

TAU: Trajectory Data Augmentation with Uncertainty for Next POI Recommendation

Zhuang Zhuang^{1, 2}, Tianxin Wei³, Lingbo Liu⁴, Heng Qi^{1, 2*}, Yanming Shen^{1, 2}, Baocai Yin^{1, 2}

¹Dalian University of Technology, China

²Key Laboratory of Social Computing and Cognitive Intelligence (Dalian University of Technology), Ministry of Education, Dalian, China

³University of Illinois Urbana Champaign, USA

⁴Pengcheng Laboratory, Shenzhen, China

zhuang97@mail.dlut.edu.cn, {hengqi, shen, ybc}@dlut.edu.cn, twei10@illinois.edu, lingbo.liu@polyu.edu.hk

Abstract

Next Point-of-Interest (POI) recommendation has been proven effective at utilizing sparse, intricate spatial-temporal trajectory data to recommend subsequent POIs to users. While existing methods commonly alleviate the problem of data sparsity by integrating spatial-temporal context information, POI category features, and social relationships, they largely overlook the fact that the trajectory sequences collected in the datasets are often incomplete. This oversight limits the model's potential to fully leverage historical context. In light of this background, we propose Trajectory Data Augmentation with Uncertainty (TAU) for Next POI Recommendation. TAU is a general graph-based trajectory data augmentation method designed to complete user mobility patterns by marrying uncertainty estimation into the next POI recommendation task. More precisely, TAU taps into the global transition pattern graph to identify sets of intermediate nodes located between every pair of locations, effectively leveraging edge weights as transition probabilities. During trajectory sequence construction, TAU selectively prompts intermediate nodes, chosen based on their likelihood of occurrence as pseudo-labels, to establish comprehensive trajectory sequences. Furthermore, to gauge the certainty and impact of pseudo-labels on the target location, we introduce a novel confidence-aware calibration strategy using evidence deep learning (EDL) for improved performance and reliability. The experimental results clearly indicate that our TAU method achieves consistent performance improvements over existing techniques across two real-world datasets, verifying its effectiveness as the state-of-the-art approach to the task.

Introduction

With the rapid growth of the mobile internet, individuals are increasingly sharing their experiences and insights on Location-Based Social Network (LBSN) platforms, such as Gowalla, Foursquare, or Yelp. This phenomenon has become a pervasive trend, leading to the accumulation of a substantial volume of spatial-temporal updates (e.g., check-in records). These updates serve as the foundation for predicting users' future movements (Gonzalez, Hidalgo, and

*Corresponding author: hengqi@dlut.edu.cn
Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

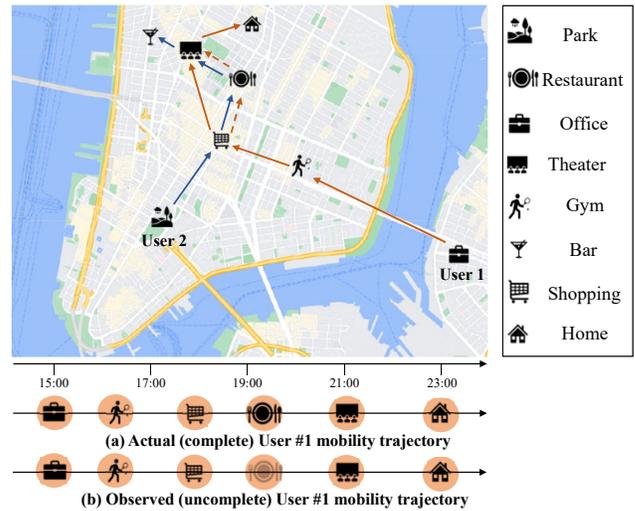


Figure 1: A simplified example demonstrating the operational principle of TAU from a temporal perspective. Solid lines indicate travel trajectories that can be observed in the dataset, while dashed lines represent potential trajectories that might be missing from the dataset.

Barabasi 2008; Feng et al. 2017) by comprehending their preferences. In response to this, the next Point-of-Interest (POI) recommendation has emerged as a fundamental ingredient in smart city applications, encompassing the prediction of a particular user's next probable POI visit based on his/her historical trajectories (Xi et al. 2019; Kim et al. 2021; Zhuang et al. 2022; Luo et al. 2023a). It not only enhances travel experiences (Gao et al. 2022) by presenting more delightful options aligned with users' past preferences but also empowers businesses to achieve precise advertising delivery, thus facilitating their promotional efforts.

A user's trajectory is formed by sequences of check-ins arranged in chronological order, enabling us to draw a comparison with word sequences (i.e., sentences) in language models for modeling. Existing solutions for modeling sequential data commonly include approaches based on Markov chains, Recurrent Neural Networks (RNNs), and

attention-based models. Initially, Markov chains were proposed to capture sequential patterns (Cheng et al. 2013). As computing power grew and data quality improved, deep learning methods advanced rapidly. This led to the integration of memory-enhanced RNNs and attention-based models to effectively capture the long-term periodicity and short sequential features within user trajectories (Zhu et al. 2017; Feng et al. 2018; Sun et al. 2020; Luo et al. 2023b). Besides basic sequential patterns, time intervals or/and geographical distances between two consecutive visits are also found with robust predictive capability (Liu et al. 2016; Zhao et al. 2020; Yang et al. 2020; Luo, Liu, and Liu 2021; Wang et al. 2022). Furthermore, techniques like hierarchical grid-based partitioning and global transition graphs (Lian et al. 2020; Rao et al. 2022), along with category features (Pang and Zhang 2017; Yang, Liu, and Zhao 2022), are used to enrich the existing features, leading to enhanced overall performance.

However, individual mobility often remains unpredictable (Song et al. 2010) due to data missing and sparsity. As shown in Figure. 1, due to the voluntary nature of users' check-in sharing on the platform, User 1 has not shared his/her check-in record at the restaurant. This leads to the incompleteness of trajectory data, which can compromise the model's ability to capture patterns in the user's trajectory context (Yang et al. 2020). By analyzing the Gowalla dataset, we have discovered that the average time interval between two consecutive check-ins is approximately 51.28 hours. This observation highlights a significant degree of sparsity in the data. Yet, existing methods for trajectory prediction assume complete data and struggle with this sparsity. Furthermore, missing POI identification works (Xi et al. 2019) haven't a validation of the task's applicability in enhancing downstream POI recommendations. To alleviate the problem, we propose augmenting incomplete trajectories with potentially missing check-ins. Nevertheless, designing the augmentation process in the next POI recommendation task is not trivial, as it entails addressing the following three challenges: (1) how to identify potentially missing check-ins between any two mobile movements in order to complete the trajectory sequence? (2) how to retain the crucial spatial contextual information after completing the augmented trajectory sequence? (3) how to ensure that the incorporated check-ins within the trajectory sequence effectively enhance the predictive accuracy for the next POI?

Against this background, we proposed TAU, a general augmentation method designed for modeling incomplete user mobility trajectory sequences. More precisely, our solution explicitly employs breadth-first search to locate paths within a specified length that can reach the target nodes; as a result, expanding the initial input trajectory sequences with potentially missing check-ins prompts the generation of a more contextually enriched sequence (w.r.t, challenge (1)). This enhancement enables the sequence model to directly benefit from the abundant context. To effectively incorporate geographical relationship patterns within the completed trajectory, we explicitly utilize spatial context to identify highly predictive hidden states generated by the self-attention layer (w.r.t, challenge (2)). It's worth noting that our TAU has not

modified the self-attention module, allowing it to seamlessly integrate with self-attention layers. Furthermore, to assess the impact and certainty of the padded check-ins on predicting the next target POI, we introduce confidence-aware uncertainty estimation to enhance the reliability of pseudo-labels (w.r.t, challenge (3)). We summarize the primary contributions of our work as follows:

- We propose the TAU model based on the self-attention mechanism for accurate the next POI recommendation. Our approach adeptly addresses the challenge of incomplete trajectory sequences caused by inherent gaps in user check-in data within the dataset.
- We have developed a novel approach for retrieving intermediary nodes between any two target nodes on a graph to prompt the completion of potentially missing check-ins. Additionally, we enhance the impact of these completed check-ins on predicting the next location by incorporating novelty confidence-aware uncertainty estimation.
- We comprehensively assess the performance of our approach in comparison to an extensive array of baselines across two well-known real-world Location-Based Social Networking (LBSN) datasets. The results reveal that TAU consistently and significantly outperforms existing state-of-the-art methods.

Related Work

Evidential Deep Learning with Uncertainty Modeling

Evidence-based deep learning with uncertainty modeling is an approach that focuses on improving the robustness, reliability, and interpretability of deep neural networks by considering and quantifying uncertainties in both input data and model predictions. This framework recognizes the inherent uncertainty that exists in real-world data and seeks to offer a more comprehensive comprehension of the behavior exhibited by the model. Uncertainty in input data, stemming from noise, incompleteness, or ambiguities, can be managed through modeling, making deep learning models more robust. Bayesian neural networks (BNN) and dropout (Srivastava et al. 2014) are techniques employed to capture and account for uncertainties inherent in input data. Recently, evidential deep learning (EDL) has been developed by integrating evidential theory into deep neural networks (Sensoy, Kaplan, and Kandemir 2018; Bao, Yu, and Kong 2021), yielding promising results in classification tasks. In this study, to the best of our knowledge, we are pioneering the integration of evidential learning for POI recommendations.

Next POI Recommendation

The fundamental concept underlying the majority of existing POI recommendation models is the synergistic incorporation of spatial-temporal factors into the time-series module. STRNN (Liu et al. 2016) is an earlier work that incorporates spatial and temporal interval context-parameterized transition matrices or gates as spatial-temporal factors into the recurrent hidden states of RNNs. STGN (Zhao et al.

2020) incorporates the time gate and distance gate to regulate the updates of short-term and long-term interests, respectively. Flashback (Yang et al. 2020) explicitly employs spatial-temporal context to search for historical hidden states that exhibit similar contexts to the current one. However, due to the sparsity of check-ins and excessively long time intervals between consecutive check-ins, the improvement in model performance is relatively limited. LSTPM (Sun et al. 2020) proposes to model long-term preferences and short-term geographical preferences separately using the nonlocal network and geo-dilated RNN, respectively. GETNext (Yang, Liu, and Zhao 2022) incorporates global transition patterns and diverse data trends into a transformer model. Graph-Flashback (Rao et al. 2022) constructs a weighted POI transition graph based on the learned representations and incorporates it into the sequential model. However, all these studies overlook the potential benefits of considering missing POIs in the trajectory sequences for prediction.

Preliminaries

Notations and Definitions

The user trajectories recorded on location-based service platforms are represented as a sequence of visit records (i.e., check-ins) ordered chronologically. Here, the trajectory information records the user set, location (POI) set, and timestamp set which are represented as $U = \{u_1, u_2, \dots, u_{|U|}\}$, $L = \{l_1, l_2, \dots, l_{|L|}\}$, and $T = \{t_1, t_2, \dots, t_{|T|}\}$ respectively, where $|U|$, $|L|$ and $|T|$ represent the sizes of their respective sets and are positive integers.

Definition 1 (Check-in and Trajectory Sequence). *The recorded check-in information consists of a quadruple $s_i = (u, l_i, t_i, g_i)$. It signifies that user u visited the location l_i at time t_i , while the geographical coordinates of POI l_i are represented by $g_i = (\text{latitude} = \alpha_i, \text{longitude} = \beta_i)$. We can formulate the visiting records as $S_u = \{s_i = (u, l_i, t_i, g_i) | i = 1, 2, \dots, N\}$ in chronological order. Based on geographical coordinates, we can easily obtain the spatial intervals Δs_{ij} between any two locations in the trajectory.*

Definition 2 (Trajectory Flow Graph). *We represent the global directed transition pattern as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{w})$. Here, $\mathcal{V} = \{v_1, \dots, v_N\}$ is a set of $N = |\mathcal{V}|$ vertices representing location (i.e., POI), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges representing the connectivity among vertices. \mathbf{A} is the weighted adjacency matrix of global transition pattern \mathcal{G} . \mathbf{w} is the weight of the edge, it represents the transition possibilities between the two check-ins.*

Problem Formalization

Sequential POI recommendation is typically carried out through the following process: Given a user’s trajectory sequence, we leverage the temporal order of contextual relationships and spatial distance metrics to extract dependencies among POIs. By amalgamating sequential visit patterns and spatial POI distribution factors, this approach comprehensively captures user behavior and inclinations, the subsequent step involves predicting the suitable Top-k recommendation locations offered to the user.

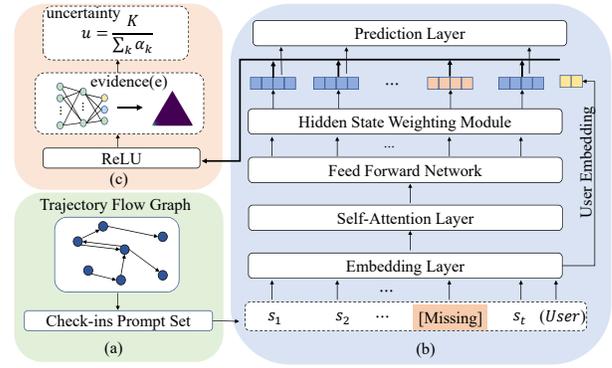


Figure 2: The overall architecture of TAU. (a) Trajectory Completion Module. (b) Base Model. (c) Confidence-Aware Evidential Uncertainty Calibration Module.

Base Model

Embedding Layer. In the embedding mapping stage, we first obtain an initialized POI embedding matrix $M \in R^{|L| \times d}$, and refer to the Graph-Flashback (Rao et al. 2022) method to enrich the POI representation $M' \in R^{|L| \times d}$. Given an input sequence of length n , we apply the look-up operation from M' to form the input presentation sequence $E \in R^{n \times d}$. Besides, we incorporate a learnable position encoding matrix $P \in R^{n \times d}$ and sum it with M' to obtain the updated E' (i.e., $E' = E + P$).

Self-Attention Aggregation Layer. Motivated by the effectiveness of the self-attention mechanism for modeling long sequences, we use the sequence encoder by stacking multiple Transformer (Vaswani et al. 2017) blocks. A Transformer block usually consists of two sublayers, which include a multi-head self-attention layer and a point-wise feed-forward network. Specifically, the multi-head self-attention is defined as:

$$S(F^l) = [\text{head}_1, \text{head}_2, \dots, \text{head}_h]W^O, \quad (1)$$

$$\text{head}_i = \text{Attention}(F^lW^Q, F^lW^K, F^lW^V),$$

where the F^l is the input of the l -th layer ($F^0 = E_I$), and the projection matrix W^Q, W^K, W^V, W^O are the learnable parameters. And, the attention function is as follows:

$$\text{Attention}(F^lW^Q, F^lW^K, F^lW^V) = \text{softmax}\left(\frac{F^lW^Q(F^lW^K)^T}{\sqrt{d/h}}\right)F^lW^V, \quad (2)$$

where $\sqrt{d/h}$ is the scale dot-product operation to help prevent large values that could lead to gradients vanishing or exploding during training. Subsequently, a point-wise feed-forward network layer is added, comprising two linear transformations with a ReLU activation in between.

Prediction Layer. To capture the user’s personalized preferences, we designed the user embedding representation e^u and concatenate it with the output \hat{y}_t of the aggregation layer at each time step t and then feed it into a fully connected layer to generate the final output as follows:

$$\hat{y}_t^u = W_f[h_t \oplus e^u], \quad (3)$$

where $W_f \in R^{|L| \times 2d}$ is a learnable weight matrix, \oplus denotes the concatenation operation. We use the cross-entropy function as our loss function.

$$Loss = - \sum_{S^u \in S} \sum_{i=1}^n (\log \sigma(\hat{y}_k^u) + \sum_{j=1, j \neq k}^{|L|} \log(1 - \sigma(\hat{y}_j^u))), \tag{4}$$

where n is the length of the trajectory sequence, σ is the softmax function. $\hat{y}_k^u \in \hat{y}_i^u$ is the predicted value of label l_k , $\hat{y}_j^u \in \hat{y}_i^u$ is the other POIs $l_j \neq l_k$ for current loaction l_i , respectively, and σ is the softmax function.

Methods

We present the framework of TAU in Fig. 2. It consists of a Trajectory Completion Module, a base model, and an Evidential Uncertainty Calibration Module. The Trajectory Completion Module uses a potential missing nodes search strategy to collect potentially missing check-ins between source and target locations and completes the potentially missing check-ins along the source input. Then, the base model explicitly utilizes spatial context to search the hidden representations generated by self-attention. Finally, we employ the Confidence-Aware Evidential Uncertainty Calibration Module for regularization during training.

Trajectory Completion Module

The Trajectory Completion Module converts the raw input S_u (i.e., incomplete trajectory sequence) into a sequence of trajectories S'_u that are completed using potential check-ins. This provides richer contextual information for the subsequent modeling of trajectory sequence contextual relationships. Due to space constraints, we have included the pseudocode in the supplementary material.

Potential Missing Nodes Search Strategy. We approach the issue of identifying potentially missing nodes between two consecutive check-ins (referred to as the source node and the target node) as a form of local search conducted on a trajectory flow graph. In Figure 3(a), we illustrate a graph where, given a source node denoted as U , our objective is to generate (sample) its set of transition nodes $N_S(U, T)$ leading to the target node T . In order to proficiently delve into potentially missed check-ins, we propose to use breadth-first sampling (BFS) for efficient exploration. Broadly, BFS is a graph traversal algorithm that commences from the root node, delving into neighboring nodes at the current depth before proceeding to nodes at subsequent depth levels. Using the current node as a starting point, it searches for intermediary nodes leading to the target node. During each subsequent graph traversal layer, we check whether any neighboring nodes contain the target location node. If found, the node transition path between the starting node and the target node is recorded in the memory set $walks$, linked by corresponding node pairs. At the same time, considering memory limitations, we set the search depth to a hyper-parameter m , which will be studied in the experiments. Additionally, we calculate the average of all edge weights for the trajectory segments obtained from each search, and these averages are used as the weights to select the respective segments.

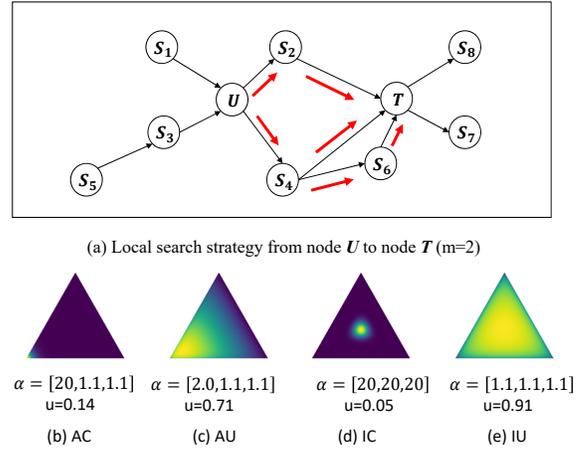


Figure 3: Local Search Strategy and Examples of Probability Simplex. In (a), we use the red solid line to indicate valid potential missing node search paths, and we obtain $[S_2]$ and $[S_4, S_6]$ as the sets of missing nodes for the prompts between U and T . In (b,c,d,e), we use 3-class classification as an example to show how Accuracy and Certainty (AC) are predicted. A well-calibrated model should give AC predictions or Inaccurate and Uncertain (IU) predictions, while AU and IC cases need to be reduced.

The process comprehensively identifies the potential missing check-ins for subsequent processing.

Once we have the set $walks$ of possible transition nodes between any pair of nodes, we can consider inserting them into the trajectory by randomly sampling them according to their weights. Thus, a sequence of trajectories with richer contextual information can be obtained. The pseudo-code of the search algorithm is shown in Appendix A.

Spatial-Aware Hidden State Weighting Module

In order to more effectively consider spatial factors, inspired by Flashback (Yang et al. 2020), We explicitly use the spatial context distance as a similarity function ω , which responds to the spatial relationship between the historical hidden state h_j and the current one h_i ($j < i$). As we found through our ablation experiments in the Flashback method that time intervals did not play a decisive role in the model’s performance, we chose to concentrate solely on modeling spatial context relationships. Accordingly, we use a distance exponential decay weight as the similarity function:

$$\omega(\Delta d_{i,j}) = e^{-\eta \Delta d_{i,j}}, \tag{5}$$

where $\Delta d_{i,j}$ is the spatial interval between the coordinates of POIs l_i and l_j , η represents the spatial decay weight, controlling how fast the weights decrease according to $\Delta d_{i,j}$. Subsequently, this module incorporates the similarity function $\omega(\cdot)$ and the historical hidden state h_j ($j < i$) into current one at each time step i as follows:

$$\hat{h}_i = \frac{\sum_{j=0}^i \omega_j * h_j}{\sum_{j=0}^i \omega_j}, \tag{6}$$

where ω_j denotes the similarity function $\omega(\Delta d_{i,j})$.

Confidence-Aware Evidential Uncertainty Calibration Module

While the spatial-aware aggregation layer effectively captures spatiotemporal periodic patterns, the supplemented missing nodes carry substantial uncertainty, making it challenging to ensure their consistently positive impact on context modeling. Therefore, we incorporate uncertainty calibration to quantify the value of the supplemented nodes for trajectory context modeling. To address this problem, we introduce a calibration of the relationship between the accuracy and uncertainty of the missing check-ins for predicting the target location. We encourage the model to learn a skewed and sharp Dirichlet simplex for accurate prediction (Figure. 3(b)) and to provide an unskewed and flat Dirichlet simplex for incorrect prediction (Figure. 3(e)). Therefore if a padded check-in is assigned a high level of uncertainty, it implies that it could be incorrect, thus identifying valueless padding.

Background of Evidential Deep Learning. Since the softmax layer outputs scores that are point estimates of the predictive distribution and tends to be overconfident about false predictions. We are motivated by the principles of uncertainty modeling in evidential deep learning (EDL) (i.e., derived from the evidence framework of Dempster-Shafer Theory (DST) and subjective logic (SL)) and assume that class probability follows a prior Dirichlet distribution. Here, we use the evidence function (i.e., ReLU function) g instead of the softmax layer to get non-negative evidence $e_k = g(F^l)$. The predictive uncertainty u is determined by $u = K/S$, where K is the number of class, and $S = \sum_{k=1}^K \alpha_k$ denotes the total strength of a Dirichlet distribution $Dir(p|\alpha_k = e_k + 1)$.

Evidential Uncertainty Calibration. In particular, we propose an Evidential Uncertainty Calibration (UC) method to constrain the relationship between confidence p_i and uncertainty u_i :

$$L_{UC} = -\lambda_t \sum_{i \in \{\hat{y}_i = y_i\}} p_i \log(1 - u_i) - (1 - \lambda_t) \sum_{i \in \{\hat{y}_i \neq y_i\}} (1 - p_i) \log(u_i), \quad (7)$$

where p_i is the maximum class probability of the next check-in in raw trajectory and u_i is the associated uncertainty, $\lambda_t = \min(t/T, 1)$ is the annealing factor. The motivation for setting the annealing factor is that the accuracy of the prediction is different at different periods of training. Specifically, We expect the first term to obtain a low uncertainty ($u_i \rightarrow 0$) when the model gives accurate predictions ($\hat{y}_i = y_i, p_i \rightarrow 1$), while the second term gives a high uncertainty ($u_i \rightarrow 1$) when the model gives inaccurate predictions ($\hat{y}_i \neq y_i, p_i \rightarrow 0$).

Confidence-Aware Evidential Uncertainty Calibration. To enhance the model's effectiveness in optimizing the uncertainty of potential missing check-ins, we specifically introduced a confidence-aware threshold. This signifies that we optimize only the hidden states of potential missing check-

Dataset	Gowalla	Foursquare
#users	52,979	46,065
#locations	121,851	69,005
#check-ins	3,300,986	9,450,342
#Average time between Successive check-ins	51.28 hours	58.59 hours
	2.13 days	2.44 days

Table 1: Basic dataset statistics.

ins with relatively lower uncertainty (i.e., higher confidence levels). Then, the confidence-aware uncertainty calibration on potential missing check-ins is defined as:

$$L_{CUC} = -\lambda_t \sum_{i \in \{\hat{y}_i = y_i\}} I(\max(u_i) < \gamma) p_i \log(1 - u_i) - (1 - \lambda_t) \sum_{i \in \{\hat{y}_i \neq y_i\}} I(\max(u_i) > (1 - \gamma))(1 - p_i) \log(u_i), \quad (8)$$

where $I(\max(p_i) \geq \gamma)$ is an indicator function that yields 1 if $\max(p_i) \geq \gamma$ is satisfied, and outputs 0 otherwise in the first term. $0 < \gamma < 1$ is a predefined threshold.

Compared with uncertainty calibration, the biggest advantage of L_{CUC} is that it only considers "highly confident" potential missing check-ins in the first term and "highly unconfident" potential missing check-ins in the second term. And they are determined by threshold γ in optimization. This mechanism helps reduce potential training noise by filtering out potential missing check-ins with less apparent uncertainty, thereby further enhancing the practical performance of the model. Combining $Loss$ and L_{CUC} with a hyper-parameter λ , the ultimate loss function for model optimization is formulated as follows:

$$L = Loss + \lambda \times L_{CUC}, \quad (9)$$

Experiments

Datasets

We evaluate our TAU on two real-world LBSN datasets: Gowalla¹, Foursquare². The user check-in data collected from the Gowalla dataset spans from February 2009 to October 2010, while the Foursquare dataset was collected from April 2012 to January 2014. Each check-in record comprises userID, POIID, latitude, longitude, and timestamp. The number of users, locations, check-ins, collection period, and average time interval are shown in Table 1. To ensure the quality of the datasets, we discarded inactive users with fewer than 100 check-ins and sorted the check-ins for each remaining user in order of increasing timestamps. During the partition of datasets, we use the first 80% check-ins of each user as a training set. In order to align the length of the input data sequence, each user's check-in sequence is divided into multiple sequences of equal length (e.g., typically

¹<https://snap.stanford.edu/data/loc-gowalla.html>

²<https://sites.google.com/site/yangdingqi/home>

Model	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
PRME	0.0740	0.2146	0.2899	0.1503	0.0982	0.3167	0.4064	0.2040
STRNN	0.0900	0.2120	0.2730	0.1508	0.2290	0.4310	0.5050	0.3248
DeepMove	0.0625	0.1304	0.1594	0.0982	0.2400	0.4319	0.4742	0.3270
STGN	0.0624	0.1586	0.2104	0.1125	0.2094	0.4734	0.5470	0.3283
SASRec	0.0787	0.1817	0.2511	0.1434	0.2399	0.4557	0.5479	0.3246
LSTPM	0.0721	0.1843	0.2327	0.1306	0.2484	0.4489	0.5018	0.3365
Flashback	0.1158	0.2754	0.3479	0.1925	0.2496	0.5399	0.6236	0.3805
STAN	0.0891	0.2096	0.2763	0.1523	0.2265	0.4515	0.5310	0.3420
GETNext	0.1343	0.3238	0.4039	0.2237	0.2667	0.5601	0.6382	0.3977
Graph-Flashback	0.1495	0.3399	0.4242	0.2401	0.2786	0.5733	0.6501	0.4109
TAU	0.1604	0.3672	0.4568	0.2571	0.2931	0.6174	0.6992	0.4410

Table 2: Performance on Baseline Datasets.

set at 20). And the remaining 20% is used as a testing set. In addition, potential check-in sequences are searched on a trajectory flow graph composed of the training set data.

Baselines

To evaluate the effectiveness of TAU, we compare it with the following state-of-the-art methods in the experiments: (1) **PRME** (Feng et al. 2015): An embedding method, which explores the sequential transition patterns by modeling user and POI embeddings. (2) **STRNN** (Liu et al. 2016): An invariant RNN model, which incorporates spatial-temporal factors between consecutive check-ins. (3) **Deepmove** (Feng et al. 2018): It uses RNN to model long-range dependencies and a historical attention module to capture the periodicity. (4) **STGN** (Zhao et al. 2020): Enhancing long-short term memory network by introducing spatial-temporal gates. (5) **SASRec** (Kang and McAuley 2018): An attention-based method that identifies the most relevant information from historical sequences and uses them to predict the next target. (6) **LSTPM** (Sun et al. 2020): A LSTM-based method, which uses a nonlocal network and a geo-dilated RNN for long-term preference and short-term preference, respectively. (7) **Flashback** (Yang et al. 2020): A RNN-based model, which uses spatial-temporal interval features to search for hidden states in historical information that have similar contexts to the current one. (8) **STAN** (Luo, Liu, and Liu 2021): An attention-based model, which explicitly incorporates spatial-temporal interval information to explore the effects between non-adjacent check-ins in historical sequences. (9) **GETNext** (Yang, Liu, and Zhao 2022): A transformer-based model, which incorporates the global transition patterns, spatial-temporal context, and category embeddings together into the model. (10) **Graph-Flashback** (Rao et al. 2022): A RNN-based model, which incorporates the weighted POI transition graph to capture the sequential transition patterns.

Model Settings. We implemented TAU on the PyTorch framework 1.10.1 and conducted all experiments on a Linux server with 128GB RAM, 16-core Intel i9-12900K@5.2GHz CPU, and Nvidia RTX 3090 GPU. The

key hyperparameters used in our model are listed below. We set the dimensions of POI embedding and user embedding to 10, and the self-attention aggregation module dimension is also 10. For the trajectory completion module, the depth m of BFS is 3. For the self-attention aggregation module, we stacked two encoder layers and the number of attention heads is 3 in the multi-headed attention module. The hyperparameter λ is 100. Moreover, we employed the Adam optimizer with a $1e - 2$ learning rate, and the temporal and spatial decay factors are set up the same as (Yang et al. 2020).

Evaluation Metrics. To verify the performance of our method, we use two widely used metrics in the experiments: (1) Accuracy@K (Acc@K), (2) Mean Reciprocal Rank (MRR). Acc@K represents the proportion of correctly predicted positive samples within the top- K predicted positive samples. And, we present the performance of the model in terms of recommendations at $k = 1, 5, 10$. MRR measures the average rank of the first correct prediction in a list of predictions.

Performance Comparison

Table. 2 shows the recommendation performance of TAU and baselines on the two datasets. The results in bold indicate the best performance, while the underlined results represent the second best. Based on the results of the baseline models and our proposed model, the following findings can be observed: (1) In predicting the next location, both temporal periodicity and spatial dependencies play an essential role in improving the model’s performance. (2) Modeling the hidden states by incorporating the temporal-spatial gaps between non-adjacent check-ins, represented as Graph-Flashback, STAN, Flashback, tends to outperform the approach represented by STGN and STRNN, which relies on modeling continuous time intervals between consecutive check-ins. And, the improved performance of LSTPM when utilizing geo-dilated POI sequences can also be interpreted as a strategy to account for non-continuous spatial information in check-ins. (3) TAU attainment of optimal performance can be predominantly attributed to its incorporation of check-in completion, which results in a more comprehen-

Model	Gowalla			
	Acc@1	Acc@5	Acc@10	MRR
w/o UCM	0.1487	0.3564	0.4477	0.2474
w/o confidence	0.1576	0.3656	0.4531	0.2564
w/o TCM	0.1536	0.3527	0.4442	0.2447
w/o SAM	0.1541	0.3559	0.4471	0.2492
w/o initialize	0.1415	0.3253	0.4027	0.2258
TAU	0.1604	0.3672	0.4568	0.2571

Table 3: Effectiveness of different modules in TAU.

Base Model	Gowalla			
	Acc@1	Acc@5	Acc@10	MRR
SASRec	0.0787	0.1817	0.2511	0.1434
TAU _{SASRec}	0.1041	0.2660	0.3463	0.1827
Flashback	0.1158	0.2754	0.3479	0.1925
TAU _{Flashback}	0.1415	0.3253	0.4027	0.2258
STAN	0.0891	0.2096	0.2763	0.1523
TAU _{STAN}	0.1241	0.2933	0.3677	0.2043

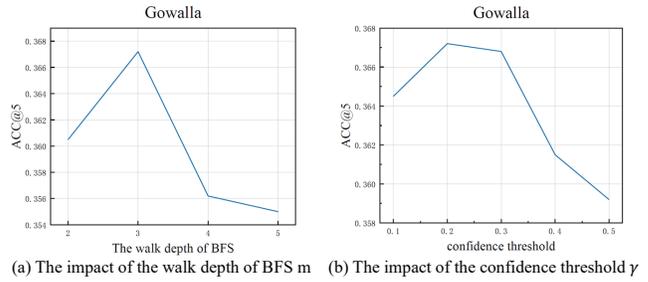
Table 4: Performance on Other Base Model.

sive context modeling. Further experiments are performed in Appendix B to validate the ability of the proposed TAU to accurately complete trajectory sequences.

Ablation Study

To analyze the effects of these components in our model, we conduct an ablation study, and the results on the Gowalla dataset are shown in Table 3. We denote the based model as TAU and drop different components to form variants as follows: (1) w/o Uncertainty Calibration Module (w/o UCM): we removed the Uncertainty Calibration Module and retained only the effect of trajectory completion on the model. (2) w/o Confidence-Aware Uncertainty Calibration Module (w/o confidence): we remove the effect of confidence thresholds in the modeling uncertainty calibration process. (3) w/o Trajectory Completion Module (w/o TCM): we remove the Trajectory Completion Module and use only the raw trajectory sequence as input to the model. (4) w/o self-attention mechanism (w/o SAM): we substitute the spatial-aware self-attention module with the spatial-temporal LSTM in GraphFlashback. (5) w/o initialize: we remove the initialized representation of POI embedding by trajectory flow graph and use only random initialization with spatial-temporal LSTM.

From Table 3, we have made the following findings: (1) The sequence of complemented potential check-in trajectories contributes to the modeling effect, which is mainly because the complemented trajectories contain rich contexts to help model the sequence contextual relationship. Despite the existence of noise sampling in the completed trajectory based on random sampling, it can still contribute to performance improvement. (2) Setting the confidence threshold contributes to model optimization. (3) The self-attention mechanism is superior to LSTM in our task scenario. (4) Modeling with only completed trajectory sequences per-

Figure 4: The performance comparison about the number of the walk depth m of BFS and the confidence threshold γ .

forms slightly worse than the method of initializing POI embeddings using transition graphs, but it still outperforms conventional methods (e.g., Flashback) by a significant margin.

Parameter Sensitivity Analysis

We implemented parameter sensitivity experiments on BFS walk depth m and confidence threshold γ to find the optimal parameter values on the Gowalla dataset. We vary the depth of BFS walk m from 2 to 5. And, we experiment with a series of the number of confidence-aware uncertainty threshold γ from 0.1 to 0.5. Figure 4 shows the $m = 3$ is the best depth, and $\gamma = 0.2$ is the optimality threshold.

The Performance on Other Base Model

To validate the generalization and versatility of our model, we also employed various alternative models to replace the base model and analyzed their corresponding effects on the Gowalla dataset, and the results are shown in Table 4. It is evident that utilizing uncertainty-guided completion of potential missing check-ins can significantly enhance the performance of each baseline model. This highlights that our approach can better enable context-aware modeling methods to capture contextual relationships more effectively.

Conclusion

In this work, we proposed TAU, a general attention architecture designed for modeling incomplete user mobility trajectory sequence by augmenting potential missing check-ins. Specifically, we utilize a global transition pattern graph to prompt potential missing nodes between source and target locations. Then, we incorporate the prompt memory set into the sequence by employing a weight-based random sampling approach, generating newly completed trajectory sequences. Simultaneously, to ensure the credibility of padded potential missing check-ins, we subject them to confidence-aware uncertainty calibration. Furthermore, we utilize spatial context to search for high-predictive hidden states within the completed trajectory. We perform comprehensive ablation experiments, parameter sensitivity analysis, and the performance of other base models in the experimental section. Comparative experiments with baseline models unequivocally demonstrate the superiority of our model, surpassing state-of-the-art approaches.

Acknowledgments

This work was supported in part by the NSFC under Grant No. 62072069, in part by the Fundamental Research Funds for the Central Universities (Grant. DUT22YG114), in part by the CAAI-Huawei MindSpore Open Fund, in part by the NSFC under Grant No. 62306258.

References

- Bao, W.; Yu, Q.; and Kong, Y. 2021. Evidential deep learning for open set action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13349–13358.
- Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.
- Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, 1459–1468.
- Feng, S.; Cong, G.; An, B.; and Chee, Y. M. 2017. Poi2vec: Geographical latent representation for predicting future visitors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Feng, S.; Li, X.; Zeng, Y.; Cong, G.; and Chee, Y. M. 2015. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, 2069–2075. ACM.
- Gao, Q.; Wang, W.; Zhang, K.; Yang, X.; Miao, C.; and Li, T. 2022. Self-supervised representation learning for trip recommendation. *Knowledge-Based Systems*, 247: 108791.
- Gonzalez, M. C.; Hidalgo, C. A.; and Barabasi, A.-L. 2008. Understanding individual human mobility patterns. *nature*, 453(7196): 779–782.
- Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.
- Kim, M.; Song, H.; Kim, D.; Shin, K.; and Lee, J.-G. 2021. Premere: Meta-reweighting via self-ensembling for point-of-interest recommendation. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 35, 4164–4171.
- Lian, D.; Wu, Y.; Ge, Y.; Xie, X.; and Chen, E. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2009–2019.
- Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.
- Luo, Y.; Duan, H.; Liu, Y.; and Chung, F.-I. 2023a. Timesteps as Prompts for Geography-Aware Location Recommendation. *arXiv preprint arXiv:2304.04151*.
- Luo, Y.; Liu, Q.; and Liu, Z. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*, 2177–2185.
- Luo, Y.; Liu, Y.; Chung, F.-I.; Liu, Y.; and Chen, C. W. 2023b. End-to-End Personalized Next Location Recommendation via Contrastive User Preference Modeling. *arXiv preprint arXiv:2303.12507*.
- Pang, J.; and Zhang, Y. 2017. DeepCity: A feature learning framework for mining location check-ins. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 652–655.
- Rao, X.; Chen, L.; Liu, Y.; Shang, S.; Yao, B.; and Han, P. 2022. Graph-flashback network for next location recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1463–1471.
- Sensoy, M.; Kaplan, L.; and Kandemir, M. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31.
- Song, C.; Qu, Z.; Blumm, N.; and Barabási, A.-L. 2010. Limits of predictability in human mobility. *Science*, 327(5968): 1018–1021.
- Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.
- Sun, K.; Qian, T.; Chen, T.; Liang, Y.; Nguyen, Q. V. H.; and Yin, H. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 214–221.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.
- Wang, E.; Jiang, Y.; Xu, Y.; Wang, L.; and Yang, Y. 2022. Spatial-temporal interval aware sequential POI recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2086–2098. IEEE.
- Xi, D.; Zhuang, F.; Liu, Y.; Gu, J.; Xiong, H.; and He, Q. 2019. Modelling of bi-directional spatio-temporal dependence and users' dynamic preferences for missing poi check-in identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5458–5465.
- Yang, D.; Fankhauser, B.; Rosso, P.; and Cudre-Mauroux, P. 2020. Location prediction over sparse user mobility traces using rnns. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2184–2190.
- Yang, S.; Liu, J.; and Zhao, K. 2022. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*, 1144–1153.
- Zhao, P.; Luo, A.; Liu, Y.; Xu, J.; Li, Z.; Zhuang, F.; Sheng, V. S.; and Zhou, X. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE*

Transactions on Knowledge and Data Engineering, 34(5): 2512–2524.

Zhu, Y.; Li, H.; Liao, Y.; Wang, B.; Guan, Z.; Liu, H.; and Cai, D. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. In *IJCAI*, volume 17, 3602–3608.

Zhuang, Z.; Wang, L.; Hao, Z.; Qi, H.; Shen, Y.; and Yin, B. 2022. Traffic Knowledge Graph Based Trajectory Destination Prediction. In *CCF Conference on Big Data*, 194–215. Springer.