Time-aware User Modeling with Check-in Time Prediction for Next POI Recommendation

Xin Wang
School of Computer Science, Wuhan University
Wuhan, China
xinwang0920@whu.edu.cn

Xiao Liu
School of Information Technology, Deakin University
Geelong, Australia
xiao.liu@deakin.edu.au

Li Li
Faculty of Information Technology, Monash University
Melbourne, Australia
Li.Li@monash.edu

Xiao Chen
Faculty of Information Technology, Monash University
Melbourne, Australia
xiao.chen@monash.edu

Jin Liu*
School of Computer Science, Wuhan University
Wuhan, China
jinliu@whu.edu.cn

Hao Wu
School of Information Science and Engineering, Yunnan University
Kunming, China
haowu@ynu.edu.cn

2021 IEEE International Conference on Web Services (ICWS) | 978-1-6654-1681-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICWS53863.2021.00028

***Abstract*—POI (point-of-interest) recommendation as an important type of location-based services has received increasing attention with the rise of location-based social networks. Although significant efforts have been dedicated to learning and recommending users' next POIs based on their historical mobility traces, there still lacks consideration of the discrepancy of users' check-in time preferences and the inherent relationships between POIs and check-in times. To fill this gap, this paper proposes a novel recommendation method which applies multi-task learning over historical user mobility traces known to be sparse. Specifically, we design a cross-graph neural network to obtain time-aware user modeling and control how much information flows across different semantic spaces, which makes up the inadequate representation of existing user modeling methods. In addition, we design a check-in time prediction task to learn users' activities from a time perspective and learn internal patterns between POIs and their check-in times, aiming to reduce the search space to overcome the data sparsity problem. Comprehensive experiments on two real-world public datasets demonstrate that our proposed method outperforms several representative POI recommendation methods with 8.93% to 20.21% improvement on Recall@1, 5, 10, and 9.25% to 17.56% improvement on Mean Reciprocal Rank.**

***Index Terms*—Location-based Services, Location-based Social Networks, POI Recommendation, Multi-Task Learning, Cross-Graph Neural Network, Time-aware User Modeling**

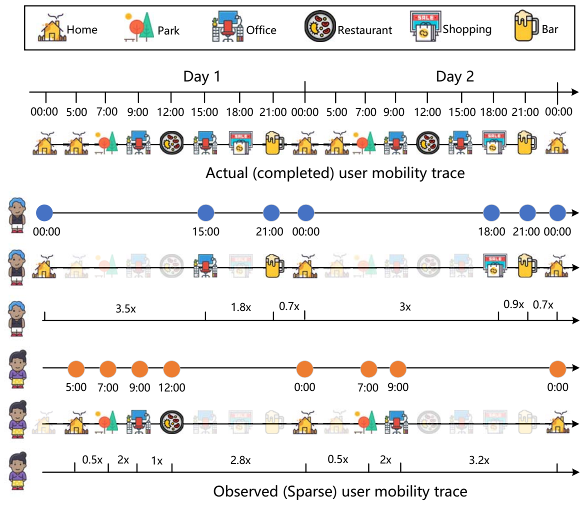


Fig. 1. Examples of spare user mobility traces (check-in POIs and check-in timeslots).

I. Introduction

Location-based services have become increasingly popular with the rapid growth of location-based social networks (LBSNs) [1]–[3], such as GoWalla[1], Foursquare[2] and JiePang[3]. User mobility traces accumulated in these online platforms offer valuable opportunities for understanding human dynamics, which is a fundamental ingredient for developing smart services [4]–[8]. Both customers and service providers can benefit from personalized **P**oint-**O**f-**I**nterest (POI) recommendation services. For customers, high-quality POI recommendation can help them find interesting spots when they are in unfamiliar areas. For service providers, POI recommendation service can increase the exposure of their services so as to achieve more profits. POI recommendation is an important type of location-based services which aims to predict a user's next location based on user's historical check-in sequence. POI recommendation is important to both users and service providers, and has thus attracted growing attention from researchers in recent years. Many existing efforts have been devoted for better POI recommendations [9]–[13].

Traditional methods capture user mobility patterns using hand-crafted features, such as historical visit counts [14]. These methods, while exploiting static features of check-in histories, ignore sequential patterns of user mobility traces. More recently, researchers attempt to leverage deep learning

*Corresponding author.
[1]https://blog.gowalla.com/
[2]https://foursquare.com/
[3]https://jiepang.app/

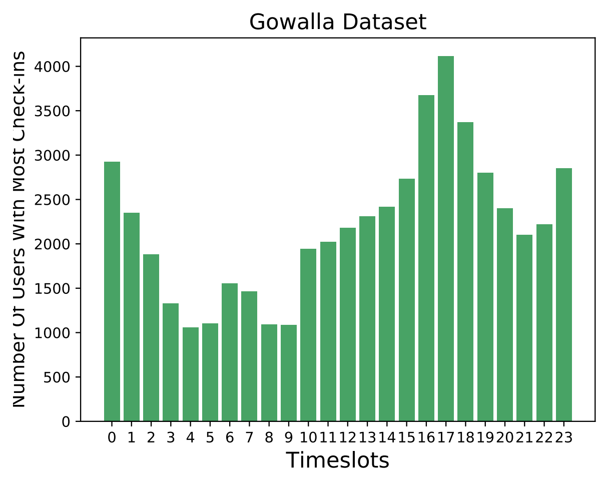


Fig. 2. The timeslot distribution of most check-ins for users in Gowalla dataset.

techniques such as Recurrent Neural Networks (RNNs) and their variants like Gated Recurrent Unit (GRU) [15] and Long Short-Term Memory (LSTM) [16] to model user mobility traces [17]. To deal with sparse or incomplete mobility traces, existing works strive to incorporate spatiotemporal factors into RNN architectures [18]. For example, Huang et al. [5] have fed the spatiotemporal contextual information into an attention-based LSTM network. Yang et al. [19] have used spatiotemporal contexts of POIs to search past hidden states with a RNN architecture.

In order to alleviate the data sparsity problem, some researchers incorporate the contextual information into the recommendation method [20]. For example, IRenMF [21] recommends new POIs close to the POIs visited before based on geographical neighborhood characteristics considering item collaborative signals. Specifically, IRenMF constructs a label matrix to assist recommendations by adding the weighted sum of neighboring POIs' ratings to each POI. Zhao et al. [7] proposed a two-stage framework to recommend locations for meteorological observation stations leveraging the factors of functions of architecture and building cost from multi-source urban big data. However, despite their promising results, we argue that they still suffer from two major limitations:

1) **Limitation 1: Lack of considering time-aware user embedding.** Most existing works on next POI recommendation only use descriptive features, such as ID or attributes, to build embedding function without considering internal interactions of the entities. These interactions are only used to define the objective function of model training [19]. In other words, although these methods have introduced a lot of additional information (geography, time, category, etc) accumulated in LBSN to assist recommendation, the correlation between these entities is ignored in the embedding stage, only obtaining suboptimal embedding representations. To alleviate this issue, some recent methods attempt to inject user-POI interactions [22], [23] or geographical influence [24] to capture highly non-linear representations of terms. While existing methods overlook the users' preferences of check-in times when modeling users, we argue that it is an essential factor to obtain sufficient user embeddings. For example, as illustrated in Figure 1, the first user prefers to check-in in the afternoons and evenings (15:00-0:00). In contrast, the second user prefers to check-in in the mornings (5:00-12:00). As a preliminary study, we summarized the distribution of users' check-in times (presented in hour-long timeslots) in the Gowalla dataset [19]. As shown in Figure 2, not all users have the same preferences of check-in times. Approximately 3,000 users prefer to check-in around midnight, while only around 1,000 users prefer to check-in in the early morning (e.g., 4:00 or 5:00). This observation further suggests the significance of taking users' time preferences into account in user embedding.
2) **Limitation 2: Lack of considering inherent relationships between POIs and check-in times.** We observed that different POIs have inherent relationships with their usual check-in times. For instance, restaurants may have more check-ins during lunchtime and dinner time, while most of the nightclubs' check-ins happen late in the evening. Most existing methods ignore the inherent patterns between POIs and check-in times. A recent method [25] predicts the time interval between successive check-ins. Our empirical analysis shows that the average time between successive check-ins is more than two days (cf. Table I). As check-in data is very sparse, it is challenging to capture user preferences accurately in this way. We argue that taking the patterns between POIs and check-in times into account can reduce the search space so as to overcome data sparsity problem and produce better POI recommendations.

To address the above limitations, in this paper, we propose a POI recommendation method with a cross-graph neural network in a multi-task learning manner. Specifically, we design a cross-graph neural network to jointly learn user embeddings in different semantic relations (user-POI interaction graph and user-timeslot graph) and automatically control how much information flows across two semantic spaces (**to address Limitation 1**). In addition, we exploit a multi-perspective self-attention to adaptively learn comprehensive preferences of user mobility traces. Finally, when we predict the next POI, we also predict the check-in timeslot according to users' check-in timeslot sequences, and learn internal association patterns between POIs and timeslots, aiming to reduce the search space to overcome the data sparsity problem (**to address Limitation 2**).

The main contributions of this paper can be summarized as follows:

- We design a cross-graph neural network to learn time-aware user embeddings in different semantic relations (user-POI interactions and user-timeslot interactions) and automatically control how much information flows across two semantic spaces by sharing cross-graph information.

- We design a check-in timeslot prediction task to predict the check-in time and learn the inherent association patterns between POIs and check-in timeslots as an auxiliary of the next POI recommendation.
- We conduct comprehensive experiments on two real-world public datasets, which demonstrate that our method can outperform various existing representative methods with at least 8.93% to 20.21% improvement on Recall@{1, 5, 10} and at least 9.25% to 17.56% improvement on MRR.

II. Related Work

Many researchers have invested many efforts on the task of POI recommendation [5]–[7], [26]. Generally speaking, existing works can be divided into two categories: conventional POI recommendation and successive POI recommendation.

A. Conventional POI Recommendation

Conventional POI recommendation mainly considers various contextual information including geographical distance [26], temporal information [14], [27] and social influence [28], [29] to estimate similarity for location prediction. Zhang et al. [26] proposed a geographical location recommendation method utilizing geographical influence, called iGeoRec. They adopted a probabilistic method to personalize the geographical influence as a personal distribution for each user and predict the probability of a user visiting any new location by this distribution. Yang et al. [28] construct a location-based social graph and exploit the graph embedding technique to learn feature vectors for entities. Then they rank a POI according to its similarity with user and time in the feature space. Gao et al. [14] present a social-historical model to explore the user's check-in behavior in LBSNs, which integrates the social and historical effects and assesses the role of social correlation. Yuan et al. [30] define a time-aware POI recommendation problem and develop a collaborative recommendation model incorporating temporal information and geographical information. Xie et al. [31] propose a generic graph-based embedding model for POI recommendation. They jointly capture the sequential effect, geographical influence, temporal cyclic and semantic effect in a unified way by embedding four corresponding relational graphs (POI-POI, POI-Region, POI-Time, and POI-Word) into a shared low dimensional space.

B. Successive POI Recommendation

Unlike conventional POI recommendations, successive POI Recommendation focuses on capturing sequential patterns of user's recent check-ins attaching importance for location prediction. Markov chains based methods [32]–[34] assume that the next item is conditioned on only the previous item. They obtain recommendation results by successfully characterizing short-range item transition. Rendle et al. [32] present a factorized personalized Markov chain (FPMC) model that subsumes both a common Markov chain and the normal matrix factorization. In addition, they also introduce an adaption of the Bayesian Personalized Ranking (BPR) framework for sequential data. Zhang et al. [33] employ Personalized Ranking Metric Embedding (PRME) to extend FPMC by modeling user-location and location-location distance in two different vector spaces. Zhao et al. [35] utilize the personalized Markov chain into the historical user check-ins and take the user's movement constraint into account for POI recommendation.

Nowadays, many researchers attempt to utilize Recurrent Neural Networks (RNNs) to deal with successive POI recommendation, as it shows promising performance for extracting sequential information [5], [36]. In order to make full use of time and geographic information, many studies try to incorporate spatiotemporal factors into RNN architectures. For example, Liu et al. [18] extended RNN and propose a novel method called Spatial-Temporal Recurrent Neural Networks (STRNN). They model local temporal and spatial contexts in each layer with time-specific transition matrices and distance-specific transition matrices. Zhao et al. [37] proposed a new spatiotemporal gated network (STGN) by enhancing long-short term memory network (LSTM). They design spatiotemporal gates to capture the spatiotemporal relationships between successive check-ins. Kong et al. [38] combined spatial-temporal influence into LSTM and propose a Spatial-Temporal Long-Short Term Memory (ST-LSTM) model. Further, they utilize a hierarchical extension of the proposed model (named HST-LSTM) to model the contextual historic visit information in an encoder-decoder manner. Huang et al [5] presented an attention-based spatiotemporal LSTM network for next POI recommendation. They incorporated the spatiotemporal contextual information into the LSTM network at the each step and leveraged an attention mechanism to consider the different importances of POIs. Yang et al. [19] proposed a novel method, named Flashback, which exploits spatiotemporal contexts to weight historical hidden states. Then they use the weighted average of historical hidden states as the final representation of the user mobility trace for recommendation.

III. Problem Formulation

We denote the existence of a user's check-in by 0 and 1 for otherwise. Suppose we have L users $U = \{u_1, u_2, ..., u_L\}$, M POIs $P = \{p_1, p_2, ..., p_M\}$ and N timeslots $T = \{t_1, t_2, ... , t_N\}$. We define the user-POI interaction matrix and user-timeslot interaction matrix as $X^{L\times M} = \{x_{up} | u \in U, p \in P\}$ and $Y^{L\times N} = \{y_{ut} | u \in U, t \in T\}$, respectively, with a binary value at each entry. $x_{up} = 1$ presents an observed check-in of user u in POI p. $y_{ut} = 1$ indicates an observed check-in of user u in timeslot t. We denotes the user mobility traces as S, then the user u's check-in POI sequence can be defined as $S_{p,u} = \{p^u_1, p^u_2, ..., p^u_s \mid p^u_i \in P\}$, where s is the length of the sequence. Based on the above definition, the problem of POI recommendation can be defined as follows:

Input: User set U, POI set P, timeslot set T, user-POI interaction matrix X, user-timeslot interaction matrix Y and user mobility trace set S.

Output: The probability that user u will interact with next POI p and the probability that user u will check in at the timeslot t.

IV. Methodologies

In this section, we illustrate the details of our proposed method for next POI recommendation. The objective of our method is to predict the probability distribution of the user's next check-in by optimizing an objective function, which includes two terms, i.e., POI prediction error term and timeslot prediction error term. The overall architecture of our proposed model is illustrated in Figure 3.

A. Cross-Graph Neural Network Component

The cross-graph neural network component can transfers graph information across different semantic spaces (i.e., user-POI interaction graph and user-time interaction graph) through common entities. We argue that cross-graph information can automatically control how much information flow across different semantic spaces and is beneficial for personalized user modeling, which is overlooked by previous works. In this study, we split a day to 24 hours (e.g., 24 timeslots), then we aggregate cross-graph information (user-POI interaction graph and user-timeslot interaction graph) and mitigate the semantic gap to achieve time-aware user embeddings for different users.

Formally, give two interaction graphs, i.e., user-POI interaction graph $\mathcal{G}_{up}(V_{up}, E_{up})$ and user-timeslot interaction graph $\mathcal{G}_{ut}(V_{ut}, E_{ut})$. V_{up} denotes a set of nodes consisting of both user nodes and POI nodes, and E_{up} refers to a set of edges showing user-POI interactions. V_{ut} denotes a set of nodes, including both user nodes and timeslot nodes, and E_{ut} refers to a set of edges presenting user-timeslot interactions. The architecture of the two interaction graphs is shown in the second block of Figure 3.

We use $\mathcal{U}, \mathcal{P}, \mathcal{T}, \mathcal{D}$ to represent the matrices of initial user embeddings, POI embeddings, timeslot embeddings and distance embeddings, respectively, which has the same embedding size. A user's check-in is generally affected by the distance between two neighboring locations [39]. We calculate the spatial distance using the GPS coordinates of two neighboring POIs, and round the distance into integer (e.g., 3.15 $\rightarrow$ 3) same as in [40].

Inspired by [41], we only use the neighborhood aggregation to capture complicated topology and higher-order connectivity in LBSNs, as shown in the third block of Figure 3. We ignore the self-connections of nodes because the layer combination operation essentially captures the same effect. Firstly, after obtaining the user-POI interaction matrix X, the adjacency matrix of user-POI graph can be easily expressed as follows:

$$\mathbf{A}_{up} = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{X} \\ \hdashline \mathbf{X}^\top & \mathbf{0} \end{array}\right]. \tag{1}$$

To model latent features of user-POI interactions, we construct an embedding propagation layer updates a node embedding based on the aggregations of its neighbors, and recursively perform such neighborhood aggregation to capture high-order connectivities in linear time complexity. The neighborhood aggregation process from the current layer to the next layer can be defined as:

$$\left[\frac{\mathcal{U}_{(l_1)}}{\mathcal{P}_{(l_1)}}\right] = \hat{\mathbf{A}}_{up}\left[\frac{\mathcal{U}_{(l_1-1)}}{\mathcal{P}_{(l_1-1)}}\right] \tag{2}$$

$$\mathcal{U}_{up} = \sum_{i=0}^{l_1} \alpha_i^{up}\mathcal{U}_{(i)}, \qquad \mathcal{P}_{up} = \sum_{i=0}^{l_1} \alpha_i^{up}\mathcal{P}_{(i)} \tag{3}$$

where $\hat{\mathbf{A}}_{up} = \mathbf{D}_{up}^{-\frac{1}{2}}\mathbf{A}_{up}\mathbf{D}_{up}^{-\frac{1}{2}}$, $\mathbf{D}_{up}$ is the degree matrix of $\mathbf{A}_{up}$, l_1 is the number of layers in graph neural network. We combine features learned at each layer to achieve the final user feature matrix $\mathcal{U}_{up}$ and POI feature matrix $\mathcal{P}_{up}$ in user-POI interaction graph to avoid over-smoothing [42]. The α_i^{up} denotes the importance of the i-th layer feature in constituting the final embedding. Learning the layer combination coefficient is technically viable, such as using an attention network. However, we find that learning α_i on the training data does not improve the performance while increasing the model complexity. So we exploit the same importance to combine embeddings of each layer, e.g., $\alpha_i = \frac{1}{l_1+1}$.

Similar to user-POI interaction graph, the adjacency matrix of the user-timeslot graph can be expressed as follows:

$$\mathbf{A}_{ut} = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{Y} \\ \hdashline \mathbf{Y}^\top & \mathbf{0} \end{array}\right]. \tag{4}$$

Similar to the operation of equation 3-4, after propagation-based embeddings, the feature matrix of users, timelots are obtained as follows:

$$\left[\frac{\mathcal{U}_{(l_1)}}{\mathcal{T}_{(l_1)}}\right] = \hat{\mathbf{A}}_{ut}\left[\frac{\mathcal{U}_{(l_1-1)}}{\mathcal{T}_{(l_1-1)}}\right] \tag{5}$$

$$\mathcal{U}_{ut} = \sum_{i=0}^{l_1} \alpha_i^{ut}\mathcal{U}_{(i)}, \qquad \mathcal{T}_{ut} = \sum_{i=0}^{l_1} \alpha_i^{ut}\mathcal{T}_{(i)} \tag{6}$$

where $\hat{\mathbf{A}}_{ut} = \mathbf{D}_{ut}^{-\frac{1}{2}}\mathbf{A}_{ut}\mathbf{D}_{ut}^{-\frac{1}{2}}$, $\mathbf{D}_{ut}$ is the degree matrix of $\mathbf{A}_{ut}$. $\mathcal{U}_{ut}$ and $\mathcal{T}_{ut}$ are user feature matrix and timeslot feature matrix in user-timeslot interaction graph, respectively.

In the process of neighborhood aggregation, we do not introduce nonlinear activation function and feature transformation matrix, because these operations directly inherited from GCN [43], [44] will bring no benefits, but negatively increases the difficulty for model training on recommendation tasks [41]. Finally, the complexity of this part in our method is only equivalent to the standard matrix factorization (MF).

The previous studies pay little attention on the fusion of user presentations under different semantic spaces. Automatically controlling information flow across different semantic spaces is meaningful for modeling personalized representations with different relationships. We use a gate mechanism to control how much information flows across two semantic spaces and update user embeddings:

$$\mathcal{U}^* = gate(\mathcal{U}_{up}, \mathcal{U}_{ut}) = g * \mathcal{U}_{up} + (1-g) * \mathcal{U}_{ut}) \tag{7}$$

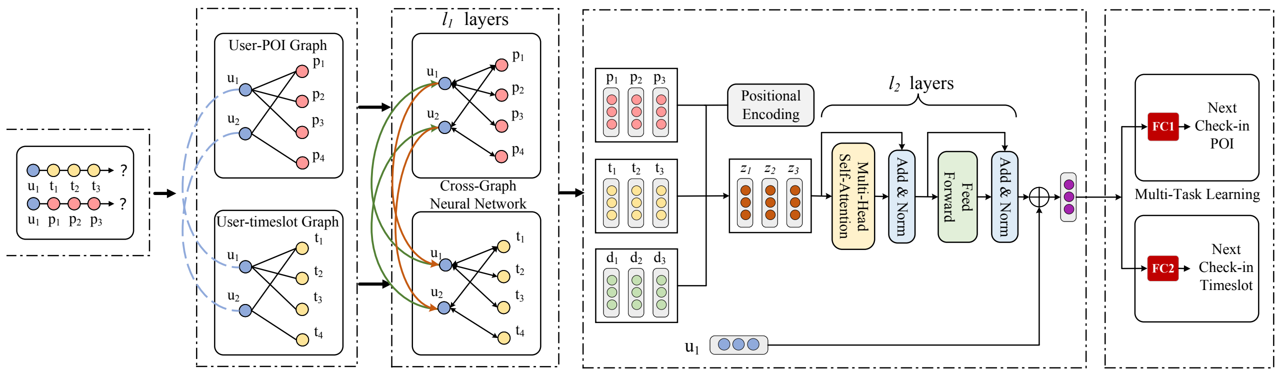


Fig. 3. A sample illustration of our proposed model. Blue nodes are user nodes, pink nodes are POI nodes, and yellow nodes are timeslot nodes. $\oplus$ denotes the concatenation operation.

where $\mathcal{U}_{up}$ and $\mathcal{U}_{ut}$ are feature matrices of users in $\mathcal{G}_{up}$ and $\mathcal{G}_{ut}$ after l_1-th layer propagation. g is the gate mechanism, which is computed as:

$$g = \sigma(W_g(\text{Concat}(\mathcal{U}_{up}, \mathcal{U}_{ut})) + b_g) \tag{8}$$

where σ is the sigmoid activation function constraining the output values within the range of 0 to 1, W_g, b_g, are trainable parameters.

After the above operations, the final user $\mathcal{U}^*$, POI $\mathcal{P}_{up}$ and timeslot $\mathcal{T}_{ut}$ embedding matrices are obtained, which can be passed to the downstream layers.

B. Multi-perspective Self-Attention Component

In this section, we introduce the Transformer [45] encoder to replace the RNN architecture that is commonly used in many existing works shown in the forth block of Figure 3. we utilize the self-attention to consider multi-perspective relations of check-ins. The embeddings of each entity in the check-ins is the sum of corresponding POI, timeslot, distance and position embeddings:

$$\mathbf{e}_i = \mathbf{p}_i + \mathbf{t}_i + \mathbf{d}_i + \mathbf{pos}_i \tag{9}$$

where $\mathbf{p}_i \in \mathcal{P}_{up}$ is the embedding of POI p_i, $\mathbf{t}_i \in \mathcal{T}_{ut}$ is the embedding of timeslot t_i, $\mathbf{d}_i \in \mathcal{D}$ is the embedding of distance d_i between p_{i-1} and p_i, **pos** denotes the position embedding of the POI i in the check-in sequence. We use $\mathbf{e}_i$ to denote the updated embedding of POI p_i, which will be acted on downstream layers.

Then we adopt l_2-layer Transformer [45] encoder to encode the embedding matrix of the check-in sequence into contextual representations $H_l = \text{Transformer}_l(H_{l-1}), l \in [1, l_2]$. The detail of each Transformer layer is defined by:

$$\mathcal{Z}_l = \text{LN}(\text{MHA}(H_{l-1}) + H_{l-1}) \tag{10}$$

$$H_l = \text{LN}(\text{FFN}(\mathcal{Z}_l) + \mathcal{Z}_l) \tag{11}$$

where MHA denotes a multi-headed self-attention mechanism, LN representd a layer normalization, and FFN indicates a two-layer feed forward network. S_l denotes the output of the l-th Transformer. When encoding a POI, we need to consider the influence of other POIs in the sequence on it. The propagation rule of a multi-headed self-attention in the l-th layer Transformer is defined as follows:

$$Q_i = H_{l-1}W_i^Q, \quad K_i = H_{l-1}W_i^K, \quad V_i = H_{l-1}W_i^V \tag{12}$$

$$head_i = \text{Softmax}(\frac{Q_i K_i^{\mathrm{T}}}{\sqrt{d_k}})V_i \tag{13}$$

$$\check{\mathcal{Z}}_l = \text{Concat}(head_1, head_2, ..., head_n)W_l^O \tag{14}$$

where the output of the previous layer $H_{l-1} \in \mathbb{R}^{|\boldsymbol{S}| \times d_h}$ is linearly projected to query (Q), key (K) and query (Q) matrices via three weight matrices $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_h \times d_k}$, respectively. d_h is the hidden size, d_k is the dimension of a head, n indicates the number of heads. After concatenating the feature vectors of all heads, the output of a multi-headed self-attention $\check{\mathcal{Z}}_l$ in the l-th layer Transformer is achieved. $W_l^O \in \mathbb{R}^{d_h \times d_h}$ is a learnable weight matrix.

Finally, we use the average representation $\bar{\mathcal{Z}}$ of all POIs in the same check-in sequence to represent the entire sequence.

C. Multi-Task Learning Component

Different locations have inherent patterns with different timeslots. In addition, users' check-in sequence also contains users' activities in time perspective, while existing methods ignore the explicit modeling of check-in timeslots. Due to the sparse user mobility traces, we predict the timeslot (a day can be divided into 24 timeslots) of the next check-in rather than the time interval.

We concatenate the vector of check-in sequence with the vector of the user. Then we put them into two fully connected layers:

$$\hat{Y}_p = \rho(FC_1(\text{Concat}(\mathcal{U}_u^*, \bar{\mathcal{Z}}))) \tag{15}$$

$$\hat{Y}_t = \rho(FC_2(\text{Concat}(\mathcal{U}_u^*, \bar{\mathcal{Z}}))) \tag{16}$$

where FC_1 and FC_2 are two fully connected layers, containing M and 24 units, respectively (because the number of POIs is M and there are 24 hours in a day). $\hat{Y}_p$ and $\hat{Y}_t$ denote the prediction for the next POI and check-in timeslot, respectively, ρ denotes a softmax function, $\mathcal{U}_u^*$ represents the feature vector of user u. The loss function of the next POI prediction task is defined as the cross-entropy error over all labeled user check-in POI sequences:

$$\mathcal{L}_p = -\sum_{k}^{C_{s,p}} \sum_{f}^{F_1} Y_{p,kf} \ \ln \hat{Y}_{p,kf} \tag{17}$$

where $C_{s,p}$ is the set of check-in POI sequence indices that have POI labels, and F_1 is the dimension of the output features in FC_1, which is equal to the number of POIs (M). Y_p is the POI label indicator matrix.

Similar to the next POI prediction task, we define the loss function of the next check-in timeslot task as follows:

$$\mathcal{L}_t = -\sum_{k}^{C_{s,t}} \sum_{f}^{F_2} Y_{t,kf} \ \ln \hat{Y}_{t,kf} \tag{18}$$

where $C_{s,t}$ is the set of check-in timeslot indices that have labels for check-in timeslots, and $F_2 = 24$ is the dimension of the output features in FC_2, which is equal to the number of timeslots. Y_t is the label indicator matrix for check-in timeslots.

The final loss function for the multi-task learning is the sum of two prediction tasks:

$$\mathcal{L}_{loss} = \mathcal{L}_p + \mathcal{L}_t + \lambda \|\Theta\|^2 \tag{19}$$

where Θ contains all trainable parameters in the model. λ is the L_2 regularization coefficient, which is used to prevent overfitting.

TABLE I
STATISTICS OF DATASETS

Dataset	Gowalla	Foursquare
Number of users	52,979	46,065
Number of POIs	121,851	69,005
Number of check-ins	3,300,986	9,450,342
Density	0.0511%	0.2973%
Average time between successive check-ins	51.28 hours (2.13 days)	58.59 hours (2.44 days)
Collection period	2009/02-2010/10	2012/04-2014/01

V. EXPERIMENTS

In this section, we conduct comprehensive experiments on two real-world public datasets to evaluate our proposed method for the next POI recommendation over sparse user mobility traces.

A. Dataset

We evaluate our proposed method on two widely used check-in datasets [19] collected from two popular location-based social networks: Gowalla and Foursquare, respectively. Some basic statistics of the dataset are demonstrated in Table I. We split all user mobility traces into 80% for training and 20% for testing chronologically. For each sample, we truncate sequences of the same length (20 by default). In each check-in POI sequence for a user u, $S_{p,u} = \{p_1^u, p_2^u, ..., p_s^u \mid p_i^u \in P\}$, where s is the sequence length, the next POI is used as a label. Similar to check-in POI sequences, each corresponding check-in timeslot sequence is also processed in a consistent manner. For fair comparison, we keep the same data preprocessing and experimental settings as [19].

B. Methods for Comparison

Currently, there are many methods based on different technical principles for the next POI recommendation. For this reason, we select some representative methods from 4 categories for comparison to demonstrate the effectiveness of our proposed method.

MF-based Methods:

- **WRMF** [46]: This method utilizes matrix factorization to learn user preferences on POIs.
- **BPR** [29]: It learns user preferences on POIs by optimizing a pairwise ranking loss.

Feature-based Methods:

- **MFT** [14]: This method ranks POIs according to users' historical check-in counts and timeslots on POIs.
- **LBSN2Vec** [28]: It exploits the graph embedding technique to learn feature vectors of the user, time, and POI, and ranks a POI through estimating its similarity with user and time.

Markov-Chain-based Methods:

- **FPMC** [32]: It includes a common Markov chain and a normal matrix factorization model that factorizes the tensor of transition cube.
- **PRME** [33]: It captures personalized POI transition patterns through learning user and POI embeddings.
- **TribeFlow** [34]: It captures the transition matrix in an implicit space by utilizing a semi-Markov chain model.

Spatiotemporal RNNs:

- **STRNN** [18]: It extends RNN and models local temporal and spatial contexts in each layer with time-specific transition matrices and distance-specific transition matrices.
- **STGN** [37]: It enhances LSTM by introducing spatiotemporal gates to capture the spatiotemporal relationships between successive check-ins.
- **Flashback** [19]: This method explicitly exploits spatiotemporal contexts to search past hidden states for the next POI recommendation.

C. Evaluation Metrics

We evaluate our method versus other methods in terms of two widely used metrics: Recall@N (R@N, $N \in \{1, 5, 10\}$) and MRR. For each user, Recall@N indicates what percentage of her interacted next POI can emerge in the top-N recommended POIs. MRR is the Mean Reciprocal Rank, which takes the position of the first correctly recommended POIs into account.

TABLE II
RECALL AND MRR OF OUR METHOD COMPARED WITH VARIOUS REPRESENTATIVE METHODS.

Methods	Gowalla				Foursquare			
	R@1	**R@5**	**R@10**	**MRR**	**R@1**	**R@5**	**R@10**	**MRR**
WRMF	0.0112	0.0260	0.0367	0.0178	0.0278	0.0619	0.0821	0.0427
BPR	0.0131	0.0363	0.0539	0.0235	0.0315	0.0828	0.1143	0.0538
MFT	0.0525	0.0948	0.1052	0.0717	0.1945	0.2692	0.2788	0.2285
LBSN2Vec	0.0864	0.1186	0.1390	0.1032	0.2190	0.3955	0.4621	0.2781
FPMC	0.0479	0.1668	0.2411	0.1126	0.0753	0.2384	0.3348	0.1578
PRME	0.0740	0.2146	0.2899	0.1503	0.0982	0.3167	0.4064	0.2040
TribeFlow	0.0256	0.0723	0.1143	0.0583	0.0297	0.0832	0.1239	0.0645
STRNN	0.0900	0.2120	0.2730	0.1508	0.2290	0.4310	0.5050	0.3248
STGN	0.0624	0.1586	0.2104	0.1125	0.2094	0.4734	0.5470	0.3283
Flashback	0.1158	0.2754	0.3479	0.1925	0.2496	0.5399	0.6236	0.3805
Ours	**0.1321**	**0.3306**	**0.4182**	**0.2263**	**0.2743**	**0.5982**	**0.6793**	**0.4157**
Improve	14.08%	20.04%	20.21%	17.56%	9.90%	10.80%	8.93%	9.25%

D. Experimental Settings

We use an Adam optimizer [47] to optimize the parameters in the model. Since there are so many hyper-parameters, we can not analyze all these hyper-parameters, so we list the value ranges of some common hyperparameters. The learning rate is searched in {1e-2, 5e-3, 1e-3, 5e-4, 1e-4}. The dropout rate from $\{0.1, 0.2, 0.3\}$ and the L_2 regularization coefficient from {1e-4, 1e-5, 1e-6, 1e-7, 1e-8} are used to prevent overfitting. l_1 and l_2 are searched in {1, 2, 3, 4}. We set the same dimension searched in {16, 32, 48, 64, 80, 96, 112, 128} for embeddings.

Finally, the learning rate is set to 0.001, the batch size is set to 200 on Gowalla dataset and 350 on Foursquare dataset, the λ is set to 1e-7, the embedding size is set to 80 on Gowalla dataset and 48 on Foursquare dataset, the dropout rate is set to 0.2, the number of layers is set to $l_1 = 3$ for cross-graph neural network and $l_2 = 3$ for Transformer encoder, the number of self-attention heads is set to $n = 4$. The experiments are run on a GPU of Nvidia RTX 2080 Ti. The algorithms are implemented by leveraging the deep learning library PyTorch[4].

E. Experimental Results

The comparison results of our method and other representative methods are shown in Table II. The optimal results are marked in bold and the underlined values are the suboptimal results.

STRNN and STGN outperform MFT and FPMC, which demonstrates the stronger ability of the RNN architectures in capturing sequential patterns of user mobility traces. LSBN2Vec captures relevant information in location-based social networks through graph embedding technique. It achieves the best recommendation accuracy in all non-RNN based methods, reflecting the importance of side information for the next POI recommendation. However, they fail to go beyond the Flashback model, which takes advantage of searching past hidden states from the spatiotemporal perspective to further enhance the next POI recommendation. Finally, we observe that our method consistently and significantly outperforms all representative methods. Specifically, compared with the second-best performing model (Flashback), Our method shows an improvement of 17.56% and 9.25% on MRR, in Gowalla and Foursquare datasets, respectively.

[4]https://pytorch.org/

Our method outperforms all these methods for three major reasons. Firstly, we design a cross-graph neural network to integrate user-POI preferences with user-timeslot preferences in the same semantic space by sharing cross-graph information and automatically controlling the propagation of information flow. Secondly, our method can adaptively capture comprehensive preferences of user check-ins through a multi-perspective self-attention component. Thirdly, our method introduces an additional check-in timeslot prediction task, which helps predict users' check-in time, and learn inherent association patterns between POIs and check-in timeslots, as the effective auxiliary of next POI recommendation.

In addition, we also observe that the experimental results reported on Foursquare dataset are better than the results on Gowalla dataset. The major reason is that Gowalla dataset is more sparse, and the data sparsity declines the recommendation accuracy.

F. Ablation Study

In order to understand how each component contributes to the recommendation accuracy of our proposed method, including personalized time-aware user modeling with cross-graph neural network, multi-perspective self-attention, and check-in timeslot prediction task, we perform an ablation analysis in Table III. In (1), we adopt the GCN to learn the user embeddings and POI embeddings through **U**ser-**P**OI interactions (**UP**). In (2), we adopt the GCN to learn embeddings of users, POIs and timeslots through user-POI graph and **U**ser-**T**imeslot graph (**UT**), and use the sum of learned user embeddings as the final user embeddings. In (3), we design a **C**ross-graph (**C**) neural network to control how much information flows across two graphs. In (4), we introduce a **M**ulti-**P**erspective (**MP**) self-attention module to explore comprehensive preferences of users. In (5-8), we incorporate the **M**ulti-**T**ask (**MT**) learning into (4) with a different number of layers in graph neural network and Transformer encoder an the same time.

TABLE III
THE ABLATION ANALYSIS ON GOWALLA AND FOURSQUARE DATASETS.

Architecture	Gowalla		Foursquare	
	R@5	MRR	R@5	MRR
(1) UP	0.2360	0.1536	0.4809	0.3221
(2) UP-UT	0.2625	0.1769	0.5205	0.3588
(3) C-UP-UT	0.2774	0.1828	0.5366	0.3691
(4) C-UP-UT-MP	0.2980	0.2087	0.5604	0.3886
(5) C-UP-UT-MP-MT ($l = 1$)	0.3290	0.2249	0.5889	0.4067
(6) C-UP-UT-MP-MT ($l = 2$)	0.3287	0.2239	0.5878	0.4034
(7) C-UP-UT-MP-MT ($l = 3$)	**0.3306**	**0.2263**	**0.5982**	**0.4157**
(8) C-UP-UT-MP-MT ($l = 4$)	0.3300	0.2254	0.5943	0.4109

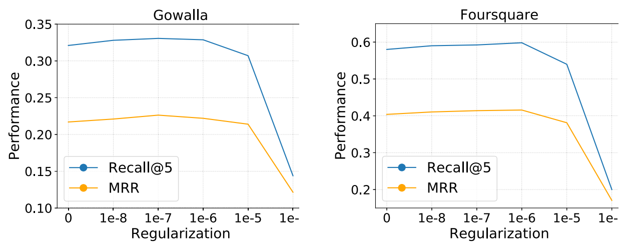
From the results shown in Table III, we have some observations. In (1), GCN shows the great ability in feature learning only with user-POI interactions, which has been proved in previous works [41], [48]. In (2), by integrating user-timeslot interactions into user modeling, the performance obtains further improvements (e.g., UP-UT vs. UP), which indicates the time information plays a critical role in user modeling. In (3), we design a cross-graph neural network to jointly learn user representations in different semantic spaces and automatically control how much information flows across two semantic spaces for personalized user modeling. The performance is further improved than simply adding user representations. In (4), we try to capture multi-perspective preferences of user interests, which achieves better results. In (5-8), by adding check-in timeslot prediction task into the model, our method obtains large performance gains on both datasets, which indicates the users' activities of time perspective and the internal association patterns between timeslots and POIs are conducive for better POI recommendations. Explicitly modeling the check-in timeslot of the user can provide a significant supplementary for better next POI recommendation.

G. Hyper-parameter Study

We analyze the effect of two hyper-parameters: the regularization coefficient λ and the embedding size d.

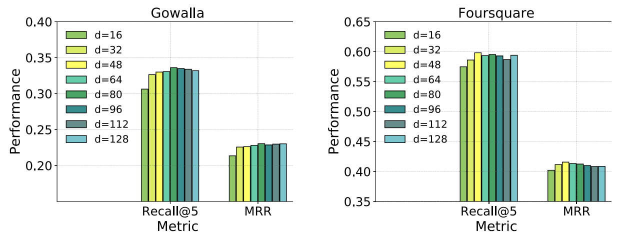
1) Impact of Regularization Coefficient λ: The impacts of λ on Gowalla and Foursquare datasets are shown in Figure 4. We can observe that our method is relatively insensitive to λ. Even when λ is set to 0, our method can still perform very well, which shows that our method is less prone to overfitting and easy to train and regularize. The optimal values for Gowalla and Foursquare are $1e-7$ and $1e-6$, respectively. When λ is larger than $1e-5$, the performance drops quickly, which indicates that too strong regularization will have a negative impact on the training of the model.

2) Impact of Embedding Size d: We also study the impact of the embedding size of features. From Figure 5, we can observe that a small dimension of embedding size is not sufficient to express the latent features of users and POIs. By increasing the dimension of embedding size, the model has more capacity to model the complex features. The recommendation accuracy first improves and then becomes steady when the embedding size is searched in $\{16, 32, 48, 64, 80, 96, 112, 128\}$. The optimal settings of embedding size on Gowalla and Foursquare are close to 80 and 48, respectively.



(a) (b)

Fig. 4. Impact of regularization coefficient λ on *Gowalla* and *Foursquare* datasets.



(a) (b)

Fig. 5. Impact of hidden size d on *Gowalla* and *Foursquare* datasets.

H. Training Efficiency

Flashback is the sub-optimal method, which outperforms STRNN, STGN, LBSN2Vec, PRME, and other baseline methods. Therefore, we evaluate and compare the training efficiency of our method with Flashback in terms of the training speed (time taken for one epoch of training). To make a fair comparison, we set the same hyper-parameters, especially the batch size and sequence length. All experiments are conducted on a GPU of Nvidia RTX 2080 Ti. Two methods are executed 20 epochs, and we report the average training cost, which is shown in Table IV.

TABLE IV
THE TRAINING TIME PER EPOCH COMPARISON ON TWO DATASETS IN TERMS OF SECONDS

	Gowalla	Foursquare
Flashback	89.702s	263.383s
Ours	**48.472s**	**147.201s**
Improve	85.06%	78.93%

From the results in Table IV, we can observe that our method yields the fastest training speed on all datasets. Flashback assigns weights to all past hidden states based on spatiotemporal factors, increasing the accuracy of recommendation while bringing substantial computational overhead. The time complexity of this operation is $O(N^2)$, according to the original code provided by [19]. In all, compared with Flashback, our proposed method can significantly reduce training costs while improving the recommendation accuracy. Specifically, the gains of training efficiency reach about 85.06% and 78.93% on Gowalla and Foursquare datasets, respectively.

VI. Discussion

A. Impact on industry and academia

From the industry's perspective, high-precision POI recommendations can help develop smart city applications providing users with interesting tour route guidance, which helps to enhance user engagement and promote consumption. Our method can generate high-precision POI recommendations with low computational overhead, providing a brand-new solution for the industry to provide users with personalized location recommendation services.

From academia's perspective, POI recommendation is a trending and essential research topic, with the rapid growth of LBSNs. User mobility traces accumulated in online platforms offer valuable opportunities for understanding human dynamics. Our method provides a new exploration, especially for learning time-aware user modeling, exploring users' activities in time perspective, and mining internal association patterns between POIs and timeslots.

B. Threats to Validity

Threats of internal validity are related to errors in our implementation and personal bias in the data processing. To avoid implementation errors, we have carefully checked our experiment steps and parameter settings. Specifically, we conduct an ablation study to verify the effectiveness of several key designs in our method. We also conduct a hyper-parameter study to analyze the effect of two important hyper-parameters, and other hyper-parameters are selected by using a grid search. To avoid the personal biases of data processing, we implement the same data processing as [19].

Threats of external validity are related to our experimental datasets' quality and the generalizability of our experiment results and findings. To guarantee the quality of our datasets, we have adopted two widely used check-in datasets collected from two popular LBSNs, e.g., Gowalla and Foursquare. In order to guarantee the generalizability of our experiment results and findings, We have selected ten representative methods from four categories to compare with our method. Through verifying the results reported by our method, we conclude that our method can obtain better POI recommendation accuracy than all other methods, 33.06% on Recall@5 and 22.63% on MRR in Gowalla dataset, and 59.82% on Recall@5 and 41.57% on MRR in Foursquare dataset. In comparing training efficiency, we conduct the experiments on the same Linux system with a single GPU of Nvidia RTX 2080 Ti.

VII. Conclusion and Future Work

This paper presented a novel POI recommendation method which includes a cross-graph neural network component, a multi-perspective self-attention component, and a multi-task learning component. Specifically, we use the cross-graph neural network to jointly learn time-aware user embeddings and control how much information flows across different semantic spaces, which makes up the inadequate representation of existing user modeling methods. Then we utilize a multi-perspective self-attention component to capture comprehensive preferences of users. Finally, we design a check-in time prediction task to learn users' activities from a time perspective and learn internal patterns between POIs and their check-in times, aiming to reduce the search space to overcome the data sparsity problem. Comprehensive experimental results in comparisons with representative methods on two real-world public datasets have successfully demonstrated that our method can achieve better recommendation accuracy with much less training costs.

As part of our future work, we plan to consider time-aware POI-POI complementarities to generate POI sequences with specific timestamps to provide users with personalized tour arrangements.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants 61972290 and the National Key R&D Program of China under Grant 2018YFC1604000. This work is also supported by the National Natural Science Foundation of China (61962061) and partially supported by the Yunnan Provincial Foundation for Leaders of Disciplines in Science and Technology (202005AC160005).

References

[1] K. Zhai, B. Jiang, and W. K. Chan, "Prioritizing test cases for regression testing of location-based services: Metrics, techniques, and case study," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 54–67, 2014.

[2] D. Zhang, A. Liu, G. Jin, and Q. Li, "Edge-based shortest path caching for location-based services," in *2019 IEEE International Conference on Web Services (ICWS)*, 2019, pp. 320–327.

[3] C. Xu, L. Zhu, Y. Liu, J. Guan, and S. Yu, "Dp-ltod: Differential privacy latent trajectory community discovering services over location-based social networks," *IEEE Transactions on Services Computing*, pp. 1–1, 2019.

[4] T. Koskela, S. Järvinen, M. Liu, and M. Ylianttila, "User experience in added value location-based mobile music service," in *2010 IEEE International Conference on Web Services*, 2010, pp. 465–472.

[5] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal lstm network for next poi recommendation," *IEEE Transactions on Services Computing*, pp. 1–1, 2019.

[6] Y. Gao, X. Gao, X. Li, B. Yao, and G. Chen, "An embedded grasp-vns based two-layer framework for tour recommendation," *IEEE Transactions on Services Computing*, pp. 1–1, 2020.

[7] G. Zhao, T. Liu, X. Qian, T. Hou, H. Wang, X. Hou, and Z. Li, "Location recommendation for enterprises by multi-source urban big data analysis," *IEEE Transactions on Services Computing*, vol. 13, pp. 1–1, 2020.

[8] X. Wang, X. Liu, J. Liu, X. Chen, and H. Wu, "A novel knowledge graph embedding based api recommendation method for mashup development," *World Wide Web*, vol. 24, no. 3, pp. 869–894, 2021.

[9] Y. Si, F. Zhang, and W. Liu, "Ctf-ara: An adaptive method for poi recommendation based on check-in and temporal features," *Knowledge-Based Systems*, vol. 128, pp. 59–70, 2017.

[10] C. Comito, "Next: a framework for next-place prediction on location based social networks," *Knowledge-Based Systems*, vol. 204, p. 106205, 2020.

[11] G. Zhao, P. Lou, X. Qian, and X. Hou, "Personalized location recommendation by fusing sentimental and spatial context," *Knowledge-Based Systems*, p. 105849, 2020.

[12] Y. Jiang, W. He, L. Cui, and Q. Yang, "User location prediction in mobile crowdsourcing services," in *International Conference on Service-Oriented Computing*, 2018, pp. 515–523.

[13] J.-D. Zhang and C.-Y. Chow, “Enabling probabilistic differential privacy protection for location recommendations,” *IEEE Transactions on Services Computing*, 2018.

[14] H. Gao, J. Tang, and H. Liu, “Exploring social-historical ties on location-based social networks,” in *6th International AAAI Conference on Weblogs and Social Media, ICWSM 2012*, 2012, pp. 114–121.

[15] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *EMNLP’2014 Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[16] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] K. Sun, T. Qian, T. Chen, Y. Liang, H. N. Q. Viet, and H. Yin, “Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation,” in *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, 2020, pp. 214–221.

[18] Q. Liu, S. Wu, L. Wang, and T. Tan, “Predicting the next location: a recurrent model with spatial and temporal contexts,” in *AAAI’16 Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 194–200.

[19] D. Yang, B. Fankhauser, P. Rosso, and P. Cudre-Mauroux, “Location prediction over sparse user mobility traces using rnns: Flashback in hidden states!” in *IJCAI 2020: International Joint Conference on Artificial Intelligence*, 2020, pp. 2184–2190.

[20] F. Zhu, Y. Wang, C. Chen, G. Liu, and X. Zheng, “A graphical and attentional framework for dual-target cross-domain recommendation,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, vol. 3, 2020, pp. 3001–3008.

[21] Y. Liu, W. Wei, A. Sun, and C. Miao, “Exploiting geographical neighborhood characteristics for location recommendation,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 739–748.

[22] H. Xu, J. Wei, Z. Yang, and J. Wang, “Graph attentive network for region recommendation with poi- and roi-level attention,” in *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, 2020, pp. 509–516.

[23] B. Chang, G. Jang, S. Kim, and J. Kang, “Learning graph-based geographical latent representation for point-of-interest recommendation.” in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2020, pp. 135–144.

[24] K. Yang and J. Zhu, “Next poi recommendation via graph embedding representation from h-deepwalk on hybrid network,” *IEEE Access*, vol. 7, pp. 171 105–171 113, 2019.

[25] J. Zhong, C. Ma, J. Zhou, and W. Wang, “From when to where: A multi-task learning approach for next point-of-interest recommendation.” in *WASA (1)*, 2020, pp. 781–793.

[26] J.-D. Zhang, C.-Y. ChowMember, and Y. Li, “igeorec: A personalized and efficient geographical location recommendation framework,” *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 701–714, 2015.

[27] Y. Liu, L. Zhao, G. Liu, X. Lu, P. Gao, X. Li, and Z. Jin, “Dynamic bayesian logistic matrix factorization for recommendation with implicit feedback,” in *IJCAI 2018: 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3463–3469.

[28] D. Yang, B. Qu, J. Yang, and P. Cudre-Mauroux, “Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach,” in *The World Wide Web Conference on*, 2019, pp. 2147–2157.

[29] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidtthieme, “Bpr: Bayesian personalized ranking from implicit feedback,” in *Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[30] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, “Time-aware point-of-interest recommendation,” in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, pp. 363–372.

[31] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, “Learning graph-based poi embedding for location-based recommendation,” in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 15–24.

[32] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized markov chains for next-basket recommendation,” in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.

[33] J.-D. Zhang, C.-Y. Chow, and Y. Li, “Lore: exploiting sequential influence for location recommendations,” in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014, pp. 103–112.

[34] F. Figueiredo, B. Ribeiro, J. M. Almeida, and C. Faloutsos, “Tribeflow: Mining & predicting user trajectories,” in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 695–706.

[35] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, “Stellar: spatial-temporal latent ranking for successive point-of-interest recommendation,” in *AAAI’16 Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 315–321.

[36] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu, “Sequential click prediction for sponsored search with recurrent neural networks,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI’14, 2014, p. 1369–1375.

[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,” *AAAI 2019 : Thirty-Third AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 5877–5884, 2019.

[38] D. Kong and F. Wu, “Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction,” in *IJCAI 2018: 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 2341–2347.

[39] S. Feng, X. Li, Y. Zeng, G. Cong, and Y. M. Chee, “Personalized ranking metric embedding for next new poi recommendation,” in *IJCAI’15 Proceedings of the 24th International Conference on Artificial Intelligence*. ACM, 2015, pp. 2069–2075.

[40] L. Zhang, Z. Sun, J. Zhang, Y. Lei, C. Li, Z. Wu, H. Kloeden, and F. Klanner, “An interactive multi-task learning framework for next poi recommendation with uncertain check-ins,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, vol. 4, 2020, pp. 3551–3557.

[41] X. He, K. Deng, X. Wang, Y. Li, Y.-D. Zhang, and M. Wang, “Lightgcn: Simplifying and powering graph convolution network for recommendation,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.

[42] J. Klicpera, A. Bojchevski, and S. Günnemann, “Predict then propagate: Graph neural networks meet personalized pagerank,” in *ICLR 2019 : 7th International Conference on Learning Representations*, 2019.

[43] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR 2017 : International Conference on Learning Representations 2017*, 2017.

[44] X. Wang, J. Liu, X. Liu, X. Cui, and H. Wu, “A novel dual-graph convolutional network based web service classification framework,” in *2020 IEEE International Conference on Web Services (ICWS)*, 2020, pp. 281–288.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[46] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 2008, pp. 263–272.

[47] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.

[48] S. Wu, Y. Zhang, C. Gao, K. Bian, and B. Cui, “Garg: Anonymous recommendation of point-of-interest in mobile networks by graph convolution network,” *Data Science and Engineering*, pp. 1–15, 2020.