idea behind such a mechanism is impelling the model to attach more importance to the more relevant parts of the input. In the recommendation scenario, existing methods, such as [34], [35], [36], employ attention mechanisms to learn the importance of different items, features, or interactions for predicting user preferences on candidates. Recently, Transformer [19], a model solely-based on the attention mechanism, has shown superior performance in machine translation tasks, which used to be dominated by RNN/CNN-based approaches [20]. It proposed a novel self-attention mechanism to calculate the attention weights of different tokens dynamically and flexibly according to the input sequence. Inspired by the advanced performance, [20] first introduced the self-attention-based encoder in sequential item recommendation problems.

5.2 Positional Representation

Due to the symmetry property, the self-attention mechanism is non-sensitive to the order of different tokens in sequence. To capture the token's positional information, Transformer [19] encodes the positions with a fixed sinusoidal function. For large-scale and pre-trained models, [33] shows that learning a set of absolute positional embedding performs better than the fixed positional encoding. Differently, [15] focuses on learning the relative positional relations among tokens rather than directly learning positions. Some works also utilize hybrid representations that contain absolute and relative positional information [37].

5.3 Next Location Recommendation

Along with the information technology development and user-location interaction data accumulating, methods for sequential location recommendation have been evolving from Markov Chain [38], [39] and Matrix Factorization [40], [41] to Multi-Layer Perceptron (MLP) [42], [43], Recurrent Neural Network (RNN) [24], [44], [45] and Convolution Neural Network (CNN) [46], [47] over the past decades. SASRec [20] first introduced a self-attention network into the sequential recommendation problem and [25] extended to the bidirectional version. [11], [12], [13] integrates temporal information via a modified attention mechanism and learnable encoding functions. [6], [27] utilizes the tile map to encode geographical coordinates. The most recent STAN [14] exploits a novel spatial-temporal attention network, utilizes the learned spatial-temporal interval representations, and has achieved state-of-the-art performance.

6 Concluding Remarks

This paper proposes an end-to-end mobility trajectory prediction framework, namely STiSAN$^+$. It employs a Rotary Time Aware Position Encoder (RoTAPE) to encode the absolute time span and relative periodical pattern among locations in historical trajectory, stacks multi-layer Interval Aware Attention Block (IAAB) Encoder-Decoder architecture to introduce spatial-temporal relation matrix, and couples with Spatial-Temporal Relation Memory (STR Memory) to forecast future spatial-temporal relations. Besides the overall performance under in two scenarios, Next Location recommendation and Multi-location Trajectory Prediction, we extend RoTAPE and IAAB to the basic self-attention network and explain their principles of validity through visualization experiments.

In the future, we plan to exploit a pre-trained Spatial-Temporal Relation Memory, which can be effectively extended to any self-attention-based sequential location recommender or trajectory predictor.

Acknowledgment

This work was supported by the General Program of National Natural Science Foundation of China 62072209, the National Natural Science Foundation of Youth Fund under Grant 62002123, the Key Project of Science and Technology Development Plan of Jilin Province Grant 20210201082GX, Jilin Education Science Foundation JJKH20221010KJ, Jilin Science and Technology Research Project 20230101067JC.

REFERENCES

[1] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations," *ACM Trans. Inf. Syst.*, vol. 39, no. 1, pp. 10:1–10:42, 2020.

[2] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, F. Rossi, Ed. IJCAI/AAAI, 2013, pp. 2605–2611.

[3] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, Eds. AAAI Press, 2015, pp. 2069–2075.

[4] R. Li, Y. Shen, and Y. Zhu, "Next point-of-interest recommendation with temporal and multi-level context attention," in *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 2018, pp. 1110–1115.

[5] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, D. Schuurmans and M. P. Wellman, Eds. AAAI Press, 2016, pp. 194–200.

[6] D. Lian, Y. Wu, Y. Ge, X. Xie, and E. Chen, "Geography-aware sequential location recommendation," in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020, pp. 2009–2019.

[7] Y. Liu, J. Wang, Y. Zhang, L. Cheng, W. Wang, Z. Wang, W. Xu, and Z. Li, "Vernier: Accurate and fast acoustic motion tracking using mobile devices," *IEEE Trans. Mob. Comput.*, vol. 20, no. 2, pp. 754–764, 2021.

[8] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, and R. Ghani, Eds. ACM, 2014, pp. 831–840.

[9] M. Ye, P. Yin, W. Lee, and D. L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, W. Ma, J. Nie, R. Baeza-Yates, T. Chua, and W. B. Croft, Eds. ACM, 2011, pp. 325–334.

[10] Y. Zhang and X. Zhang, "Price learning-based incentive mechanism for mobile crowd sensing," *ACM Trans. Sens. Networks*, vol. 17, no. 2, pp. 17:1–17:24, 2021.

[11] J. Li, Y. Wang, and J. J. McAuley, "Time interval aware self-attention for sequential recommendation," in *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, J. Caverlee, X. B. Hu, M. Lalmas, and W. Wang, Eds. ACM, 2020, pp. 322–330.

[12] L. Zhang, Z. Sun, Z. Wu, J. Zhang, Y. S. Ong, and X. Qu, "Next point-of-interest recommendation with inferring multi-step future preferences," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, L. D. Raedt, Ed. ijcai.org, 2022, pp. 3751–3757.

[13] Y. Lin, H. Wan, S. Guo, and Y. Lin, "Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 4241–4248.

[14] Y. Luo, Q. Liu, and Z. Liu, "STAN: spatio-temporal attention network for next location recommendation," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds. ACM / IW3C2, 2021, pp. 2177–2185.

[15] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *CoRR*, vol. abs/2104.09864, 2021. [Online]. Available: https://arxiv.org/abs/2104.09864

[16] D. Zhou, Y. Shi, B. Kang, W. Yu, Z. Jiang, L. Yuan, X. Jin, Q. Hou, and J. Feng, "Refiner: Refining self-attention for vision transformers," *CoRR*, vol. abs/2106.03714, 2021.

[17] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, and T. Zhang, "Modeling localness for self-attention networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Association for Computational Linguistics, 2018, pp. 4449–4458.

[18] F. Yu, L. Cui, W. Guo, X. Lu, Q. Li, and H. Lu, "A category-aware deep model for successive POI recommendation on sparse check-in data," in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Y. Huang, I. King, T. Liu, and M. van Steen, Eds. ACM / IW3C2, 2020, pp. 1264–1274.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008.

[20] W. Kang and J. J. McAuley, "Self-attentive sequential recommendation," in *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 2018, pp. 197–206.

[21] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016.

[22] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Y. Chang, C. Zhai, Y. Liu, and Y. Maarek, Eds. ACM, 2018, pp. 565–573.

[23] P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou, "Where to go next: A spatio-temporal gated network for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2512–2524, 2022.

[24] D. Yang, B. Fankhauser, P. Rosso, and P. Cudré-Mauroux, "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 2020*, C. Bessiere, Ed. ijcai.org, 2020, pp. 2184–2190.

[25] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds. ACM, 2019, pp. 1441–1450.

[26] E. Wang, Y. Jiang, Y. Xu, L. Wang, and Y. Yang, "Spatial-temporal interval aware sequential POI recommendation," in *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 2022, pp. 2086–2098.

[27] S. Jiang, W. He, L. Cui, Y. Xu, and L. Liu, "Modeling long- and short-term user preferences via self-supervised learning for next poi recommendation," vol. 17, no. 9, 2023.

[28] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola, "COFI RANK - maximum margin matrix factorization for collaborative ranking," in *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2007, pp. 1593–1600.

[29] Q. Gao, F. Zhou, K. Zhang, F. Zhang, and G. Trajcevski, "Adversarial human trajectory learning for trip recommendation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 4, pp. 1764–1776, 2023.

[30] D. Chen, C. S. Ong, and L. Xie, "Learning points and routes to recommend trajectories," in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, S. Mukhopadhyay, C. Zhai, E. Bertino, F. Crestani, J. Mostafa, J. Tang, L. Si, X. Zhou, Y. Chang, Y. Li, and P. Sondhi, Eds. ACM, 2016, pp. 2227–2232.

[31] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency," *Knowl. Inf. Syst.*, vol. 54, no. 2, pp. 375–406, 2018.

[32] G. Yuan, F. Yuan, Y. Li, B. Kong, S. Li, L. Chen, M. Yang, C. Yu, B. Hu, Z. Li, Y. Xu, and X. Qie, "Tenrec: A large-scale multipurpose benchmark dataset for recommender systems," in *NeurIPS*, 2022.

[33] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.

[34] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, N. Kando, T. Sakai, H. Joho, H. Li, A. P. de Vries, and R. W. White, Eds. ACM, 2017, pp. 335–344.

[35] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, C. Sierra, Ed. ijcai.org, 2017, pp. 3119–3125.

[36] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attention-based transactional context embedding for next-item recommendation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 2532–2539.

[37] G. Ke, D. He, and T. Liu, "Rethinking positional encoding in language pre-training," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[38] R. He, C. Fang, Z. Wang, and J. J. McAuley, "Vista: A visually, socially, and temporally-aware model for artistic recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, S. Sen, W. Geyer, J. Freyne, and P. Castells, Eds. ACM, 2016, pp. 309–316.

[39] R. He, W. Kang, and J. J. McAuley, "Translation-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*, P. Cremonesi, F. Ricci, S. Berkovsky, and A. Tuzhilin, Eds. ACM, 2017, pp. 161–169.

[40] S. Kabbur, X. Ning, and G. Karypis, "FISM: factored item similarity models for top-n recommender systems," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, and R. Uthurusamy, Eds. ACM, 2013, pp. 659–667.

[41] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki, Eds. ACM, 2009, pp. 447–456.

[42] S. Wan, Y. Lan, P. Wang, J. Guo, J. Xu, and X. Cheng, "Next basket recommendation with neural networks," in *Poster Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16, 2015*, ser. CEUR Workshop Proceedings, P. Castells, Ed., vol. 1441. CEUR-WS.org, 2015.

[43] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, R. Baeza-Yates, M. Lalmas, A. Moffat, and B. A. Ribeiro-Neto, Eds. ACM, 2015, pp. 403–412.

[44] D. Le, H. W. Lauw, and Y. Fang, "Modeling contemporaneous basket sequences with twin networks for next-item recommendation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, J. Lang, Ed. ijcai.org, 2018, pp. 3414–3420.

[45] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen, "Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Y. Guo and F. Farooq, Eds. ACM, 2018, pp. 1734–1743.

[46] T. X. Tuan and T. M. Phuong, "3d convolutional networks for session-based recommendation with content features," in *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*, P. Cremonesi, F. Ricci, S. Berkovsky, and A. Tuzhilin, Eds. ACM, 2017, pp. 138–146.

[47] F. Yuan, X. He, H. Jiang, G. Guo, J. Xiong, Z. Xu, and Y. Xiong, "Future data helps training: Modeling future contexts for session-based recommendation," in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Y. Huang, I. King, T. Liu, and M. van Steen, Eds. ACM / IW3C2, 2020, pp. 303–313.

Yiheng Jiang received his B.E. degree in the College of Plant Science from Jilin University, Changchun, in 2018 and his M.E. degree in computer science and technology from Jilin University, Changchun, in 2022. He is currently a Ph.D. student in computer science and technology at Jilin University, Changchun. His research interests include applications of data mining, recommender systems, and mobile computing. He has published some research results at conferences such as INFOCOM and ICDE.

Yongjian Yang received his B.E. degree in automatization from Jilin University of Technology, Changchun, Jilin, China, in 1983; and M.E. degree in Computer Communication from Beijing University of Post and Telecommunications, Beijing, China, in 1991; and his Ph.D. in Software and theory of Computer from Jilin University, Changchun, Jilin, China, in 2005. He is currently a professor and a PhD supervisor at Jilin University, Director of Key lab under the Ministry of Information Industry, Standing Director of Communication Academy, member of the Computer Science Academy of Jilin Province. His research interests include: Theory and software technology of network intelligence management; Key technology research of wireless mobile communication and services.

Yuanbo Xu received his B.E. degree in computer science and technology from Jilin University, Changchun, in 2012, his M.E. degree in computer science and technology from Jilin University, Changchun, in 2015, and his Ph.D. in computer science and technology from Jilin University, Changchun, in 2019. He is currently a Postdoc in the Department of Artificial Intelligence at Jilin University, Changchun. He is also a visiting scholar in the Management Science and Information Systems Department at Rutgers, the State University of New Jersey. His research interests include applications of data mining, recommender systems, and mobile computing. He has published some research results on journals such as TKDE, TMM, TNNLS, TKDD and conferences as INFOCOM, ICDE, and ICDM.

En Wang received his B.E. degree in software engineering from Jilin University, Changchun, in 2011, his M.E. degree in computer science and technology from Jilin University, Changchun, in 2013, and his Ph.D. in computer science and technology from Jilin University, Changchun, in 2016. He is currently an Associate Professor in the Department of Computer Science and Technology at Jilin University, Changchun. He is also a visiting scholar in the Department of Computer and Information Sciences at Temple University in Philadelphia. His current research focuses on the efficient utilization of network resources, scheduling, and drop strategy in terms of buffer-management, energy-efficient communication between human-carried devices, and mobile crowd-sensing.