Spatial-Temporal Interval Aware Sequential POI Recommendation

En Wang[†,§], Yiheng Jiang[†,§], Yuanbo Xu[†,§,*], Liang Wang[‡], Yongjian Yang[†,§]
[†]*Department of Computer Science and Technology, Jilin University, China*
[§]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China*
[‡]*School of Computer Science, Northwestern Polytechnical University, China*
[†,§]{wangen, yuanbox, yyj}@jlu.edu.cn; [†,§]yhjiang19@mails.jlu.edu.cn; [‡]liangwang@nwpu.edu.cn

2022 IEEE 38th International Conference on Data Engineering (ICDE) | 978-1-6654-0883-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDE53745.2022.00202

***Abstract*—The past flourishing years of sequential point-of-interest (POI) recommendation began with the introduction of Self-Attention Network (SAN), which quickly superseded CNN or RNN as the state-of-the-art backbone. To realize the fine-grained users' behavior patterns modeling, recent works utilize modified attention mechanisms or neural network layers to process spatial-temporal factors. However, due to the significant increase on either model's parameter scale or computational burden, we argue that these methods can be further improved. In this paper, we exploit two lightweight approaches, Time Aware Position Encoder (TAPE) and Interval Aware Attention Block (IAAB), to impel SAN by considering the spatial-temporal intervals among POIs separately, where requiring neither extra parameters nor high computational cost. On the one hand, TAPE, adjusting the positions in sequences based on the timestamps dynamically and generating positional representations with sinusoidal transformation, can enhance sequence representations to reflect both the absolute order and relative temporal proximity among all POIs. On the other hand, IAAB, point-wise adding the scaled spatial-temporal intervals to the attention map, can promote the attention mechanism attaching importance to the spatial relation among all POIs under the constraints of time conditions and providing more explainable recommendation. We integrate these two modules into SAN and propose a Spatial-Temporal Interval-Aware sequential POI recommender, namely STiSAN, as an end-to-end deployment. Experimental results based on three public LBSN datasets and one real-world city transportation dataset demonstrate STiSAN's superior performance (average 13.01% improvement against the strongest baseline). Moreover, we validate the extensibility and interpretability of TAPE and IAAB through metric evaluation and visualization separately.**

***Index Terms*—sequential POI recommendation, positional encoding, attention mechanism**

I. Introduction

Point-of-interest (POI) recommendation, as the sharp tool of conjecturing users' preferences and providing pleasant suggestions, is the hot-spot for both industry and academia (e.g. Location-Based Social Networks, LBSNs). According to the assumption of users' preferences dynamics, it can be classified into two streams [1]: ***conventional POI recommendation*** is inclined to make predictions from the view of static, where defaulting the preference is stable, while ***sequential POI recommendation*** holds the opinion that a user's historical check-ins should be considered, which might bring changes to the user's current or future preference.

* Yuanbo Xu is the corresponding author.

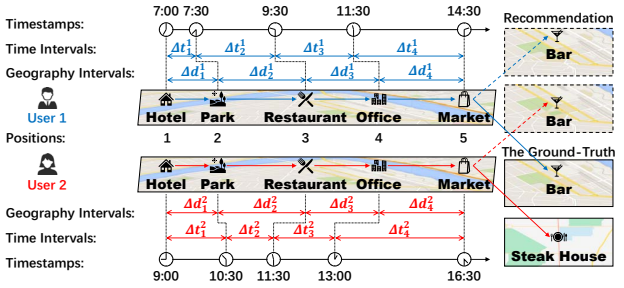


Fig. 1. The influence of spatial-temporal factors on POI check-in activities. The same historical POI sequence with different time intervals may lead user 1 and user 2 to visit different POIs.

The ideal behind sequential POI recommendation is always mining the dependency from historical sequences [2], while its paradigm has been shifting over the past decade along with the development of information technology. Early works like [3]–[10] are conducted on the basis of Markov Chain (MC) and Matrix Factorization (MF). For instance, FPMC [11] carries out a linear combination of MC and MF to model the personalized transition between POIs. Credited to the great improvements of computational power and data quality, deep learning has achieved satisfactory performance in sequential POI recommendation, and derived large amount of Neural Network (MLP/RNN/CNN)-based models. For example, HRM [12] utilizes multi-layer feed-forward network to capture the complex and nonlinear relationship among POIs, GRU4Rec [13] employs a modified gated recurrent unit to learn the dependency inside sequence and Caser [14] adopts convolution kernels to model the local dependent relationship across POIs. The flourishing of sequential POI recommendation began with the introduction of Self-Attention Network (SAN) [15] for its remarkable potential in dealing with sequential issues. SASRec [16] is a milestone work in this direction, which quickly superseded CNN or RNN as the state-of-the-art sequential recommendation backbone [17], [18].

The ***spatial*** and ***temporal information*** are two pivotal and complementary factors in sequential POI recommendation [19]–[22]. On the one hand, spatial information (e.g. geography interval Δd) can describe the physical proximity between

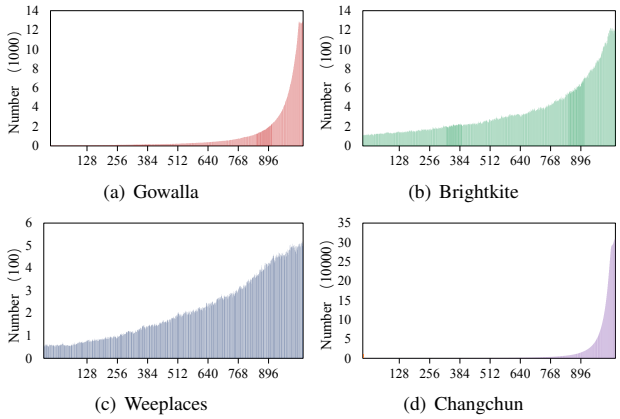
2375-026X/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDE53745.2022.00202

POIs [23], especially when individual mobility history [27] usually exhibits the spatial clustering phenomenon [24]–[26]. On the other hand, temporal information (e.g. time interval Δt) can reflect the relative temporal proximity among POIs, which contributing to more personalized individual preference modeling [28]. As the example in Fig. 1, user 1 and user 2 shared the same historical POI sequence "$Hotel \rightarrow Park \rightarrow Restaurant \rightarrow Office \rightarrow Market$", and then they visited "Bar", "$SteakHouse$" separately. Concentrated on their time intervals Δt^1 and Δt^2, we can find the distinctiveness of their historical sequences, which indicating their different behavior patterns (e.g. user 1's 2nd POI is closer to 1st POI rather than the third one from the view of time, while user 2 is on the contrary). If a sequential recommender only considers POIs and spatial information, the representations for these two users' sequences would be highly similar, and the recommendations might be biased from the ground-truth (as the right half of Fig. 1). Intuitively, modeling temporal factors would be conducive for distinguishing such same sequences and realizing finer grained sequence representations.

Various approaches have been attempted to explore the influence brought by temporal factors. Embedding the timestamps into sequence representations is the most direct one, but its performance is not always as expected. For the reason that POIs have no attributes about time which leads to the mismatching between historical sequence space and candidate POI space [23]. Recent works try to integrate temporal information more reasonably by modifying the network architecture, e.g. conducting self-attention on temporal information to capture the dependency [28] and introducing temporal relation matrix to exploit temporal effect [29]. However, mapping temporal information into high dimension space, as the premise of all these advanced methods, brings a significant increase on either parameter scale or computational cost, especially when the computational complexity of backbone SAN is quadratically correlated with the sequence length.

Towards this issue, we propose a Time Aware Position Encoder (TAPE) in this paper, requiring neither extra parameters nor high computational burden (i.e., ***lightweight***), to consider the time intervals among POIs. It is sparked by the positional encoding [15], where defaulting the POIs' positions in sequence are "$1 \rightarrow 2 \rightarrow 3$" (as shown in Fig. 1). Our TAPE adjusts the difference between positions dynamically based on the corresponding timestamps (e.g. "$1 \rightarrow 2.2 \rightarrow 4.7$" for user 1 and "$1 \rightarrow 2.1 \rightarrow 3.5$" for user 2), and then generates positional representations via sinusoidal transformation. After processed with TAPE, the representations are enhanced with the ability of reflecting the relative temporal proximity between POIs, which can be effectively captured by SAN to distinguish similar sequences and provide fine-grained preference modeling.

Take a step further, due to the global attention design of SAN, the weighted averaging inhibits the relation among local POIs [30], [31]. In sequential POI recommendation, it can be described as SAN is expert in learning about dependency from the whole sequence while weakening the spatial correlation among neighboring POIs. POIs with strong spatial correlations are critical for recommendation because the smaller geography distance usually leads to the higher visiting probability [32]. To verify this, we define the maximum geography interval as 10km [32] and visualize the distribution of historical POIs sharing strong spatial correlation with the corresponding target POI[1]. From Fig. 2, we can find that these POIs distribute not only in users' short-term check-ins (e.g. the last 128 POIs) but also among users' earlier visits (e.g. positions ranging from 640 to 896 and from 768 to 896 in Gowalla and Changchun, the whole sequence in Brightkite and Weeplaces). Unfortunately, the aforementioned drawback limits SAN to consider the spatial relation among all these local POIs and decreases the recommendation accuracy especially when processing longer sequences.



(a) Gowalla (b) Brightkite (c) Weeplaces (d) Changchun

Fig. 2. The distribution of strong spatial correlated POIs among users' historical sequences in four datasets. The horizontal axis denotes positions, and the vertical axis counts the number of POIs which are physically proximity to the corresponding target POI (less than 10 km).

To avoid this issue, we are inspired by [29], [31] and propose another lightweight module, Interval Aware Attention Block (IAAB), where alternating an interval-aware attention layer and a two-layer feed-forward network. Specifically, we construct a spatial-temporal relation matrix based on the geography and time intervals (Δd, Δt) among historical POIs to reflect the spatial correlations under the constraints of temporal conditions. Then, the attention layer 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 among all POIs in the whole sequence, which relieving the insufficient local attention issue. Moreover, the explicit utilization of spatial-temporal information in IAAB improves the interpretability.

We integrate the above two lightweight approaches into the Self-Attention Network and propose a Spatial-Temporal interval aware sequential POI recommendation framework,

[1]The target POI denotes a user's last visited POI. The statistic is carried on three public LBSN datasets and one real-world city transportation dataset: Gowalla, Brightkite, Weeplaces and Changchun.

namely STiSAN, as an end-to-end deployment. Our contributions can be summarized as:

- We exploit two lightweight approaches Time Aware Position Encoder (TAPE) and Interval Aware Attention Block (IAAB) to consider the relative spatial-temporal proximity among POIs separately, where requiring neither extra parameter nor significant computational burden.
- We integrate TAPE, IAAB into SAN and propose a Spatial-Temporal interval aware sequential POI recommendation framework (STiSAN), as an end-to-end deployment, to realize finer grained preference modeling and provide more explainable recommendation.
- We conduct enormous experiments to evaluate our method on three public LBSN datasets and one real-world city transportation dataset. The experimental results show that STiSAN gains 13.01% improvement on average against several state-of-the-art baselines. Apart from validating the effectiveness of TAPE and IAAB under our framework, we further carry on metric evaluations and visualizations to verify their extensibility and interpretability separately.

The rest of this paper is organized as follows. We first give several basic concept definitions and formally state the Top-K sequential POI recommendation problem in Section II. Then, we elaborate on the details of our proposed approaches in Section III. Next, we analyze the experimental results in Section IV. Finally, we review related works in Section V and conclude this paper in Section VI.

II. Preliminary

In this section, we start from defining several basic concepts and then providing a formal statement for Top-K sequential POI recommendation problem. All important notations and corresponding descriptions are listed in Table I.

A. Basic Definition

1) Definition 1 (Check-in): A check-in is denoted as a quad-tuple $c^u = \langle u, p, g, t \rangle$, which indicates that user u visited POI p at time t and the POI's location (GPS coordinate) is g.

2) Definition 2 (POI Sequence): A user u's POI sequence $S^u = c_1^u \to c_2^u \to \cdots \to c_{|S^u|}^u$ records his/her $|S^u|$ check-ins by chronological order. Each POI's position in sequence is the subscript of the corresponding check-in, e.g. $1, 2, \cdots, |S^u|$.

3) Definition 3 (Relative Spatial-Temporal Proximity): The relative spatial-temporal proximity between the i-th and j-th POI in sequence is denoted as r_{ij} which consists of the geography interval Δd_{ij} and time interval Δt_{ij}.

B. Top-K Sequential POI Recommendation

Top-K sequential POI recommendation is carried out according to the following process: Given a user u's POI sequence $S^u = c_1^u \to c_2^u \to \cdots \to c_{|S^u|}^u$, mining the dependency from visited POIs, modeling the user's preference based on the dependency and spatial-temporal information and recommending a list of K ranked POIs. It can be described as the following equation,

$$TopK^u = \text{Rec}\left(S^u\right), \tag{1}$$

where $TopK^u$ is the Top-K recommendation list contains K POI that the user u might visit in the future. $\text{Rec}\left(\cdot\right)$ is an abstract function representation symbol to signify any sequential recommender which takes as input the user's historical sequence and provides a list of Top-K POIs.

TABLE I
Important Notations and Corresponding Meanings

Notations	Descriptions
S^u	historical POI sequence for user u:
S	the set of all training sequences
C	the candidate POI set
p, g, t	POI, corresponding exact location and timestamp
pos	POI's position in sequence
n	the maximum historical POI sequence length
d, d_h	dimensions for latent representations
$\Delta t, \Delta d$	time and geography interval
k_t, k_d	thresholds for maximum time and geography interval
r	relative spatial-temporal proximity
μ, σ	the mean and standard-deviation for input vector
α, β, ϵ	the parameters in layer normalization
N	number of stacked Interval Aware Attention Blocks
L	number of negative samples for model training
T	the term to control negative samples distribution
w	the weight for negative sample
$y_{i,j}$	the preference score over POI j at step i
$\boldsymbol{R}$	spatial-temporal relation matrix
$\boldsymbol{E}$	representation matrix for input sequence
$\boldsymbol{P}$	position representation matrix
$\boldsymbol{W}_{\{Q,K,V\}}$	converting matrices for query, key, value
$\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$	query, key, value matrix
$\boldsymbol{A}$	the attentive representations for input sequence
$\boldsymbol{F}$	the output of Interval Aware Attention Block
$\boldsymbol{S}$	representation matrix for user preference
$\boldsymbol{C}$	representation matrix for candidate set

III. Methodology

A. Framework Overview

The architecture of the proposed spatial-temporal interval aware sequential POI recommendation framework STiSAN is shown in Fig. 3. It follows the classic encoder-decoder structure, where capturing the sequential dependency and improving users' preference representations separately. Two proposed lightweight modules Time Aware Position Encoder and Interval Aware Attention Block respectively replace the vanilla positional encoding and self-attention mechanism [15]. One for enhancing POI sequence representations to reflect relative temporal proximity, and the other pays focus attention to the spatial correlation among local POIs.

During the training process, STiSAN takes the POI sequence excluded the last check-in $c_1^u \to c_2^u \to \cdots \to c_{|S^u|-1}^u$ as source, and the sequence excluded the first $c_2^u \to c_3^u \to \cdots \to c_{|S^u|}^u$ as target. At each step i, it aims at predicting the $i+1$-th visited POI. During the recommendation stage,

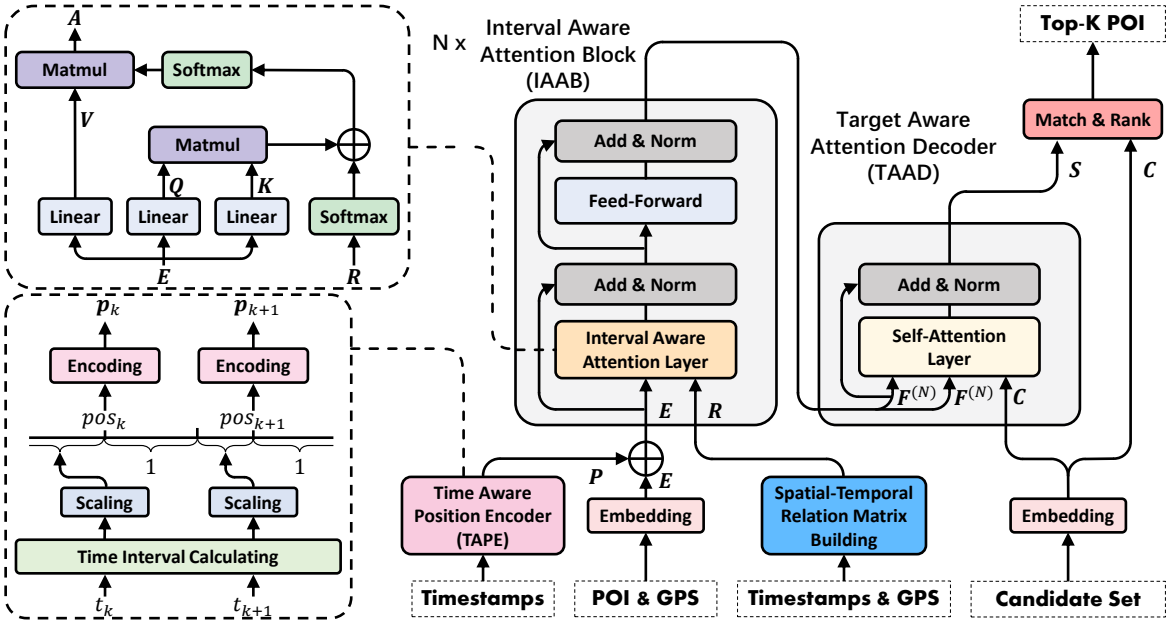


Fig. 3. The framework of STiSAN. The details of two significant components TAPE and IAAB are revealed in dashed boxes.

STiSAN takes as input the whole POI sequence, and suggests a list of ranked K POIs. Each component of STiSAN will be elaborated on in the following sections.

B. Embedding Module

The Embedding module takes as input a user u's POI sequence S^u and outputs the sequence representation $\boldsymbol{E} \in \mathbb{R}^{n \times d}$, where n is the maximum sequence length and d is the dimension. Each check-in's representation vector is the concatenation of POI embedding[2] and GPS coordinate encoding[3].

Considering that different users' sequences are inconsistent in length, we split longer ones into several non-overlapping sub-sequences of length n. For shorter sequences, we repeatedly add a "padding" check-in in the head of the sequence until the lengths grow to n. We encode the padding check-ins with zero vectors to avoid influencing the gradient updating.

C. Time Aware Position Encoder (TAPE)

Time Aware Position Encoder (TAPE) is modified on the basis of vanilla positional encoding [15]. The core idea behind TAPE is dynamically adjusting the differences between positions according to the time intervals in POI sequence, and further reflecting the relative temporal proximity. To be specific, TAPE takes as input the timestamps t in sequence and calculates the $k+1$-th POI position as the following equation:

$$pos_{k+1} = pos_k + \frac{\Delta t_{k,k+1}}{\overline{\Delta t}} + 1, \tag{2}$$

[2]We embed POI p with `torch.nn.embedding()`

[3]The GPS coordinate g is processed by the geography encoder proposed in [23]:https://github.com/libertyeagle/GeoSAN

where pos_k is the previous POI's position and $\Delta t_{k,k+1} = t_{k+1} - t_k$ is the time interval. Considering that timestamps distribute variously across different users, we normalize the interval by the sequence average time interval $\overline{\Delta t} = \frac{1}{n-1}\sum_{k=1}^{n-1} \Delta t_{k,k+1}$. We also add an extra 1 to make model distinguish POIs sharing extremely small time intervals. Recall the user 1 in Fig. 1, the positions are now transformed from "$1 \to 2 \to 3 \to 4 \to 5$" to "$1 \to 2.2 \to 4.3 \to 6.4 \to 9$", where the differences between positions are in line with the temporal correlations. Then, TAPE encodes the positions into d dimension space with the following sinusoidal function [15]:

$$\begin{aligned} \mathrm{PE}(pos, 2i) &= \sin(pos/10000^{2i/d}), \\ \mathrm{PE}(pos, 2i+1) &= \cos(pos/10000^{2i+1/d}), \end{aligned} \tag{3}$$

where $i = 1, 2, \cdots, d/2$. The representation matrix for all positions in sequence is denoted as $\boldsymbol{P} \in \mathbb{R}^{n \times d}$, and we inject it into sequence representation by $\boldsymbol{E} = \boldsymbol{E} + \boldsymbol{P}$.

Algorithm 1 PyTorch-like Pseudo-code of TAPE

```
class Time_Aware_Position_Encoder(nn.Module):
    def __init__(self, d):
        super(Time_Aware_Position_Encoder, self).__init__()
        self.div_term = torch.exp(torch.arange(d,2)*-(math.log(10000.0)/d))
    def forward(self, x, t):
        # x shape: (n, d)
        # t shape: (n)
        pre_t = torch.clone(t)
        pre_t[1:] = pre_t[:-1]
        delta_t = t - pre_t
        delta_t /= torch.sum(delta_t)/(x.size(0) - 1)
        pos = torch.zeros_like(t)
        pos[0] = 1.0
        for k in range(1, x.size(1)):
            pos[k] = pos[k - 1] + delta_t[k] + 1
        tape = torch.zeros_like(x)
        tape[:, 0::2] = torch.sin(pos * self.div_term)
        tape[:, 1::2] = torch.cos(pos * self.div_term)
        return x + tape
```

Notably, our TAPE, as a function, can reflect the temporal information through carefully designed positions without additional parameters. The PyTorch-like pseudo-code of TAPE is revealed is Algorithm 1[4]. Compared to the vanilla positional encoding, TAPE only adds $O(n)$ computational complexity which can be negligible to the subsequent attention operations.

D. Spatial-Temporal Relation Matrix Building

Before devoting into the details of attention mechanism, we first construct a spatial-temporal relation matrix $\boldsymbol{R} \in \mathbb{R}^{n \times n}$, as an inductive bias to reflect the relative spatial-temporal proximity. As follows,

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & 0 & \cdots & 0 \\ r_{21} & r_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix},$$

where we set $\boldsymbol{R}$ as the shape of lower triangle for preventing information leakage [16], [23] that the model can only attend to the previous i POIs when predicting the $i+1$-th POI at each step i. The spatial-temporal relation between the i-th and j-th POI is denoted as r_{ij}. To achieve r_{ij}, we define $\hat{r}_{ij} = \Delta t_{ij} + \Delta d_{ij}$ consists of the corresponding time and geography intervals. First, we consider that the precise intervals are not useful beyond a certain threshold [28] and then clip the intervals as formulated in (4) by maximum time and geography interval thresholds k_t, k_d, respectively:

$$\begin{aligned} \Delta t_{ij} &= \min\left(k_t, \mid t_i - t_j \mid\right), \\ \Delta d_{ij} &= \min\left(k_d, \text{Haversine}\left(g_i, g_j\right)\right), \end{aligned} \tag{4}$$

where $\text{Haversine}\left(\cdot\right)$ calculates the physical distance between two GPS coordinates. Second, 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 $\hat{r}_{ij}$.

E. Interval Aware Attention Block (IAAB)

To impel the model attaching important the spatial information among local POIs and providing more explainable recommendation, we introduce the spatial-temporal relation matrix $\boldsymbol{R}$ and design an Interval Aware Attention Block (IAAB). As shown in Fig. 3, IAAB alternates an interval aware attention layer and a feed-forward network along with the residual connection and layer normalization.

1) Interval Aware Attention Layer: The attention layer takes sequence representation $\boldsymbol{E}$ and relation matrix $\boldsymbol{R}$ as input. Firstly, it converts sequence representation $\boldsymbol{E}$ into query, key, value matrices through three distinct matrices $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V$,

$$\boldsymbol{Q} = \boldsymbol{E}\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{E}\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{E}\boldsymbol{W}_V, \tag{5}$$

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, the layer explicitly combines the attention map (i.e., sequential dependency) with the relation matrix by point-wise addition,

$$\boldsymbol{A} = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}} + \boldsymbol{R})\boldsymbol{V}, \tag{6}$$

[4]For the sake of simplicity, we remove the dimension operations like `unsqueeze.()` or `squeeze.()`.

Algorithm 2 PyTorch-like Pseudo-code of IAAB

```
class Interval_Aware_Attention_Layer(nn.Module):
    def __init__(self, drop_rate):
        super(Interval_Aware_Attention_Layer, self).__init__()
        self.dropout = nn.Dropout(drop_rate)
    def forward(query, key, value, r_mat, mask):
        # query, key, value shape: (n, d)
        # r_mat shape: (n, n)
        # mask shape: (n, n),
        # where upper triangle elements are zero
        r_mat = F.softmax(r_mat, dim=-1)
        attn_scores = torch.matmul(x, x.transpose(-2, -1))
        attn_scores /= math.sqrt(query.size(-1))
        probs = r_mat + attn_scores
        probs = F.softmax(probs.masked_fill(mask == 0, -1e9), dim=-1)
        probs = self.dropout(probs)
    return torch.matmul(probs, value)
```

where $\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d} \in \mathbb{R}^{n \times n}$ denotes the attention map and $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ is the attentive results. Note that we scale $\boldsymbol{R}$ with Softmax before the addition for normalization as shown in Fig. 3. Moreover, the aforementioned information leakage issue is also needed, and we achieve this point via setting the upper triangle of attention map with "$-\infty$" [23], [29].

In this way, our IAAB utilizes the spatial-temporal relation to provide the attention map with positive revisions, which strengthening model's ability of considering the relative spatial proximity among local POIs. Rather than embedding the relation into high dimension space, the explicit combination improves models' interpretability. The pseudo-code of the interval aware attention layer is shown in Algorithm 2. Compared to the vanilla self-attention mechanism [15], ours requires neither extra parameters nor significant computational burden and only increases nd FLOPs.

2) 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 [15]. It consists of two distinct linear layers and activation function ReLU. As follows,

$$\boldsymbol{F} = \max\left(0, \boldsymbol{A}\boldsymbol{W}_1 + b_1\right)\boldsymbol{W}_2 + b_2, \tag{7}$$

where $\boldsymbol{F} \in \mathbb{R}^{n \times d}$, $\boldsymbol{W}_1 \in \mathbb{R}^{d \times d_h}$, $\boldsymbol{W}_2 \in \mathbb{R}^{d_h \times d}$, s.t. $d_h > d$ and $b_1, b_2 \in \mathbb{R}^{1 \times d}$ are the learned bias terms.

3) Residual Connection and Layer Normalization: Recent works [15] have proved that multi-layer neural networks can help model capture the hierarchical features of the input. However, as the network goes deeper, the accuracy tends to be saturated and then quickly degrades. As reported in [33], the issue is caused by accumulated training errors of more layers rather than over-fitting.

Thus, we stack $N = 4$ Interval-Aware Attention Blocks, combined with residual connection and layer normalization, in STiSAN for stabilizing and speeding up the training process. Specifically, assuming the input is a vector $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{x} + \text{Layer}\left(\text{LayerNorm}\left(\boldsymbol{x}\right)\right), \tag{8}$$

where $\text{Layer}(\cdot)$ can be either the attention layer or the feed-forward layer and the normalization is conducted as:

$$\text{LayerNorm}\left(\boldsymbol{x}\right) = \alpha \odot \frac{\boldsymbol{x} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \tag{9}$$

where $\odot$ is an element-wise product (i.e., Hadamard product), μ and σ are the mean and standard-deviation of $\boldsymbol{x}$ respectively. α, ϵ and β are the learned parameters for scaling and biasing.

F. Target Aware Attention Decoder (TAAD)

According to [23], [29], [34], [35], the recommendation might be sub-optimal if we directly match the output of N-th IAAB $\boldsymbol{F}^{(N)}$ with candidate POI set C. Therefore, we follow [23] to introduce a Target Aware Attention Decoder (TAAD) to improve the representations of user preference over candidates,

$$\begin{aligned} \boldsymbol{S} &= \mathrm{TAAD}\left(\boldsymbol{F}^{(N)} \mid \boldsymbol{C}\right) = \mathrm{Attn}(\boldsymbol{C}, \boldsymbol{F}^{(N)}, \boldsymbol{F}^{(N)}), \\ \mathrm{Attn}(\boldsymbol{C}, \boldsymbol{F}^{(N)}, \boldsymbol{F}^{(N)}) &= \mathrm{Softmax}(\frac{\boldsymbol{C}{\boldsymbol{F}^{(N)}}^{T}}{\sqrt{d}})\boldsymbol{F}^{(N)}, \end{aligned} \tag{10}$$

where $\boldsymbol{S}$ is the representation matrix for user preference and $\boldsymbol{C}$ is the representation matrix for candidate set. We embed the candidate POIs the same way as input sequence (i.e., the concatenation of POI embeddings and GPS coordinates encodings). The aforementioned mask strategy is also needed in TAAD for preventing information leakage.

G. Matching and Ranking

Recall that the user's preference vector at the step i is $\boldsymbol{S}_i \in \mathbb{R}^{1\times d}$, we calculate the preference score $y_{i,j}$ over the candidate POI j with the following matching function (11),

$$y_{i,j} = f\left(\boldsymbol{S}_i, \boldsymbol{C}_j\right), \tag{11}$$

where $\boldsymbol{C}_j \in \mathbb{R}^{1\times d}$ is the representation vector of POI j and $f(\cdot)$ is the inner production.

After matching with all candidate POIs and ranking the corresponding preferences scores, the model recommends a list of Top-K POIs $TopK^u$ that the user u might visit in future.

H. Model Training

The binary cross-entropy loss function is widely-used for optimizing sequential recommenders [16], [28]. However, for the sake of efficient training, only one negative sample is randomly picked from all unvisited POIs, which cannot make fully effective use of the large number of negative samples [23].Thus, for each target POI o_i, we retrieve the L nearest POIs around it as negative samples, and we introduce the following weighted binary cross-entropy loss function proposed by [23] to optimize our model,

$$Loss = -\sum_{S^u\in S}\sum_{i=1}^{n}\left(\log\sigma(y_{i,o_i}) + \sum_{l=1}^{L} w_l \log(1-\sigma(y_{i,l}))\right), \tag{12}$$

where S is the set of all training sequences and $w_l = \frac{\exp(y_{i,l}/T)}{\sum_{l=1}^{L}\exp(y_{i,l}/T)}$ is the weight for negative POI l. T is the temperature parameter to control the distribution of negative samples. When T approaches positive infinity, the distribution of negative samples will be equivalent to uniform.

TABLE II
THE STATISTICS OF FOUR DATASETS (AFTER PRE-PROCESSED)

Dataset	Gowalla	Brightkite	Weeplaces	Changchun
#user	31,708	5,247	1,362	344,258
#POI	131,329	48,181	18,364	2,135
#check-in	2,963,373	1,699,579	650,690	21,471,724
sparsity	99.93%	99.33%	97.40%	97.08%
avg. seq. length	53.0	146.0	325.5	43.0

IV. EXPERIMENTS AND DISCUSSIONS

In this section, we first introduce the datasets, baselines, evaluation metrics, and implement details of STiSAN. Then, we analyse experimental results, including overall recommendation performance and ablation study. Moreover, we validate the extensibility and interpretability of TAPE and IAAB. Besides, we explore the model's sensitivity with respect to different datasets characteristics. In summary, we conduct large amount of experiments to answer the following questions:

- **RQ1** Can our STiSAN provide superior performance compared to several state-of-the-art baselines?
- **RQ2** How is the effectiveness of the two proposed components TAPE and IAAB under our framework?
- **RQ3** Can TAPE and IAAB be effectively employed to the vanilla self-attention network?
- **RQ4** How is STiSAN's sensitivity with respect to different sparsity levels?

A. Datasets

We choose three public LBSN datasets: Gowalla[5], Brightkite[6], Weeplaces[7] and one real-world city transportation dataset Changchun [36] to evaluate our proposed model. In order to ensure the quality of datasets, the "cold" users and "cold" POIs are filtered out during the pre-processing. Specifically, we remove the users who visit less than 20 POIs and the POIs that have been interacted with fewer than 10 times. During the partition of datasets, we take each user's most recent $n+1$ POIs in the whole sequence for evaluating (i.e., the last previously unvisited POI as target and the front n POIs as source input sequence), and all the POIs prior to the target for training. We set the maximum sequence length $n = 100$. Longer sequences will be divided into several non-overlapping sub-sequences of length n from the end, and shorter sequences will be repeatedly added "padding" POI in the head until their lengths grow to n.

B. Baselines

To evaluate the effectiveness of our proposed STiSAN, we compare it with various existing methods. For a better understanding of these baselines, we now briefly introduce our competitors:

[5]https://snap.stanford.edu/data/loc-gowalla.html
[6]https://snap.stanford.edu/data/loc-brightkite.html
[7]https://www.yongliu.org/datasets.html

- **POP** is a simple popularity based model which first calculates the interaction frequency of each POI and then recommends the most popular POIs for users.
- **BPR** [8] is a generic optimization criterion for personalized ranking. We apply it to the normal matrix factorization for users' preferences modeling.
- **FPMC-LR** [19] extends basic matrix factorization with geography constraints to learn the personalized transition matrix between POIs.
- **PRME-G** [20] utilizes metric learning to project users and POIs into the sequential transition space, user preference space, respectively, to learn user-specific transition patterns. It considers geographical factors by multiplying a travel-distance based weight.
- **GRU4Rec** [13] is a basic GRU based model for sequential recommendation. We implement it under our framework, i.e., use all prior POIs for training
- **Caser** [14] is a CNN based model for sequential recommendation. It employs horizontal and vertical convolution filters to capture sequential dependency from the local and global perspectives simultaneously.
- **STGN** [37] is a state-of-the-art LSTM based method, which designs extra gates for capturing the spatial-temporal correlations between successive POIs.
- **SASRec** [16] is the classic framework of applying self-attention based encoder for sequential recommendation.
- **Bert4Rec** [18] analyses the limitations of unidirectional sequential recommenders and models user's behavior sequence under the framework of BERT [38].
- **TiSASRec** [28] argues that different time intervals between interactions will influence the prediction and explore such impact with the proposed time aware self-attention layer.
- **GeoSAN** [23] exploits a novel self-attention based geography encoder which shows the state-of-the-art performance in modeling POIs' exact locations.
- **STAN** [29] is a state-of-the-art sequential POI recommender that explicitly models the relative spatial-temporal information among all POIs with the proposed bi-layer attention architecture.

C. Metrics

We adopt two widely-used metrics, Hit Rate, and Normalized Discounted Cumulative Gain (NDCG) [39], to measure how well the target POIs in the test set are ranked. The larger the values of metrics are, the better the performance is. We report the two metrics at $k = 5$ and $k = 10$ in our experiments. The specific information is as follows:

- Hit Rate at a cutoff k, denoted as HR@k, counts the fraction of times that target POI is among the top k recommendation list, as formulated in (13),

$$HR@k = \frac{\sum_{|Eval|} |TopK_k \cap trg|}{|Eval|}, \tag{13}$$

  where $Eval$ is the evaluation set. The recommendation list and target for each historical sequence in $Eval$ are denoted as $TopK_k$ and trg respectively. Note that since the number of target POI is 1, the metrics of Recall and Hit Ratio are equivalent in such scenario [40].
- NDCG at a cutoff k, denoted as NDCG@k, rewards the method that ranks the positive items in the first few positions of the Top-K ranking list, as (14),

$$NDCG@k = \frac{1}{D} \sum_{i=1}^{k} \frac{2^{|TopK_i \cap trg|} - 1}{\log_2(i+1)}, \tag{14}$$

  where $TopK_i$ is the i-th POI in the recommendation list and D, the maximum possible value of $DCG@k$, is a normalization constant.

For the sake of the efficient evaluation, we retrieve the nearest 100 previously unvisited POIs around the target as negative candidates. Hit Rate and NDCG can then be computed based on the ranking of the 101 POIs.

D. Settings

The implementation details of our STiSAN are listed as follows. For the late representations, we set the dimensions of POI embedding and GPS coordinates encoding to 128, and the sequence dimension d is concatenated to 256. For the preference modeling, we stack $N = 4$ Interval Aware Attention Blocks. During training process, we randomly pick $L = 15$ POIs from the traget's nearest 2000 neighbours as negative samples. We set the learning rate and dropout rate to 0.001 and 0.7. We train our STiSAN for 35 epochs on Gowalla, 20 epochs on Brightkite, Weeplaces and Changchun. The temperature parameter T is 1.0 for Gowalla, 100.0 for Brightkite and Weeplaces and 500.0 for Changchun. Our model is implemented on the PyTorch 1.8.0[8] and conducts all experiments on a server with 122GB RAM, 12-core AMD 9 Ryzen 5900X@3.7GHz CPU and Nvidia RTX 3090 GPU.

E. Validations and Discussions

1) Overall Performance (RQ1): To evaluate the effectiveness of our proposed STiSAN, we compare the overall recommendation performance of STiSAN with the twelve baselines. The experimental results are summarized in Table III. Note that we test all baselines for 10 rounds, and take the average value along with the variance as their final performance. The last row reveals the improvements of STiSAN over the strongest baseline. It is obvious that our method has superior performance over all datasets on every metric. Specifically, we have the following observations:

- Conventional popularity or matrix factorization based methods like POP, BPR, and FPMC-LR have relatively unsatisfactory performance compared to other baselines. The main reason is the insufficient modeling of sequential dependency or high-order interaction information.
- Due to the inability of processing long sequences and the insufficient modeling of geographical information, STGN performs poorly on four datasets even if it has considered the spatial and temporal intervals. Meanwhile,

[8]https://github.com/jiangyiheng1/STiSAN.pytorch.

TABLE III
OVERALL RECOMMENDATION PERFOMANCE COMPARISON (THE BEST SCORES ARE BOLDFACED AND THE SECOND SCORES ARE UNDERLINED)

Dataset	Gowalla				Brightkite				Weeplaces				Changchun			
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	HR@5	NDCG@5	HR@10	NDCG@10
POP	0.0146±0.000	0.0110±0.000	0.0266±0.000	0.0170±0.000	0.0259±0.000	0.0202±0.000	0.0423±0.000	0.0273±0.000	0.0369±0.000	0.0292±0.000	0.0575±0.000	0.0373±0.000	0.0246±0.000	0.0189±0.000	0.0420±0.000	0.0287±0.000
BPR	0.0142±0.001	0.0107±0.000	0.0263±0.000	0.0168±0.001	0.0450±0.001	0.0344±0.001	0.0760±0.001	0.0492±0.001	0.0749±0.001	0.0574±0.001	0.1023±0.000	0.0807±0.002	0.0681±0.004	0.0462±0.002	0.0954±0.000	0.0699±0.000
FPMC-LR	0.1264±0.000	0.0889±0.002	0.2005±0.001	0.1121±0.004	0.1731±0.006	0.1307±0.000	0.2534±0.002	0.1574±0.001	0.1975±0.000	0.1182±0.003	0.2811±0.005	0.2082±0.000	0.1738±0.001	0.0942±0.002	0.2567±0.000	0.1840±0.002
PRME-G	0.3408±0.003	0.2638±0.003	0.4579±0.003	0.3019±0.003	0.4260±0.003	0.3329±0.003	0.5442±0.003	0.3711±0.003	0.2595±0.003	0.1951±0.003	0.3549±0.003	0.2258±0.003	0.2317±0.003	0.1684±0.003	0.3372±0.003	0.2017±0.003
GRU4Rec	0.3264±0.003	0.2471±0.003	0.4503±0.002	0.2911±0.002	0.4078±0.003	0.3301±0.004	0.5282±0.003	0.3550±0.003	0.2817±0.000	0.2094±0.002	0.3838±0.003	0.2423±0.001	0.2535±0.001	0.1806±0.005	0.3528±0.002	0.2185±0.003
Caser	0.2327±0.001	0.1876±0.004	0.3688±0.002	0.2049±0.002	0.3164±0.001	0.2123±0.003	0.4302±0.001	0.3145±0.003	0.2735±0.003	0.1964±0.005	0.3712±0.000	0.2403±0.003	0.2691±0.001	0.1786±0.002	0.3577±0.001	0.2322±0.001
STGN	0.1655±0.001	0.1171±0.002	0.2915±0.002	0.1603±0.001	0.2721±0.002	0.1892±0.003	0.3614±0.000	0.2375±0.003	0.1864±0.000	0.1089±0.002	0.2671±0.003	0.1980±0.003	0.1378±0.001	0.0854±0.000	0.2176±0.001	0.1563±0.003
SASRec	0.3243±0.000	0.2452±0.000	0.4489±0.002	0.2853±0.001	0.4042±0.001	0.3217±0.000	0.5115±0.000	0.3562±0.003	0.2907±0.000	0.2171±0.000	0.3950±0.000	0.2507±0.000	0.1956±0.000	0.1435±0.000	0.3094±0.002	0.2387±0.002
Bert4Rec	0.3317±0.000	0.2440±0.001	0.4625±0.003	0.2853±0.002	0.3950±0.000	0.3051±0.000	0.5036±0.000	0.3424±0.000	0.2902±0.002	0.2105±0.004	0.3997±0.000	0.2614±0.001	0.2140±0.001	0.1577±0.000	0.3384±0.000	0.2703±0.002
TiSASRec	0.3326±0.001	0.2562±0.001	0.4831±0.000	0.3161±0.002	0.4086±0.000	0.3143±0.001	0.5122±0.000	0.3593±0.002	0.3051±0.001	0.2316±0.002	0.4379±0.000	0.2791±0.002	0.2039±0.003	0.1462±0.000	0.3143±0.000	0.2455±0.000
GeoSAN	0.4153±0.001	0.3327±0.001	0.5251±0.001	0.3680±0.001	0.4843±0.001	0.3958±0.003	0.5916±0.002	0.4303±0.002	0.3480±0.001	0.2677±0.002	0.4699±0.003	0.3069±0.000	0.2306±0.001	0.1725±0.002	0.3424±0.000	0.2706±0.002
STAN	0.4369±0.003	0.3544±0.001	0.5384±0.004	0.3864±0.005	0.4736±0.002	0.3819±0.002	0.5670±0.001	0.4263±0.001	0.3276±0.001	0.2341±0.003	0.4349±0.002	0.2830±0.001	0.2218±0.003	0.1695±0.002	0.3259±0.002	0.2597±0.003
STiSAN	**0.4617**	**0.3721**	**0.5679**	**0.4053**	**0.5310**	**0.4339**	**0.6512**	**0.4727**	**0.4332**	**0.3437**	**0.5558**	**0.3833**	**0.2653**	**0.1944**	**0.3786**	**0.3075**
Improv.	5.68%	4.99%	5.48%	4.89%	9.64%	9.63%	10.07%	9.85%	24.48%	28.39%	18.28%	24.89%	15.04%	12.71%	10.56%	13.64%

the performance of GRU4Rec proves the effectiveness of our training strategy, i.e., dividing long sequences into sub-sequences rather than only using the most recent POIs, and we find that such strategy is also suitable for SASRec. Caser shows higher recommendation accuracy than STGN because the vertical convolution operations help the model aggregate global sequence information, which avoids the gradient vanishing issue in RNN.

- Thanks to the modeling of geographical factors and evaluation metric (i.e., ranking target with nearest POIs), PRME-G has better performance than CNN/RNN-based and even some self-attention based-methods on Gowalla, Brightkite and Changchun.
- In general, self-attention-based methods have more stable and relative better performance for the strong ability to capture global sequence information. However, due to the neglect of geographical factors dominating the sequential POI recommendation scenario, SASRec, TiSASRec, and Bert4Rec are not as good as GeoSAN and STAN.
- GeoSAN and STAN are two strong baselines with decent performance. For the Gowalla dataset, the embedded spatial-temporal intervals help STAN to gain higher accuracy. While for Brightkite, Weeplaces and Changchun, the geographical location modeling and the importance based negative sampling can make up for the lack of temporal factors in GeoSAN, especially under the geography based evaluation metric.
- Our STiSAN consistently outperforms all baselines with a large margin on all four datasets and achieves up to 13.71% HR@5, 13.93% NDCG@5, 11.10% HR@10 and 13.32% NDCG@10 improvements (on average) over the second-best performances.

2) Ablation Study (RQ2): To analyse the influence of various components on our framework, we conduct ablation study. Our base model (denoted as Original) contains geography encoder [23], Time Aware Position Encoder, Interval Aware Attention Block, and Target Aware Attention Decoder. We consider the following variants of our base model:

- I. *Remove GE:* We remove the geography encoder, and use POI embedding and Time Aware Position Encoder to represent POI sequence.
- I. *Remove TAPE:* We remove the Time Aware Position Encoder, and use POI embedding, geography encoder and vanilla positional encoding to represent POI sequence.
- II. *Remove IAAB:* We remove the spatial-temporal relation matrix in (6) and modify it to (15),

$$\boldsymbol{A} = \mathrm{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}})\boldsymbol{V}, \tag{15}$$

- III. *Remove SA:* We remove the self-attention mechanism in (6) and only use the spatial-temporal relation matrix to prediction, as formulated in (16),

$$\boldsymbol{A} = \mathrm{Softmax}(\boldsymbol{R})\boldsymbol{V}, \tag{16}$$

- IV. *Remove TAAD:* We remove the Target-Aware Attention Decoder and match the output of the N-th IAAB with candidate POI directly, as formulated in (17),

$$y_{i,j} = f\left(\boldsymbol{F}_i^{(N)}, \boldsymbol{C}_j\right). \tag{17}$$

The results are summarized in Table IV. From this table, we can have the following findings:

- *Finding 1: The Time Aware Position Encoder is proved to be helpful in enhancing sequence representation.* Firstly, we find that removing TAPE leads to 3.98%, 2.58%, and 8.03% performance decrease in terms of NDCG@5 on three datasets respectively, which proves the effectiveness of our TAPE in helping model to capture relative temporal proximity between POIs. Moreover, we can see that removing either spatial (Variant I) or temporal (Variant II) descends the performance of Original. It demonstrates that spatial and temporal information are both pivotal in representing sequences.
- *Finding 2: The Interval Aware Attention Block can provide attentive results with positive revisions.* Comparing Variant III with Original, we can see that removing IAAB descends the performance in terms of NDCG@5 by 3.46%, 1.38%, and 6.26% on three datasets, respectively. It indicates that our IAAB can attach more importance

TABLE IV
ABLATION STUDY (THE BEST SCORES ARE BOLDFACED)

Dataset	Gowalla				Brightkite				Weeplaces			
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
Original	0.4617	0.3721	0.5679	0.4053	**0.5310**	**0.4339**	**0.6512**	**0.4727**	**0.4332**	**0.3437**	**0.5558**	**0.3833**
I. -GE	0.4080	0.3269	0.5082	0.3588	0.4002	0.3270	0.4911	0.3563	0.3737	0.2935	0.4853	0.3297
II. -TAPE	0.4485	0.3573	0.5524	0.3902	0.5203	0.4227	0.6388	0.4611	0.3899	0.3161	0.4993	0.3512
III. -IAAB	0.4522	0.3592	0.5564	0.3921	0.5230	0.4279	0.6394	0.4658	0.3994	0.3222	0.5132	0.3588
IV. -SA	0.4145	0.3172	0.5217	0.3511	0.4835	0.3893	0.5956	0.4255	0.3634	0.2767	0.4875	0.3165
V. -TAAD	**0.4643**	**0.3780**	**0.5682**	**0.4087**	0.5176	0.4233	0.6322	0.4602	0.4134	0.3246	0.5257	0.3609

to the spatial correlations, i.e, achieving more reasonable attention weight assignment.

- *Finding 3: The proposed TAPE and IAAB bring more significant influence on Weeplaces.* The main reason lies in the characteristics of these three datasets. As shown in Table II, the average POI sequence length in Weeplaces (325.5) is much longer than Gowalla (53.0) and Brightkite (146.0). According to our empirical observation, the spatial and temporal spans in the POI sequence are positively correlated with its length, while TAPE and IAAB are advanced in handling the relative spatial-temporal proximity for longer sequences.
- *Finding 4: Our framework still has competitive performance even without self-attention mechanism.* Comparing Variant IV with the strongest baselines, we are surprised to find that the recommendation accuracy, solely based on the enhanced sequence representation with relative temporal proximity and spatial-temporal relation matrix, is slightly lower on Brightkite while higher on Weeplaces. It demonstrates that the sequential dependencies, learned by the self-attention mechanism and contained in spatial-temporal intervals, have some similarities and can accomplish each other. We will delicately explore whether the influence of spatial-temporal intervals is beyond the self-attention mechanism in future work.
- *Finding 5: The Target-Aware Attention Decoder is useful only in certain circumstances.* The possible reason is that TAAD neglects the spatial intervals between the last POI in the input sequence and candidate POIs.

3) Extensibility and Interpretability of TAPE (RQ3): We set two experiments to answer this question from the angles of metric evaluation and principle. Firstly, we replace the positional encoding (denoted as PE) with TAPE in a vanilla Self-Attention Network to verify whether or not TAPE can bring improvements. The experimental results are revealed in Fig. 4, and we can see that TAPE leads to average 5.36% HR@10 improvement over all datasets. It proves that our TAPE can be effectively adopted to the self-attention network.

Take a step further, towards the question "*Why TAPE ?*", we conduct the following visualization experiment. Firstly, we randomly pick a user from Weeplaces whose historical

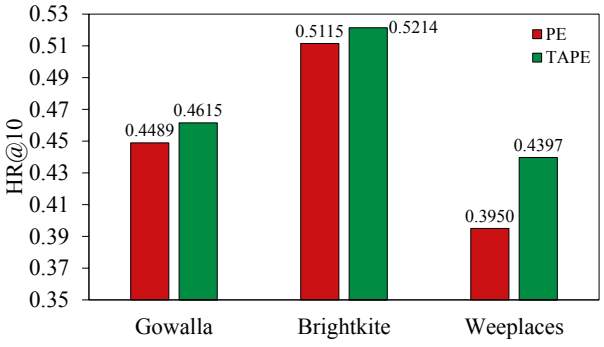
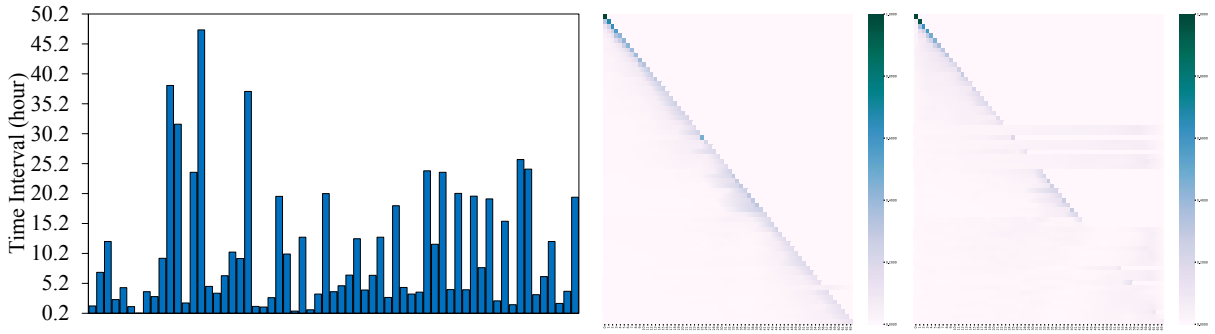


Fig. 4. Extensibility of TAPE.

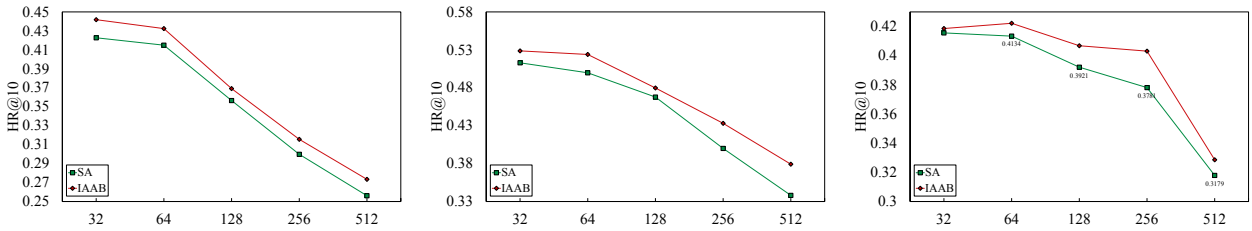
sequence length is 64, and visualize the time intervals between successive visited POIs. From Fig. 5(a), we can see that the distribution is non-uniform and varies significantly. The diagonal elements in PE's and TAPE's average attention heat-map reveal the difference between these two approaches. As shown in Fig. 5(b) and 5(c), TAPE will partly strengthen or distract the attention on the current position, and takes the corresponding counter operation on the previous position (as the lower triangle in heat-maps). In other words, the smaller time interval between two successive POIs leads to the more similar attention weights and vice versa, where proving that the relative temporal proximity can be effectively captured by the self-attention mechanism. Thus, we believe in that our TAPE is a meaningful, practical and of course lightweight approach for enhancing sequence representations.

4) Extensibility and Interpretability of IAAB (RQ3): We also set two experiments to answer this question from the view of metric evaluation and principle separately. Firstly, we replace the Self-Attention mechanism (denoted as SA) in a vanilla 4-layer Self-Attention Network with our IAAB to explore the influence of various sequence lengths. As shown in Fig. 6(a), 6(b) and 6(c), due to the insufficient attention of local POIs, SA's performance decreases dramatically especially when sequence length increases from 64 to 128. Our IAAB effectively relieve this issue, and even helps SA achieving superior recommendation accuracy on the length of



(a) Distribution of time intervals between successive visited POIs (b) Attention Heat-map of PE (c) Attention Heat-map of TAPE

Fig. 5. Interpretability of TAPE.



(a) Gowalla (b) Brightkite (c) Weeplaces

Fig. 6. Extensibility of IAAB

64 in Weeplaces. 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 POIs. Fig. 7(a) shows the geography intervals between historical and target POIs, and we can see that the strong spatial correlated POIs are at the positions from 1 to 32 and from 44 to 64. Comparing the attention heat-maps of SA and IAAB (as shown in Fig. 7(b) and 7(c)), it is obvious that our IAAB can pay significant attention to these vital POIs especially those distributed at more forward positions in this sequence. The experimental results also demonstrate that our method can provide explainable recommendations.

5) Sensitivity w.r.t Sparsity levels (RQ4): For the comprehensive analysis of our STiSAN, we set different cold user / POI thresholds in Weeplaces to achieve different sparsity levels as shown in Table V. Then, we compare STiSAN with two strongest baselines STAN and GeoSAN as shown in Fig. 8. According to the results, our STiSAN outperforms these two baselines over all sparsity levels. Moreover, we find that all methods' performance increases first and then decreases along with the dataset is becoming more dense. It is because the under-fitting issue caused by the derisory training instances (e.g., 92 users and 1324 POIs).

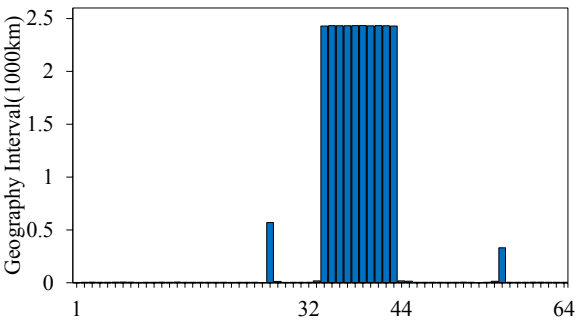
6) Computational Complexity Analysis: To further verify whether or not our method is lightweight, we calculate the Floating Point Operations (FLOPs) of our IAAB and the 4-layer self-attention mechanism (denoted as SA). As shown in Table VI, the additional computational burden is negligible

TABLE V
THE STATISTICS OF WEEPLACES UNDER DIFFERENT SPARSITY LEVELS

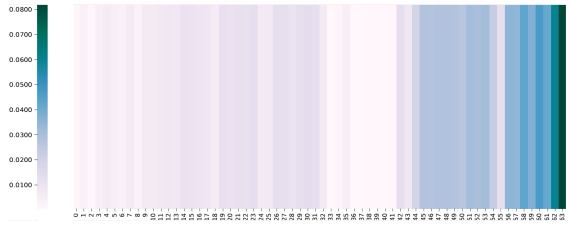
Dataset	Weeplaces			
cold POI threshold	30	60	80	90
cold user threshold	60	120	140	150
#users	709	278	133	92
#POIs	5,452	2,305	1,550	1,324
#check-ins	329,268	126,464	59,506	43,408
sparsity	91.48%	80.26%	71.13%	64.36%

(e.g. only adds 0.01M FLOPs on Brightkite and Changchun).

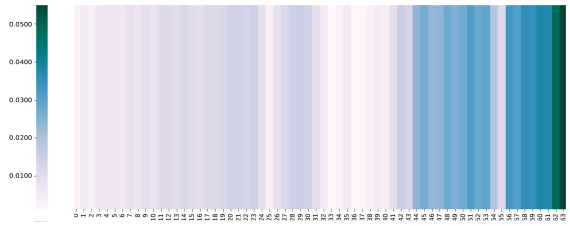
7) Hyper-parameters Analysis: We explore the influence of the thresholds k_t, k_d, which controlling the maximum time interval and geography interval in spatial-temporal relation matrix. We set $k_t = \{0, 5, 10, 20\}$ days. Correspondingly, we set $k_d = \{0, 5, 10, 15\}$ kilometers. As the blue columns in Fig. 9, when setting both k_t and k_d to zero, the recommendation accuracy reaches the lowest on all four datasets. This is because the entries in the spatial-temporal relation matrix are all zero at this time, and it will add the same value to attention weights after the Softmax function in the interval aware attention layer, which actually disabling the IAAB. Further, STiSAN achieves the best performance at $k_t = 5, k_d = 5$ on Weeplaces, Changchun and tends to be stable when k_t, k_d keeps increasing. For Gowalla and Brightkite, the most suitable sets are $k_t = 10, k_d = 15$.



(a) Distribution of geography intervals



(b) Attention Heat-map of SA



(c) Attention Heat-map of IAAB

Fig. 7. Interpretability of IAAB.

TABLE VI
COMPUTATIONAL COMPLEXITY COMPARISON (FLOPS)

Datasets	Gowalla	Brightkite	Weeplaces	Changchun
SA	0.83M	0.13M	0.04M	8.75M
IAAB	0.83M	0.14M	0.04M	8.76M

V. RELATED WORKS

In this section, we first review the related literature, including attention mechanism, positional representation and sequential POI recommendation. Then, we discuss the connections and differences between our method and existing models.

A. Attention Mechanism

Attention mechanism has been proved to be effective in various tasks ranging from computer vision, natural language processing to sequential recommender systems [16]. The core idea behind such a mechanism is impelling the model to attach more importance to the more relevant parts of input. In the scenario of recommendation, existing methods, such as [41]–[43], employ attention mechanisms to learn the importance of

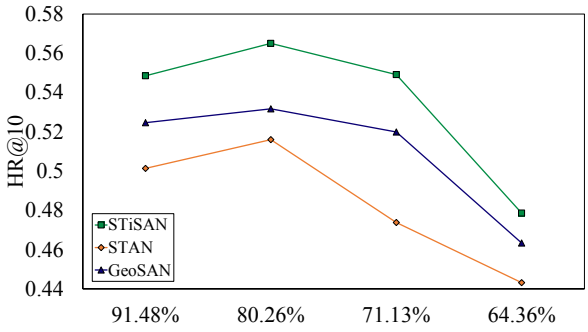
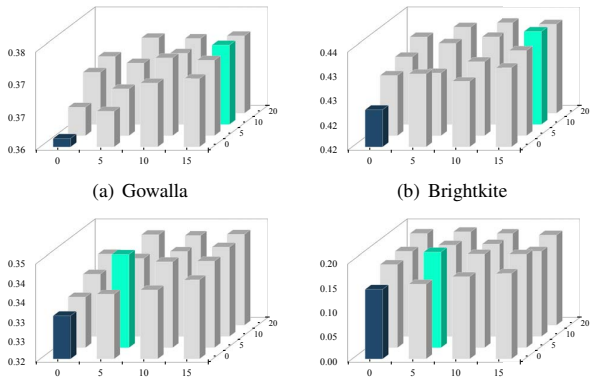


Fig. 8. Sensitivity w.r.t different sparsity levels.



(a) Gowalla (b) Brightkite

(c) Weeplaces (d) Changchun

Fig. 9. Sensitivity w.r.t different hyper-parameter settings (NDCG@5).

different items, features, or interactions for predicting users' preferences on unobserved candidates.

However, these models all treat the attention mechanism as an additional component to a basic framework (e.g. attention+RNN/CNN) [33]. Note that these models' performance might be limited by issues caused the characteristics of the original framework, such as gradient vanishing [44] and data sparsity [45]. Recently, Transformer [15], a model solely based on the attention mechanism, has shown its superior performance in machine translation tasks which used to be dominated by RNN/CNN-based approaches [16]. It proposed a novel self-attention mechanism to calculate attention weights of different tokens dynamically and flexibly according to the input sequence. Inspired by the advanced performance, [16] first introduced the self-attention-based encoder in sequential item recommendation problems and validated its effectiveness.

The inherent drawback of self-attention is that the global weights averaging inhibits the spatial relation among local POIs [30], [31]. Our Interval Aware Attention Block (IAAB), introducing the spatial-temporal relation matrix into attention map as inductive bias, can help relieving this issue.

B. 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 [15] encodes the positions by sinusoidal function, an approach of fixed and symmetric, and then adds them into sequence representations. In this way, the tokens' order in sequence can be captured when carrying dot-product operations on sequence representations. For large-scale and pre-trained models, [38] shows that learning a set of absolute positional embedding performs better than the fixed positional encoding. Differently, [46] focuses on learning the relative positional relations among tokens rather than directly learning positions. There are also some works utilizing hybrid representations that contain both absolute and relative positional information [47], [48].

We are sparked by these works and exploit the Time Aware Position Encoder (TAPE) to encode the relative temporal proximity among POIs into sequence representations, where requiring neither extra learnable parameters nor significant computational burden.

C. Sequential POI Recommendation

Along with the information technology developing and user-POI interaction data accumulating, methods for sequential POI recommendation have been evolving from Markov Chain [3]–[5] and Matrix Factorization [6]–[10] to Multi-Layer Perceptron (MLP) [12], [49], [50], Recurrent Neural Network (RNN) [51]–[55] and Convolution Neural Network (CNN) [56]–[58] over the past decades. For instance, Factorizing Personalized markov chains (FPMC) [11] linearly combines markov chains and matrix factorization to model the personalized transition between POIs. [19] extends FPMC to address the sparsity issue of modeling personalized POI transitions. For neural network-based models, Caser [14] utilizes horizontal and vertical convolution kernels to capture sequential dependency from the perspective of local and global simultaneously, and GRU4Rec [13] employs a modified gated recurrent unit to learn the pattern of users' dynamic preference. STGN [37] incorporated spatial-temporal information by designing spatial-temporal gates for controlling information flow. SASRec [16] first introduced self-attention network into the sequential recommendation problem. Bert4rec [18] extended SASRec to capture bidirectional sequential dependency. For considering spatial or temporal factors, GeoSAN [23] proposes a self-attention based geography encoder to encode POIs' locations and TiSASRec [28] designs a time aware self-attention mechanism to explore the influence of different time intervals on prediction. In the most recent STAN [29] exploits a novel spatial-temporal attention network, utilized the learned spatial-temporal interval representations, has achieved the state-of-the-art performance.

The difference of our proposed STiSAN from these exiting works lies in that our framework, as an end-to-end deployment, utilizes lightweight approaches (i.e., no extra learnable parameters and negligible computational complexity) TAPE and IAAB to consider spatial-temporal factors. On the one hand, TAPE mainly focuses on enhancing the sequence representations via encoding time intervals. On the other hand, IAAB promotes attention mechanism attaching focus importance to spatial relations and provides explainable recommendations.

VI. Concluding Remarks

In this paper, we propose two practical, meaningful and lightweight approaches, Time Aware Position Encoder (TAPE) and Interval Aware Attention Mechanism (IAAB), to promote the self-attention network considering spatial-temporal factors among POIs, where requiring neither extra parameters nor significant computational burden. On the one hand, TAPE, encoding the timestamps into sequence representations, can reflect the relative temporal proximity. On the other hand, IAAB, introducing the spatial-temporal relation into attention map explicitly, impels the attention on spatial correlations. We integrates these two approaches into the self-attention network, and proposes a sequential POI recommender which achieves state-of-the-art performance (13.01% improvement by average) on three public datasets and one real-world city transportation dataset. Aside from evaluating TAPE's and IAAB's effectiveness under our framework through ablation study, we also combine them with the vanilla self-attention network. We separately conduct metric comparison and visualization to validate their extensibility and interpretability. Moreover, we explore STiSAN's sensitivity with respect to different sparsity levels and hyper-parameter settings. Finally, we analyse the Floating Point Operations to prove the lightweight.

In future, we will delicately explore the connections and differences between the sequential dependencies learned by self-attention and contained in spatial-temporal relation matrix.

Acknowledgment

We thank anonymous reviewers for their helpful comments. This research is supported by the National Key R&D Program of China under Grants 2021ZD0112501 and 2021ZD0112502, in part by the National Natural Science Foundations of China under Grants 61772230, 61972450, 62072209, and 62102161, in part by the National Natural Science Foundation of China for Youth Scholars under Grant 62002123, in part by the CCF-Baidu Open Fund under Grant 2021PP15002000, and in part by NSF under Grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128, in part by the Jilin Provincial Key Science and Technology Development Program under Grant 20210201082GX, in part by the Jilin Provincial Development and Reform Commission Program under Grant 2020C017-2, in part by the Jilin Provincial Education Department Science and Technology Program under Grant JJKH20221010KJ.

References

[1] H. Fang, D. Zhang, Y. Shu and G. Guo, "Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations," in *TOIS*, vol. 39, no. 1, pp.10:1-10:42, 2020.

[2] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Sheng and M. A. Orgun, "Sequential Recommender Systems: Challenges, Progress and Prospects," in *IJCAI*, 2019, pp. 6332-6338.

[3] R. He, C. Fang, Z. Wang and J. J. McAuley, "Vista: A Visually, Socially, and Temporally-aware Model for Artistic Recommendation," in *RecSys*, 2016, pp. 309-316.

[4] R. He, W. Kang and J. J. McAuley, "Translation-based Recommendation," in *RecSys*, 2017, pp. 161-169.

[5] R. He and J. J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *ICDM*, 2016, pp. 191-200.

[6] S. Kabbur, X. Ning and G. Karypis, " Fism: Factored item similarity models for top-n recommender systems," in *KDD*, 2013, pp. 659-667.

[7] Y. Koren, " Collaborative filtering with temporal dynamics," in *KDD*, 2009, pp. 447-456.

[8] S. Rendle, C. Freudenthaler and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp.452-461.

[9] S. Rendle, Z. Gantner, C. Freudenthaler and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *SIGIR*, 2011, pp. 635-644.

[10] B. Twardowki, "Modelling contextual information in session-aware recommender systems with neural networks," in *RecSys*, 2019, pp. 273-276.

[11] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.

[12] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan and X. Cheng, "Learning hierarchical representation model for next basket recommendation," in *SIGIR*, 2015, pp. 403-412.

[13] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.

[14] J. Tang and K. Wang,"Personalized top-n sequential recommendation via convolutional sequence embedding" in *WSDM*, 2018, pp. 565–573.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[16] W.-C. Kang and J. McAuley,"Self-attentive sequential recommendation" in *ICDM*, 2018, pp. 197–206.

[17] L. Wu, S. Li, C-J. Hsieh and J. Sharpback, "SSE-PT: Sequential Recommendation Via Personalized Transformer," in *RecSys*, 2020, pp. 328-337.

[18] 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 *CIKM*, 2019, pp. 1441-1450.

[19] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: successive point-of-interest recommendation," in *IJCAI*, 2013, pp. 2605–2611.

[20] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *IJCAI*, 2015, pp. 2069–2075.

[21] R. Li, Y. Shen, and Y. Zhu, "Next point-of-interest recommendation with temporal and multi-level context attention," in ICDM, IEEE, 2018, pp. 1110–1115.

[22] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: a recurrent model with spatial and temporal contexts," in *AAAI*, 2016, pp. 194–200.

[23] D. Lian, Y. Wu, Y. Ge, X. Xie and E. Chen, "Geography-Aware Sequential Location Recommendation," in *KDD*, 2020, pp.2009-2019.

[24] D. Lian, C. Zhao, G. Sun, E. Chen and Y. Rui, "GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *KDD*, 2014, pp. 831-840.

[25] Y. Mao, P. Yin, W. Lee and D. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation", in *SIGIR*, 2011, pp. 325-334.

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

[27] 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," in *IEEE Trans. Mob. Comput*, vol. 20, no. 2, pp. 754-764, 2021.

[28] J. Li, Y. Wang, and J. McAuley, "Time Interval Aware Self-Attention for Sequential Recommendation," in *WSDM*, 2020, pp. 322–330.

[29] Y. Luo, Q. Liu and Z. Liu, "STAN: Spatio-Temporal Attention Network for Next Location Recommendation," in *WWW*, 2021, pp. 2177-2185.

[30] D. Zhou, Y. Shi, B. Kang, W. Yu, Z. Jiang, Y. Li, Q. Hou and J. Feng, "Refiner: Refining Self-attention for Vision Transformers," in *CoRR*, 2021, https://arxiv.org/abs/2106.03714.

[31] B. Yang, Z. Tu, D. Wong, F. Meng, L. Chao and T. Zhang, "Modeling Localness for Self-Attention Networks," in *EMNLP*, 2018, pp. 4449-4458.

[32] F. Yu, L. Cui, W. Guo, X. Lu, Q.g Li and H. Lu, "A CategoryAware Deep Model for Successive POI Recommendation on Sparse Check-in Data," in *WWW*, 2020.

[33] J. Zhao, P. Zhao, L. Zhao, Y. Liu, V. S. Sheng and X. Zhou, "Variational Self-attention Network for Sequential Recommendation," in *ICDE*, 2021, pp.1559-1570.

[34] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," in *KDD*, 2018, pp. 1754–1763.

[35] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li,and K. Gai, "Deep interest network for click through rate prediction," in *KDD*, 2018, pp. 1059–1068.

[36] E. Wang, Y. Xu, Y. Yang, F. Yang, C. Liu and Y. Jiang, "ToP: Time-dependent Zone-enhanced Points-of-interest Embedding-based Explainable Recommender system," in *INFOCOM*, 2021.

[37] P. Zhao, H. Zhu, 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," in *AAAI*, 2019, Vol. 33, pp. 5877–5884.

[38] J. Devlin, M-W. Chang, K. Lee and K. Toutanova"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[39] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola, "Maximum margin matrix factorization for collaborative ranking," in *NIPS*, 2007, pp. 1–8.

[40] W. Hrichene and S. Rendle, "On Sampled Metrics for Item Recommendation," in *KDD*, 2020.

[41] 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 *SIGIR*, 2017, pp. 335–344.

[42] 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 *IJCAI*, 2017, pp. 3119–3125.

[43] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attention-based transactional context embedding for next-item recommendation," in *AAAI*, 2018, 2532–2539.

[44] U. Khandelwal, H. He, P. Qi, and D. Jurafsky, "Sharp nearby, fuzzy far away: How neural language models use context," *arXiv* preprint arXiv:1805.04623, 2018.

[45] C. Song, Z. Qu, N. Blumm, and A.L. Barabási, "Limits of predictability in human mobility," in *Science*, 2010, pp. 1018–1021.

[46] P. Shaw, J. Uszkoreit and A. Vaswani, "Self-Attention with Relative Position Representations," in *NAACL-HLT*, 2018, pp. 464–468.

[47] G. Ke, D. He and T-Y. Liu, "Rethinking Positional Encoding in Language Pre-training," in *ICLR*, 2021.

[48] J. Su, Yu Lu, S. Pan, B. Wen and Y. Liu, "RoFormer: Enhanced Transformer with Rotary Position Embedding," in *arXiv* https://arxiv.org/abs/2104.09864.

[49] S. Wan, Y. Lan, P. Wang, J. Guo, J. Xu and X. Cheng, "Next basket recommendation with neural networks," in *RecSys*, 2015.

[50] C. Wu and M. Yan, "Session-aware information embedding for e-commerce product recommendation," in *CIKM*, 2017, pp. 2379-2382.

[51] D. Le, H. W. Lauw, and Y. Fang, "Modeling contemporaneous basket sequences with twin net-works for next-item recommendation," in *IJCAI*, 2018, pp. 3414–3420.

[52] 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 *KDD*, 2018, PP. 1734–1743.

[53] Q.,S. Wu,and L. Wang, "Multi-behavioral sequential prediction with recurrent log-bilinear model," in *TKDE*, 2017, vol. 6, pp. 1254–1267.

[54] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Bayesian personalized ranking with multi-channel user feedback," in *RecSys*, 2016, pp. 361–364.

[55] E. Smirnova and F. Vasile, "Contextual sequence modeling for recommendation with recurrent neural networks," in *RecSys*, 2017, pp. 2–9.

[56] T. X. Tuan and T. M. Phuong, "3D convolutional networks for session-based recommendation with content features," in *RecSys*, 2017, pp. 138–146.

[57] 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*, 2020, pp. 303–313.

[58] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *WSDM*, 2019, pp. 582–590.