Inquiring the Next Location and Travel Time: A Deep Learning Based Temporal Point Process for Vehicle Trajectory Prediction

Jie Zeng, Chenxi Xiao, Jinjun Tang*, Cheng Hu

***Abstract*—Trajectory prediction for individual vehicles has emerged as a vital component in IoT-based traffic management applications, inducing various control strategies for alleviating traffic congestion. This study focuses on a novel topic in this field, i.e., making joint predictions for the next location and travel time. Based on principles of vehicle mobility, we learn vehicle trajectories as discrete events in the spatiotemporal dimension and propose a neural temporal point process, named TrajTPP. This model employs two attention mechanisms to learn spatial and temporal dependencies, respectively, and a novel recurrent structure is proposed to integrate spatiotemporal features. Meanwhile, a gated residual attentive network (GRAN) is also designed to combine these learned dynamic features with static travel information. Then, the intensity-free learning strategy is employed to make probabilistic forecasting for the next travel times, and we develop a prior transition probability to involve historical travel behaviors in location predictions. Beyond the conventional prediction task, we design a sampling strategy to simulate vehicle mobilities by TrajTPP. Experiments from license plate recognition data in Changsha, China, demonstrate that our model outperforms advanced baselines, and sampling results provide evidence of its ability to accurately simulate vehicle mobilities. Moreover, its impressive accuracy on the latest next-location prediction benchmark is also listed in the Appendix.**

***Index Terms*—Trajectory prediction, temporal point process, probabilistic forecasting, trajectory generation.**

I. Introduction

IN urban road transportation systems, individual vehicles serve as the basic atoms. Their travel characteristics formulate the spatiotemporal evolution patterns of network-scale traffic flows. Therefore, exploring hidden patterns of vehicle trajectories can help to further understand the spatiotemporal dependencies in urban road networks. The rapid advancement of intelligent transportation systems (ITS) has led to the widespread proliferation of location-aware sensors, such as license plate recognition (LPR) and Bluetooth devices. With these widely installed traffic detectors, vehicle trajectories can be accurately monitored, resulting in the development of various IoT-based applications [1]. Specifically, these IoT methods facilitate accurate data fusion from traffic sensors located in different areas, leading to the development of innovative management strategies that enhance traffic control performance. This is particularly beneficial for location-based services (LBS), such as Mobility-as-a-Service (MaaS), vehicle navigation [2], etc. Currently, accurate prediction for future trajectories is regarded as the foundation of many LBS applications. Variants of individual trajectory prediction cover destination prediction [3], next location prediction [4], POI recommendation [5], etc. Due to the advantages in congestion alleviation [6] and route guidance [7], the next location prediction attracts the continuous attention of researchers [8], [9].

Since the beginning of this century [10], scholars have made sustained efforts on the next location prediction. Generally, this task aims to forecast the future locations of an individual traveler (or vehicle) based on the previous mobility records [11]. In this way, traffic managers can estimate future routes of vehicles in advance, so numerous personalized assistance strategies, such as dynamic routing, speed advice, etc., can be utilized to release potential congestion in urban road networks [12], [13]. For instance, once the future trajectories of numerous vehicles are known, adaptive signal control can be implemented through information transmission and fusion techniques among multiple detectors to reduce delays at intersections. Meanwhile, since accurate predictions for future trajectories highly rely on compressive mining of historical mobility patterns, this task can further enhance our understanding of vehicle travel behaviors in urban road networks. These features from trajectory modeling and prediction are vital to establishing capable traffic simulation models [6] to support numerous downstream tasks.

Current research in this field can be classified into two categories: (i) next-location prediction; (ii) joint prediction for

This research is funded by the Key R&D Program of Hunan Province (No. 2023GK2014), National Natural Science Foundation of China (No. 52172310), Humanities and Social Sciences Foundation of the Ministry of Education (No. 21YJCZH147), Science Research Foundation of Hunan Provincial Department of Education (No. 22B0010). *(Corresponding Author: Jinjun Tang.)*

J. Zeng was with the Smart Transport Key Laboratory of Hunan Province, School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China. He is now with the Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China. (e-mail: jzengav@connect.ust.hk)

C. Xiao, J. Tang, and C. Hu are with the Smart Transport Key Laboratory of Hunan Province, School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China. (e-mail: 234207003@csu.edu.cn; jinjuntang@csu.edu.cn; chenghu@csu.edu.cn)

the following locations and travel times (or arrival times). Since travel time is susceptible to uncertain and random factors, achieving accurate performance in the latter task is considerably challenging. Affected by this barrier, studies in the early stage are always limited in making predictions for the next location [4], [10]. However, this strategy cannot offer temporal information for traffic managers and travelers, so it may not be entirely practical. In addition, because locations and travel times are characterized by spatiotemporal patterns, the joint prediction can provide an opportunity for mutual correction, leading to higher prediction accuracies. Therefore, researchers gradually paid their interest in joint prediction about individual trajectories [14], [15], but critical limitations and barriers still lie, such as:

- Conventional studies always regarded the trajectory prediction task as the time-series prediction, so several recurrent neural networks (RNN) were directly applied [11]. However, vehicle trajectories reflect the mobility process in discrete spatial locations with discrete time stamps, such as intersections and road segments. So, this prediction task differs from classical time-series forecasting (e.g., traffic flow predictions). Although sequence characteristics are present in vehicle trajectories, we should adopt a capable approach that accommodates their discrete features, rather than directly applying traditional time-series prediction methods.
- Many deep learning models have been developed to improve trajectory prediction accuracies, but current studies mainly belong to point prediction, especially for travel times. It means that it only outputs a single value for the next travel time. However, travel times in urban road networks always express extreme uncertainty [16], so the predicted single value cannot represent the actual travel time accurately and reliably. Hence, probabilistic forecasting may be a potential solution to address this issue.
- In trajectory prediction, applications of conventional models are always limited to predicting future trajectories. Although several studies conducted interpretable analysis from prediction models, these efforts are still limited in experimental results analysis. Overall, current studies cannot provide abundant information to explore future evolution patterns of network-scale traffic flows. This problem is crucial in real-world traffic management but still lacks effective solutions. Therefore, how to generate future vehicle mobilities from these well-trained prediction models still needs further exploration.

Motivated by these limitations, this study attempts to model individual trajectories as the temporal point process (TPP) and extract the spatiotemporal dependencies. To learn the mobility patterns of individual vehicles, we propose the TrajTPP model via the effective integration of TPP and deep learning models. In this method, we first utilize the spatial and temporal attention mechanisms to capture the sequence patterns, respectively. Afterwards, a spatiotemporal gated recurrent unit (STGRU) is proposed to integrate the temporal and spatial correlations dynamically. Meanwhile, a gated residual attentive network (GRAN) is also developed to integrate these dynamic features with static information (e.g., vehicle type, day of the week, etc.). Finally, an intensity-free learning strategy is used to make probabilistic forecasting for the travel time. To further involve historical traffic information in the location prediction task, we also define a prior transition probability from historical travel behaviors and integrate it with learned context vectors. Overall, the primary contributions of this study are listed below.

- This study learns vehicle trajectories as the temporal point process (TPP) and proposes a TrajTPP model to make probabilistic predictions for the next location and travel time. TPP is employed to describe vehicle mobility patterns among discrete spatial locations, and several novel modules are further proposed and integrated to enhance learning capabilities.
- We develop a multi-view spatiotemporal learning framework to extract sequence patterns of vehicle trajectories. The STGRU model is proposed to effectively explore the interactions between spatial and temporal correlations, and GARN is designed to incorporate dynamic and static features.
- In addition to accurately predicting the next location and travel time, we further propose a vehicle trajectory simulation strategy using TrajTPP. This enables the well-trained TrajTPP model to serve as an effective simulator for generating spatiotemporal travel patterns of network-scale vehicle mobility.
- Extensive experiments from the real-world dataset in Changsha, China, reveal the superior performance of TrajTPP over several advanced trajectory prediction methods and neural TPPs. Meanwhile, the sampling experiment demonstrates its reliable capability in trajectory generation. We also show its superiority on the latest open-source benchmark in next-location prediction in the Appendix.

The remainder of this paper goes as follows. Section II briefly overviews the previous studies in trajectory prediction. We detailly describe the proposed TrajTPP model and its components in Section III. Experiment results from different aspects are summarized in Section IV. Finally, we conclude this work and provide outlooks for future studies in Section V.

II. Related Works

Trajectory prediction is a critical component in numerous LBS applications. In recent decades, numerous studies have been conducted in this field. According to prediction modes, current works can be classified into next location and joint predictions. The significant difference between these two categories is that the latter focuses on the simultaneous prediction of the next location and corresponding travel time. Representative studies and evolution trends in these two fields are summarized in the following subsections.

A. Next Location Prediction

As its name says, location prediction aims to predict future locations of an individual traveler or vehicle. Hence, exploring the hidden patterns in previous locations is the key step to achieving accurate forecasting [17]. This field arose in the early stage of this century, with pioneering scholars initially working to solve this problem using statistical models. From then on, Markov-based models have become the most influential method in the next-location prediction. Ashbrook & Starner [10] generated critical locations from GPS records by *k*-means clustering algorithm, and then a Markov chain (MC) model was proposed to forecast future movements. More details and recent developments about MC-based mobility prediction models can be found in [18], [19]. In addition to these Markov-based models, other probabilistic graphical models were also adopted in location prediction. For instance, Pathirana et al. [20] proposed a Robust Extended Kalman Filter (REKF) to predict the next location in the Global System for Mobile Communications network. Additionally, Monreale et al. [21] proposed a variant of the decision tree to estimate the next location, named T-pattern Tree.

However, limited by their simple structure, these statistical prediction methods may encounter challenges in prediction performance and generalizability. In the era of big data, deep learning methods gradually won the favor of researchers in numerous engineering problems [22]–[24]. Since the previous locations can be characterized as sequence data, the recurrent neural network (RNN) and its variants, i.e., long short-term memory (LSTM) and gated recurrent units (GRU), are widely applied in this field. For example, Liu et al. [4] combined the distance information with the conventional RNN model and proposed a spatial-temporal recurrent neural network (ST-RNN) for next-location prediction. Besides RNNs, Chen et al. [6] designed a convolutional embedding model (CEM) in next-location prediction. Liang & Zhao [12] considered the movement directions and combined it with a Seq2Seq model for vehicle trajectory prediction, named NetTraj. More recently, Chen et al. [25] introduced the multi-context features, such as individual preferences and social relations, and a graph neural network (GNN)-based model was proposed to predict the next location. Hong et al. [26] introduced the Transformer model into trajectory prediction and proposed a multi-head self-attentional neural network (MHSA) for the next-location prediction.

B. Joint Prediction

The location prediction task mentioned above focused solely on predicting the following locations, disregarding the corresponding travel times. However, incorporating future travel times could benefit both travelers and traffic managers, leading to a higher-quality traffic system. Following this idea, several efforts have been made in this field. Gidófalvi & Dong [14] employed an inhomogeneous continuous-time Markov model to predict the departure time and the next location of individual travelers. Zhao et al. [27] treated trip information as a tuple (*t*, *o*, *d*) and proposed a Bayesian *n*-gram model to make joint predictions for the start time (*t*), origin (*o*), and destination (*d*). Mo et al. [28] designed an input-output hidden Markov model (IOHMM) to make simultaneous predictions for the time and destination of the next trip. Afterwards, deep learning models are also widely adopted in joint prediction. Krishna et al. [15] developed a hybrid LSTM and a cascaded LSTM to predict the next mobility activity and the corresponding duration time. Sun & Kim [11] combined the self-attention mechanism with LSTM to predict the next location and travel time of individual vehicles.

Although these deep learning models can enhance the accuracies in trajectory prediction, the uncertainty in travel time is still unsolved. Actually, travel time in urban road networks is affected by numerous factors, so it is always characterized by uncertainty. However, current studies in trajectory prediction mainly focus on point prediction for time prediction, which means they only output a single value for the next travel time. Therefore, this type of prediction cannot explore the uncertainty in travel times, making prediction results inaccurate and unreliable. Instead of point prediction, probabilistic forecasting for the next location and travel time might be more suitable for this task. The temporal point process (TPP) model has been demonstrated as a powerful tool for modeling event sequences and making probabilistic forecasting. Currently, several researchers have applied the TPP-based models in traffic congestion modeling and prediction [29], [30]. Meanwhile, a few studies also introduced TPP into mobility prediction. Du et al. [31] proposed an RNN-based TPP model, named recurrent masked temporal point process (RMTPP), and evaluated it on the taxi pickup dataset in New York City. Additionally, Yang et al. [32] further developed a recurrent spatio-temporal point process (RSTPP) to make probabilistic forecasting for the check-in time. Wu et al. [33] introduced the attention mechanism into neural TPP models and proposed an attentive marked temporal point process (AMTPP) for individual mobility prediction in urban metro systems.

However, these previous studies focus on using deep learning methods to separately learn temporal and spatial dependencies in previous trajectories, ignoring the interactions between these two features. Meanwhile, the correlations between dynamic information (i.e., previous locations and travel times) and static factors (e.g., time of day, vehicle type, etc.) still need further exploration. Furthermore, these studies are also limited in the prediction task, and the capability to analyze future evolution patterns is still overlooked.

III. Methodology

In this section, we first provide the problem formulation to introduce the background and basic settings of our model. Then, we detail the framework and components from Sections III.B to III.F. Afterwards, a sampling-based simulation strategy is further proposed in Section III.G.

A. Problem Formulation

Table I shows the data samples of the involved LPR dataset,

TABLE I
DATA SAMPLES OF THE LPR DATASET

Index	License plate	Collection time	Location index
1	21030badd	2022-10-01 18:44:52	7
2	21030badd	2022-10-01 18:47:21	17
3	21030badd	2022-10-01 18:48:42	30
4	951f8dcc44	2022-10-01 23:11:30	99
…	…	…	…

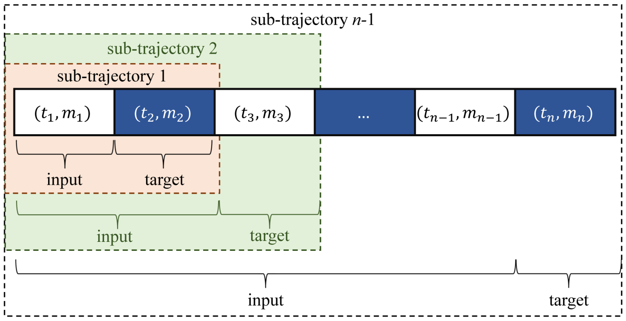


Fig. 1. Description of the sliding window strategy.

where each row represents a vehicle record at an intersection. Supposing t_j^k and m_j^k denote the arrival time and location of the jth record of vehicle k, the prediction task in this study can be concluded as: given a sequence of trajectory records of kth vehicle as $\mathcal{T}_n^k = \{(t_1^k, m_1^k), (t_2^k, m_2^k), \ldots, (t_n^k, m_n^k)\}$, the goal is to simultaneously predict the next location m_{n+1}^k and arrival time t_{n+1}^k.

Generally, prediction for the arrival time can be equivalently transformed to inter-time (i.e., travel time) forecasting, so we will employ these terms interchangeably throughout the entire paper. In this way, the prediction task can be mathematically formulated by Eq. (1) and Eq. (2). Here, $\tau_j^k = t_j^k - t_{j-1}^k$ denotes the travel time between the jth and the last location, and $\mathcal{F}(\cdot)$ represents the mapping function to estimate the probabilistic distributions of the next location and travel time. Therefore, $p(\tau_{n+1}^k, m_{n+1}^k | \mathcal{H}_n^k)$ can be regarded as the conditional probability distribution of the next travel time τ_{n+1}^k and location m_{n+1}^k based on $\mathcal{H}_n^k$, where $\mathcal{H}_n^k$ denotes the extracted features from $\mathcal{T}_n^k$.

$$\mathcal{T}_n^k = \{(t_j^k, m_j^k): j = 1,2,\ldots,n, \tau_j^k > 0\} \quad (1)$$

$$p(\tau_{n+1}^k, m_{n+1}^k | \mathcal{H}_n^k) = \mathcal{F}(\mathcal{T}_n^k) \quad (2)$$

It is worth noting that the length of trajectory (i.e., n) is variable in different trajectories, instead of using a predefined parameter as in several previous studies [12]. Additionally, we also introduce a sliding window strategy to make predictions. Taking a trajectory with n records as an example, the prediction mode can be abstractly summarized in Fig. 1, where the individual trajectory with length n can be regarded as splitting into $n-1$ sub-trajectories.

B. Framework

This study aims to use the TPP-based framework to address trajectory prediction. According to the definition by Shchur et al. [34], neural TPP always follows the following modeling steps:

(1) Encode the event sequence $\mathcal{T}_n^k$ into a feature vector y_n^k.

(2) Extract the context vector $\mathcal{H}_n^k$ by exploring the temporal dynamics in y_n^k.

(3) Parameterize the conditional distribution over the next event $p(\tau_{n+1}^k, m_{n+1}^k | \mathcal{H}_n^k)$.

Following this framework, we propose the TrajTPP model and illustrate its structure in Fig. 2. As shown in this figure, our model is composed of several critical components: (1) encode input location and travel time sequences into high-dimensional latent space and use attention mechanisms to learn the corresponding dependencies; (2) a spatiotemporal feature fusion block, which proposes the STGRU model to incorporate the extracted temporal and spatial features and employs a global attention mechanism to enhance long-term modeling capability; (3) an external feature fusion block based on the proposed GRAN model to integrate the dynamic features with static information; (4) an intensity-free learning

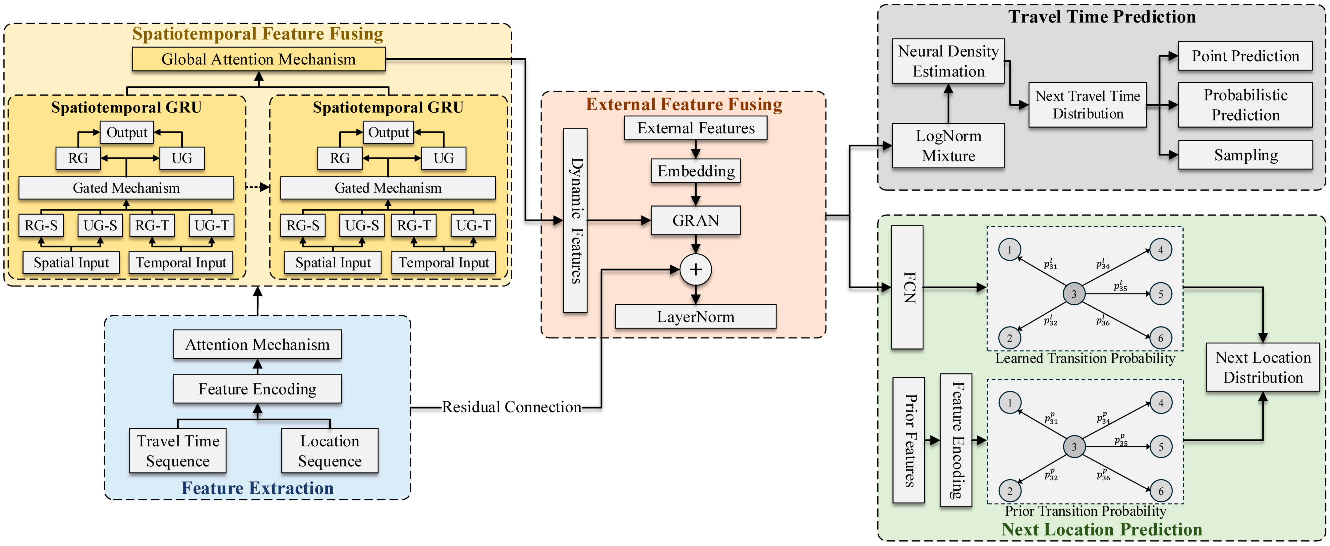


Fig. 2. Framework of TrajTPP model.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3523497

block relied on the log-norm mixture distribution to estimate distributions of the next travel time; (5) next-location prediction module, where we define a prior transition probability and integrate it with the learned context vectors to model the categorical distribution of the next location. In this figure, the "RG" and "UG" refer to the reset gate and update gate. Overall, the TrajTPP model can be regarded as an effective integration of conventional TPP and advanced spatiotemporal learning modules. All the components are described in detail in the following subsections.

C. Sequence Data Modeling

1) **Feature Encoding**

For each vehicle, its trajectory data $\mathcal{T}_n^k$ in this study records the continuous transformation patterns among different intersections. Therefore, it has typical characteristics of sequence data. More specifically, the trajectory data includes temporal patterns (i.e., travel time sequence) and spatial patterns (i.e., location sequence). Here, we represent the travel time sequence and location sequence as $s_t \in \mathbb{R}^n$ and $s_l \in \mathbb{R}^n$, where n denotes the trajectory length, and extract the hidden patterns from temporal and spatial dimensions, respectively.

It is noted that each element in the travel time sequence and location sequence is a scaler value. To fully explore the hidden information, it is necessary to encode them into higher-level features (i.e., $h_t \in \mathbb{R}^{n \times d_t}$ and $h_l \in \mathbb{R}^{n \times d_l}$) by a transformation layer. For the travel timess sequence s_t, all the elements are continuous, so we simply employ a fully-connected network (FCN) to enhance expressive capability. However, the linear transformation is unsuitable for s_l, because locations should be treated as discrete categorical variables. Currently, there are two types of methods to represent categorical features in LBS-based prediction, i.e., one-hot [11] and Embedding [3], [6]. Compared to one-hot encoding, the Embedding strategy can map encoding results into arbitrary dimensions, reducing dimensionality and enabling further exploration of correlations between different categories. This means that similar intersections can be placed closer together in the embedding space. Considering these advantages, we adopt Embedding to transform location sequence s_l into context vector h_l.

2) **Spatial and Temporal Attention Mechanism**

After the feature transformation, it is essential to explore the hidden patterns in each context vector. To achieve this goal, temporal and spatial attention mechanisms are applied to h_t and h_l, respectively. The self-attention mechanism [35] is the most widely used method in numerous sequence modeling tasks. Continuous efforts have been conducted to further improve the performance and efficiency of the conventional self-attention mechanism, e.g., Combiner [36], Interpretable multi-head attention mechanism [37], etc. Among these methods, the gated attention unit (GAU) [38] is a novel variant of the self-attention mechanism, and it shows significant superiority in effectiveness and learning capability. Especially, its well-designed structure can avoid the multi-head operations, making significant improvement in the computation cost. Therefore, we employ this method to implement the spatial and temporal attention mechanism. In this way, the sequence correlations in the travel time and location sequence can be extracted adaptively.

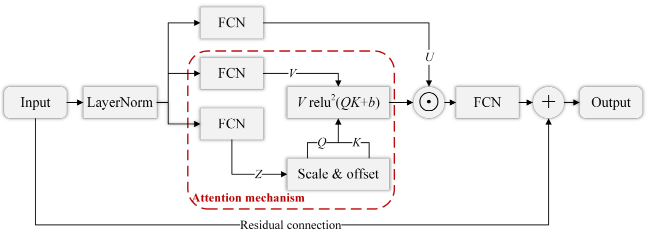


Fig. 3. Structure of the GAU model.

Fig. 3 shows the structure of the GAU model. Overall, it still inherits the "self" operation (i.e., Q, K, V) in the conventional self-attention mechanism. Here, we take the context vector of travel time sequence h_t as an example to describe the calculation process of this model. As shown in Eq. (3), a LayerNorm operation [39] (i.e., $\mathrm{LN}(\cdot)$) is firstly applied to the input data to enhance training stability. Afterwards, three parallel FCN layers are employed to transform the context vectors into different representation subspaces, which is shown in Eq. (4)-(6). In these equations, W and b represent the corresponding weight matrices and bias, and $\sigma(\cdot)$ denotes the activate function.

$$\tilde{h}_t = \mathrm{LN}(h_t) \tag{3}$$

$$U = \sigma(\tilde{h}_t W_u + b_u) \tag{4}$$

$$V = \sigma(\tilde{h}_t W_v + b_v) \tag{5}$$

$$Z = \sigma(\tilde{h}_t W_z + b_z) \tag{6}$$

Afterwards, as shown in Eq. (7)-(8), a "scale-offset" strategy is adopted to limit the learned attention weights to a specific range and eliminate the impact of numerical scales. This kind of dimensionless strategy is useful in enhancing training stability and performance. Here, a linear combination is employed to implement this strategy, where the γ and β are trainable parameters to control the boundary. Based on these two context vectors Q and K, the attention weight matrix A can be obtained through Eq. (9). In this equation, $\mathrm{relu}^2(\cdot)$ represents the square ReLU activate function (i.e., $[\mathrm{relu}(\cdot)]^2$) [40], and b denotes the bias. To avoid future information leakage, A is constrained to be a lower triangular matrix.

$$Q = \mathrm{scale\_offset}(Z) = \gamma_q Z + \beta_q \tag{7}$$

$$K = \mathrm{scale\_offset}(Z) = \gamma_k Z + \beta_k \tag{8}$$

$$A = \mathrm{relu}^2(QK + b) \tag{9}$$

Finally, as shown in Eq. (10), the attention weight matrix A is incorporated with V by multiplication, and this operation can be regarded as a "gate" to control information interactions. Here, $\odot$ denotes the element-wise production. Meanwhile, a residual connection (i.e., $h_t W_r$) is further applied to enhance the stability and convergence. Similarly, we also employ this attention mechanism on location context vector h_l to explore the spatial correlations.

$$O = (U \odot \hat{V})W_o + h_t W_r = (U \odot AV)W_o + h_t W_r \tag{10}$$

3) **Spatiotemporal Gated Recurrent Unit**

The above subsection adopts attention mechanisms to extract the corresponding correlations in the travel time and

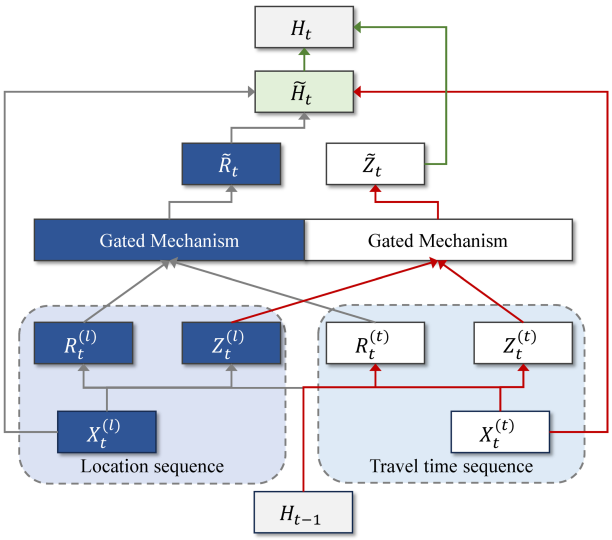


Fig. 4. Structure of the proposed STGRU model.

location sequences, but the hidden spatial and temporal correlations are separately addressed, which means there is a lack of interaction between these features. Meanwhile, conventional neural TPPs always use RNN-based structures to enhance the autoregressive modeling performance [31]. However, in these RNN-based models, the previous location sequences are mainly concatenated with the corresponding travel times as input data. According to vehicle mobility patterns, the arrival locations and times are highly related, so these structures are challenging to capture the hidden dependencies between these two sequences, leading to limited capability in sequence modeling. To address this issue, this study proposes a spatiotemporal gated recurrent unit (STGRU) model to effectively fuse these correlations.

The proposed STGRU is motivated by GRU, which has been widely used in numerous traffic prediction tasks. Supposing $X = \{x_1, x_2, \dots, x_n\}$ represents the input features, the mathematical formulation of the conventional GRU can be summarized as follows.

$$R_t = \text{sigmoid}(x_t W_{xr} + H_{t-1} W_{hr} + b_r) \tag{11}$$

$$Z_t = \text{sigmoid}(x_t W_{xz} + H_{t-1} W_{hz} + b_z) \tag{12}$$

$$\widehat{H}_t = \tanh(x_t W_{xh} + [R_t \odot H_{t-1}] W_{hh} + b_h) \tag{13}$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \widehat{H}_t \tag{14}$$

Here, the reset gate (i.e., R_t) and update gate (i.e., Z_t) are used to control the information interactions between previous and current input features. Similarly, W and b in these equations denotes the trainable weighted matrices and bias. However, the conventional GRU model only involves the temporal context, and there is no interaction between temporal and spatial information.

To address this problem, we propose the STGRU model and illustrate its structure in Fig. 4. Overall, it first separately inputs the location sequences ($X^{(l)} = \{x_1^{(l)}, x_2^{(l)}, \dots, x_n^{(l)}\}$) and the travel time sequences ($X^{(t)} = \{x_1^{(t)}, x_2^{(t)}, \dots, x_n^{(t)}\}$) into two different reset and update gates to generate the corresponding hidden features, i.e., $R_t^{(l)}$, $R_t^{(t)}$, $Z_t^{(l)}$, and $Z_t^{(t)}$. Afterwards, we further utilize a gated mechanism [41] to integrate temporal features with spatial features as follows.

$$w_r = \text{sigmoid}(R_t^{(l)} W_{r^{(l)}\tilde{r}} + R_t^{(t)} W_{r^{(t)}\tilde{r}} + b_{\tilde{r}}) \tag{15}$$

$$\tilde{R}_t = w_r \odot R_t^{(l)} + (1 - w_r) \odot R_t^{(t)} \tag{16}$$

$$w_z = \text{sigmoid}(Z_t^{(l)} W_{z^{(l)}\tilde{z}} + Z_t^{(t)} W_{z^{(t)}\tilde{z}} + b_{\tilde{z}}) \tag{17}$$

$$\tilde{Z}_t = w_z \odot Z_t^{(l)} + (1 - w_z) \odot Z_t^{(t)} \tag{18}$$

The proposed gated mechanism can be regarded as an adaptive combination of spatial and temporal information, while ensuring the sequence modeling capability. Specifically, it first calculates a weight (i.e., $w_r, w_z \in [0, 1]$) and uses it to integrate different context vectors. After this step of hidden feature fusion, we also apply this gated mechanism on the input data $x_t^{(l)}$ and $x_t^{(t)}$ to generate the comprehensive input representation $\tilde{x}_t$. Then, all the fused features are input to the following equations to update the current hidden states.

$$\widetilde{H}_t = \tanh(\tilde{x}_t W_{\tilde{x}\tilde{h}} + [\tilde{R}_t \odot H_{t-1}] W_{h\tilde{h}} + b_{\tilde{h}}) \tag{19}$$

$$H_t = \tilde{Z}_t \odot H_{t-1} + (1 - \tilde{Z}_t) \odot \widetilde{H}_t \tag{20}$$

By these operations, the proposed STGRU model can further explore the spatiotemporal dependencies among the input locations and travel times. However, since it still inherits the structure of GRU, a common problem in the conventional RNNs also arises, i.e., its long-term modeling capability is limited. Therefore, the extract context vector $\{H_1, H_2, \dots, H_n\}$ is further input to a global attention mechanism layer to enhance long-term learning performance. Here, we also select GAU to implement this attention mechanism.

D. External Feature Extraction

The deep learning modules mentioned above mainly aim to extract the spatiotemporal dependencies from dynamic features, i.e., previous locations and travel times. In addition to these dynamic features, other static factors also play a vital role in trajectory prediction [3]. Thus, we further introduce three critical external factors, i.e., start hour, day of the week, and vehicle type, to reflect collective travel patterns and enhance trajectory prediction performance. For each trip, these factors do not change over time, so we name them static features. According to their definitions, since these static factors are categorical variables, the Embedding strategy is also employed to encode them into high dimensions.

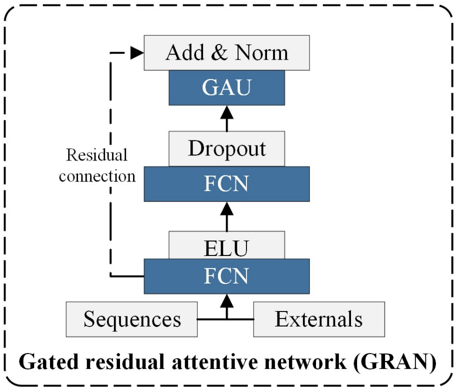


Fig. 5. Structure of the proposed GRAN model.

To improve the integration performance between dynamic and static features, based on the gated residual network (GRN) [37], we further propose a gated residual attentive network (GRAN). The framework of the proposed GRAN model is shown in Fig. 5. Supposing $H \in \mathbb{R}^{n\times d_h}$ denotes the dynamic features (i.e., context vectors from the global attention layer), where n and d_h represent trajectory length and hidden dimensions, respectively, the calculation process of the proposed GRAN model is shown below. In these equations, $C \in \mathbb{R}^{d_c}$ denotes the context vector of external features after Embedding, and $\mathrm{ELU}(\cdot)$ denotes the exponential linear unit (ELU) function [42].

$$\eta_1 = \mathrm{ELU}(HW_{1,1} + CW_{1,2} + b_1) \tag{21}$$

$$\eta_2 = \eta_1 W_2 + b_2 \tag{22}$$

$$\mathcal{H}_n = \mathrm{GRAN}(H, C) = \mathrm{LN}(H + \mathrm{GAU}(\eta_2)) \tag{23}$$

Here, $\mathcal{H}_n$ is the aggregated features that will be input to the probabilistic distribution estimation module. In Eq. (21), this method first utilizes a weighted combination of dynamic features (i.e., H) and static features (i.e., C), and the ELU function is employed to enhance nonlinear learning capability. Then, a linear layer without activation function in Eq. (22) is further utilized for feature extraction. Afterwards, a gated layer (i.e., GAU in Eq. (23)) is applied to integrated context vectors and further explores the sequence patterns in the fused features. Overall, relying on these equations, this model can effectively integrate the dynamic features with static factors, generating a more comprehensive representation of the previous travel patterns.

E. Probabilistic Distribution Estimation

1) Distribution of the Next Travel Time

Based on the extracted features from these deep learning modules, a TPP model is employed to estimate the probabilistic distribution of the next travel time. TPP is a capable tool to explore the temporal dynamics in discrete events, and it has been applied in numerous applications, including ambulance demand estimation [43], earthquake forecasting [44], etc. Conventional TPP methods always suffer limitations in flexibility and tractability to model the complex intensity function. Therefore, following the intensity-free learning strategy of TPP [45], TrajTPP aims to directly learn the probability density function (PDF) of the next travel time by a log-norm mixture distribution. Eq. (24) shows the formulation of log-norm mixture distribution. Here, $\boldsymbol{\omega} \in \mathbb{R}^M$, $\boldsymbol{\mu} \in \mathbb{R}^M$, and $\boldsymbol{\sigma} \in \mathbb{R}^M$ denote the weights, means, and standard deviations of this mixture distribution.

$$p(\tau|\boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{m=1}^{M} \frac{\omega_m}{\tau\sigma_m\sqrt{2\pi}} \exp(-\frac{(\log\tau - \mu_m)^2}{2\sigma_m^2}) \tag{24}$$

The intensity-free learning strategy learns these parameters from the aspect of neural density estimation. Specifically, it employs three parallel FCN layers with M hidden units, and then different activation functions are utilized to transform the learned context vectors to the unique constraint scope of each parameter. For instance, since weights in the mixture distribution should follow $\sum_{m=1}^{M} \omega_m = 1$ and $\omega_m \geq 0$, the softmax$(\cdot)$ activation function is employed in Eq. (25). In this way, probabilistic forecasting for the next travel time can be achieved by the estimated distribution.

$$\begin{cases} \boldsymbol{\omega} = \mathrm{softmax}(\mathcal{H}_n W_{\omega} + b_{\omega}) \\ \boldsymbol{\sigma} = \exp(\mathcal{H}_n W_{\sigma} + b_{\sigma}) \\ \boldsymbol{\mu} = \mathcal{H}_n W_{\mu} + b_{\mu} \end{cases} \tag{25}$$

2) Distribution of the Next Location

Generally, the principle of the next-location prediction task can be described as follows: given the current location and the transition probabilities to all other locations, the objective is to estimate the most probable next location. Therefore, we first apply a FCN layer on $\mathcal{H}_n$ to generate the learned transition probabilities of the input sequence as $H_n \in \mathbb{R}^{n\times N}$, where N is the number of intersections. Here, each row in this transition probability matrix, i.e., $h_n^m \in \mathbb{R}^N$, denotes the transition probability from the nth location of mth vehicle to all the candidate locations.

In addition to this learned transition probability, we can also extract several prior transition information from the historical trajectory dataset. For instance, when considering an intersection (i.e., A) with two downstream intersections, namely B and C, if the historical transition volumes from A to B are significantly higher than from A to C, we can infer that intersection B is more probable to be the subsequent location after intersection A. Following this opinion, we involve the prior volume transition probability to reflect the historic travel behaviors, and a fusion module is further developed to integrate this feature with the learned transition probability.

To achieve this goal, we first extract traffic volumes v_{ij}^p from the historical trajectory dataset, which denotes the number of vehicles from intersection i to j directly. Then, a normalization operation is utilized to generate the prior transition probability p_{ij}^p, i.e., $p_{ij}^p = v_{ij}^p / \sum_{k\in\mathbb{N}_i} v_{ik}^p$, where $\mathbb{N}_i$ denotes the potential downstream intersections of node i. Hence, p_{ij}^p can reflect the empirical likelihood between these intersections. To keep consistency with the learned transition probability h_n^k, we denote the transition probability vector from the ith location of vehicle k as $w_i^k \in \mathbb{R}^N$.

Moreover, when predicting the next location m_{n+1}^k based on the input sequence $\mathcal{T}^k = \{(t_1^k, m_1^k), (t_2^k, m_2^k), \ldots, (t_n^k, m_n^k)\}$, we can obtain n prior transition probability vectors (i.e., w_1^k, $w_2^k, \ldots, w_n^k$). To fully explore the hidden temporal patterns and extract a more expressive probability representation, we first concatenate these vectors to generate a prior transition probability matrix $W_n^k \in \mathbb{R}^{n\times N}$, and a GRU layer with hidden dimension N is applied to generate the hidden feature $\widetilde{W}_n^k \in \mathbb{R}^{n\times N}$. Here, the last row in $\widetilde{W}_n^k$, expressed as $\widetilde{w}_n^k \in \mathbb{R}^N$, can be viewed as the transition probability from the current location (i.e., m_n^k) to all the candidate intersections. Finally, to incorporate this prior information with the learned context vector h_n^k, a fusion strategy [46] is applied as follows.

$$z_h = \mathrm{sigmoid}(h_n^k \odot \widetilde{w}_n^k) \tag{26}$$

$$\tilde{h}_n^k = \mathrm{softmax}(z_h \odot h_n^k + (1 - z_h) \odot \widetilde{w}_n^k) \tag{27}$$

Algorithm 1. Sampling-based simulation strategy for future vehicle mobility generation from scratch.

```
Input:   Maximum time $T_{max}$, number of simulations $N$, number of trips $L$.
Output:  Sequence of simulated arrival times and locations $\mathcal{T} = \{\mathcal{T}_{1,1}, \dots, \mathcal{T}_{1,N}; \dots; \mathcal{T}_{L,1}, \dots, \mathcal{T}_{L,N}\}$.
for $l = 1{:}L$ do
  for $n = 1{:}N$ do
    Generating external features (i.e., $v_n^l, s_n^l, d_n^l$) and initiate location $m_{n,0}^l$ from categorical distributions.
    Set $\tilde{t}_0 = 0$, $i = 0$, and $h_0 \leftarrow \mathrm{TrajTPP}\{[t_{n,0}^l, m_{n,0}^l; v_n^l, s_n^l, d_n^l]\}$.
    while $\tilde{t}_i < T_{max}$ do
      Compute parameters $\boldsymbol{\omega}_{i+1}, \boldsymbol{\sigma}_{i+1}, \boldsymbol{\mu}_{i+1}$ for log-norm mixture distribution based on $h_i$.
      Sample the next travel time from the log-norm mixture through $\begin{cases} \boldsymbol{z} \sim \mathrm{Categorical}(\boldsymbol{\omega}_{i+1}) \\ \varepsilon \sim \mathrm{Normal}(0, 1) \\ \tau_{i+1} = \exp\,(\boldsymbol{\sigma}_{i+1}{}^T \boldsymbol{z} \cdot \varepsilon + \boldsymbol{\mu}_{i+1}{}^T \boldsymbol{z}) \end{cases}$.
      Sample the next location $\widetilde{m}_{i+1} \sim \mathrm{Categorical}\big(\mathrm{PTP}(h_i)\big)$.
      Compute the next arrival time $\tilde{t}_{i+1} = \tilde{t}_i + \tau_{i+1}$.
      if $\tilde{t}_{i+1} < T_{max}$ do
        Record the arrival time $t_{n,i+1}^l \leftarrow \tilde{t}_{i+1}$ and location $m_{n,i+1}^l \leftarrow \widetilde{m}_{i+1}$.
        Set $h_{i+1} \leftarrow \mathrm{TrajTPP}([t_{n,1}^l, \dots, t_{n,i+1}^l; m_{n,1}^l, \dots, m_{n,i+1}^l; v_n^l, s_n^l, d_n^l])$.
        Set $i \leftarrow i + 1$.
      end if
    end while
    $\mathcal{T}_{l,n} = \{(t_{n,1}^l, m_{n,1}^l), (t_{n,2}^l, m_{n,2}^l), \dots, (t_{n,i}^l, m_{n,\mathrm{i}}^l)\}$.
  end for
end for
```

In Eq. (26), by using $\mathrm{sigmoid}(\cdot)$ as the activate function, we aim to learn an element-wise weight matrix z_h, with all the elements in the range of [0, 1]. Then, Eq. (27) is utilized to make a linear combination between the prior transition probability $\widetilde{w}_n^k$ and the learnable context vector h_n^k. The final output feature $\tilde{h}_n^k$ is regarded as the categorical distribution for m_{n+1}^k, mathematically expressed as: $m_{n+1}^k \sim \mathrm{Categorical}(\tilde{h}_n^k)$.

F. Loss Function

The proposed TrajTPP method belongs to the generative models, so the negative log-likelihood (NLL) is utilized as the loss function for model training. This metric is shown in Eq. (28), where L denotes the total number of trajectories, and n_i represents the length of ith trajectory. Meanwhile, $p^*(\tau_j^i)$ and $p^*(m_j^i)$ denote the probabilities that τ_j^i and m_j^i follow the corresponding estimated distributions. Therefore, the first term and second term indicate the modeling performance of travel times and locations, respectively.

$$\mathrm{NLL} = -\frac{1}{\sum_{i=1}^{L} n_i} \sum_{i=1}^{L} \sum_{j=1}^{n_i} [\log p^*(\tau_j^i) + \log p^*(m_j^i)] \qquad (28)$$

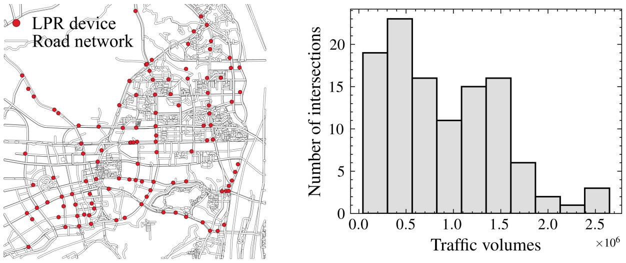
G. Sampling-based Simulation Strategy

Compared to conventional trajectory prediction methods, TPP stands out as a generative model, which extends its utility beyond basic prediction tasks. Its generative capability allows it to simulate the trajectories of numerous vehicles. In other words, we can use the well-trained TrajTPP model to simulate the spatiotemporal evolution patterns of vehicle mobility in urban road networks. Nowadays, data-driven trajectory simulation has become an important topic in human mobility analysis [47], playing a critical role in urban planning and traffic management. To address this issue cost-effectively using TrajTPP, we propose a sampling-based simulation strategy for trajectory generation based on the well-trained TrajTPP model.

Generally, there are two common approaches to generating new event sequences from TPPs, i.e., sampling from scratch and conditional sampling [48]. The difference between these two modes is that the former starts generation from $t = 0$ and knows nothing about previous events. In contrast, conditional sampling can be viewed as simulations for future trajectories based on previous k records.

Algorithm 1 summarizes the generation process of the first type. Since this study involves three external features in trajectory prediction, these features are unavailable for the scratch simulation. Thus, we first utilize three categorical distributions to fit the corresponding external features from the training and validation sets. For instance, if V denotes the categorical distribution of vehicle types, we can sample a vehicle type for l th simulation by $v_l \sim \mathrm{Categorical}\,(V)$. Similarly, the start hour s_l and day of week d_l for each sequence can be generated in a similar way. Meanwhile, since no prior location information is involved in this type of simulation, we sample an initiate intersection according to the categorical distribution of start locations from the historical trajectory dataset. Additionally, to better capture the real-world mobility patterns, we also generate a T_{max} for each iteration from empirical distributions of historical end times.

It is worth noting that this strategy is also applicable for conditional sampling, although it relies on prior knowledge about previous trajectories. In this case, the external features



(a) spatial distribution (b) traffic volume distribution

Fig. 6. Distributions of LPR devices and traffic features in the study area.

should be pre-known, so we can directly input them instead of making estimations from categorical distributions (i.e., Step 3). Additionally, h_0 in this situation should also be set as the context vector of the last record.

Overall, this algorithm can be considered as generating N potential trajectories for each trip, and this process (i.e., from Step 2 to Step 17) can be executed in parallel. In each simulation process, we iteratively generate the next locations and travel times using TrajTPP. Here, all the travel times and locations are generated by sampling from the log-norm mixture and categorical distributions, respectively. The context vector $\boldsymbol{z} \in \mathbb{R}^M$ in Step 7 denotes a one-hot vector [45], and the PTP(·) operation in Step 8 represents the prior transition probability (PTP) fusion module. Afterwards, as shown in Step 12, we integrate the current generated location $m_{n,i+1}^{l}$and arrival time $t_{n,i+1}^{l}$ with previous records to obtain a new context vector h_{i+1}. The above steps will be repeated until the termination condition T_{max} is satisfied.

IV. Experiments

A. Experiment Settings

1) Data Description

This study employs the license plate recognition (LPR) data from Changsha, China, as the case study. The LPR dataset is collected by fixed cameras at urban intersections, and it covers abundant traffic information such as passing time, license plate number, lane number, etc. As shown in Fig. 6a, there are 112 intersections in the study area, and the collection duration of this dataset was collected from October 1, 2022, to October 31, 2022. To ensure privacy protection, all license plate information has been anonymized by assigning a unique index to each vehicle. Meanwhile, the detection errors in LPR devices may introduce noise in trajectory sequences, leading to unreliable predictions. The specific solutions to this problem are summarized below.

- **Duplication recognition**. If a vehicle is detected by the same intersection multiple times within 30 seconds, we only keep one record and remove others.
- **Missed recognition**. The missed recognition of license plates may cause vehicle trajectories to be incomplete. Considering missed recognition has a low occurrence frequency, following the setting in the previous LPR-based trajectory prediction study [8], only location sequences with occurrence more than 30 times will be used for further analysis.

To obtain the consecutive sequences without parking, a temporal threshold δ_t is first applied to the original trajectories of each vehicle. That is, if the inter-time of two consecutive records from the same vehicle is longer than δ_t, we will split them into different trips. Since intersections in this study show a dense layout, we set δ_t as 15 minutes. Then, following the previous study [11], trips within $\delta_s = 5$ intersections will be regarded as abnormal trips and removed.

Existing studies have demonstrated that different types of vehicles may express different travel patterns [49], so we also involve this critical factor in trajectory prediction. According to the involved dataset, there are 4 major types of vehicles, i.e., small vehicles, heavy vehicles, small electric vehicles, and heavy electric vehicles. Therefore, an Embedding layer is employed to encode these vehicle types into high-dimensional features. Meanwhile, travel speeds and route choices may also exhibit different patterns throughout the day. After the trip splitting strategy by δ_t, the differences in the collection time for the same trajectory are not significant. Thus, for each trajectory, we further introduce the temporal information (i.e., start hour and day of the week) of the first record into external features. Finally, without loss of generality, we randomly select 100,000 trips for individual mobility modeling and prediction.

2) Evaluation Metrics

For the joint prediction of individual trajectories, this study focuses on where the next location is and how long the travel time will be. Considering the uncertainty in locations and travel times, the prediction performance of this task should be evaluated from three perspectives:

(i) Multi-class classification for the next location. Here, we employ the Mark NLL (NLL^l), accuracy (ACC), F1 score, and Recall@5 as metrics. The definition of Mark NLL can be obtained from Eq. (28) and serves as a similarity metric for the predicted and actual location distribution. Additionally, Recall@5 indicates the proportion of relevant locations successfully retrieved among the top 5 predicted locations.

(ii) Point prediction for the next travel time. The mean absolute error (MAE) is utilized to evaluate prediction performance, and we employ the median of the estimated PDF as $\hat{\tau}_i^j$ to denote the point prediction results.

(iii) Probabilistic forecasting for the corresponding travel time. Three metrics are introduced for evaluation, i.e., Time NLL (NLL^t), mean empirical coverage (MEC), and mean Pinball loss (MPL). Here, MEC is utilized to measure the average empirical coverage percentage. As shown in Eq. (29), Pinball loss is employed to evaluate the prediction performance of α-quantile. In this study, we employ {5%, 10%, …, 95%} quantile forecast to calculate MEC and MPL.

$$\mathrm{Pinball}_\alpha = \frac{1}{\sum_{i=1}^{N} n_i} \sum_{i=1}^{N} \sum_{j=1}^{n_i} \begin{matrix} \alpha \max\left(\tau_j^i - \hat{\tau}_{j,\alpha}^i, 0\right) + \\ (1-\alpha) \max\left(\hat{\tau}_{j,\alpha}^i - \tau_j^i, 0\right) \end{matrix} \tag{29}$$

For NLL^l, MAE, NLL^t, MPL, and NLL, a lower value indicates higher prediction performance. For the remaining metrics, higher values indicate better prediction performance.

3) Model Setting

We randomly divide the trajectory dataset into a training set, a validation set, and a test set, using a splitting ratio of 60%: 20%: 20%. The mini-batch training strategy with a batch size of 1024 is applied for model training. In the training process, the learning rate and regularization are set as 10^{-4} and 10^{-5}, respectively. Afterwards, the maximum training epoch is 500, and we adopt the early stop strategy on the validation set with patience of 20 to avoid overfitting.

The TrajTPP model is implemented using PyTorch on an Ubuntu workstation with RTX 3090 GPU. In this model, we set the hidden dimension of the FCN and Embedding layers in sequence data modeling to 128. For the log-normal mixture distribution, we assign the number of components as 64. As mentioned in the data description, there are four types of vehicles involved in this study, and an Embedding layer with the hidden dimension of 64 is used for feature transformation. Similarly, the static temporal information is also encoded by Embedding layers with the same dimension. Under these settings, the average training time per epoch is about 10 seconds, so the TrajTPP model can be well trained within 2 hours on a single GPU.

B. Baselines

This study introduces 11 advanced models as baselines. The involved baselines range from conventional methods to novel neural TPPs. We briefly summarize these baselines as follows.

- **FPMC** [50]. In the Factorizing Personalized Markov Chains (FPMC) model, the matrix factorization is incorporated with Markov chains to predict the next location.
- **GRU.** In this model, we first employ a GRU layer to encode the input travel time and location sequences. Then, two FCNs are applied to predict travel time and location, respectively.
- **DeepMove** [51]. DeepMove combines the attention mechanism with recurrent neural networks to make forecasting for the next location.
- **HST-LSTM** [52]. The Hierarchical Spatial-Temporal Long-Short Term Memory (HST-LSTM) model integrates the LSTM-based encoder-decoder structure with spatial-temporal influence to predict the next location.
- **H-LSTM** [11]. The Hybrid LSTM (H-LSTM) incorporates the LSTM layer with self-attention mechanism to predict the next location and travel time simultaneously.
- **S-LSTM** [11]. Sun & Kim [11] further proposed a Sequential LSTM (S-LSTM) model in the joint prediction task, which used two separate LSTM structures to learn the spatial and temporal correlations, respectively.
- **STAN** [53]. The Spatio-Temporal Attention Network (STAN) developed a bi-layer attention model to further explore the non-adjacent and non-consecutive relationships in human mobility prediction.
- **RMTPP** [31]. The RMTPP model is a pioneering effort which bridges the neural networks with conventional TPP methods. A GRU-based model is employed to capture the temporal dynamics in previous trajectories.
- **Exponential** [54]. This method employs an exponential distribution to model the intensity function of TPP. Similarly, it utilizes GRU to explore the time-varying patterns.
- **FullyNN** [55]. Motivated by the framework of conventional neural TPP, the FullyNN model also employs the GRU layer to enhance the temporal dependency. Afterwards, it proposes a learnable cumulative conditional intensity function and designs a neural network for parameter estimation.
- **LogNormMix** [45]. Shchur et al. [45] proposed an intensity-free learning framework for neural TPP modeling. It uses the log-norm mixture distribution to estimate the inter-times of the next events.

In the experiments, we employ the open-source urban computing package *libcity* [56] to implement FPMC, DeepMove, HST-LSTM, and STAN. In all the TPP-based baselines, locations and travel times are regarded as marks and inter-times, respectively. Based on this primary setting, we employ the deep learning method to estimate the parameters of each distribution. Additionally, to further show the superiority of spatiotemporal learning modules in TrajTPP, we combine these modules with density estimation methods in all the other TPP models, named TrajRMTPP, TrajExp, and TrajFullyNN in Table II.

C. Prediction Comparisons

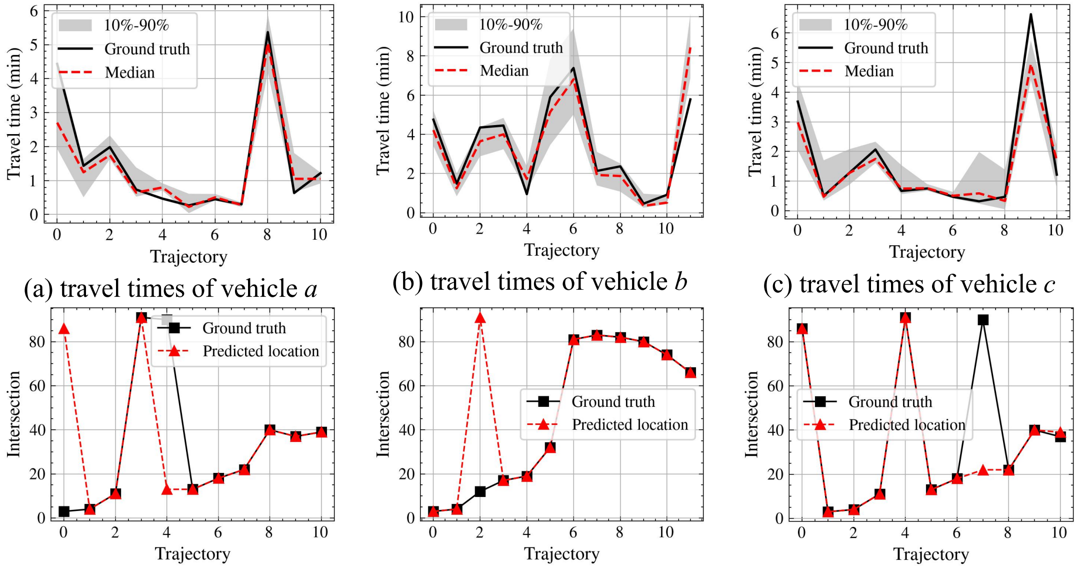
Table II summarizes the prediction performance of TrajTPP and the selected baselines. Metrics with the best performance are marked in bold. According to the prediction principle, several baselines only belong to location forecasting, so evaluation metrics for travel time prediction are unavailable. Meanwhile, we also take 3 vehicles as instances and compare their predictions with ground truths by TrajTPP in Fig. 7.

Overall, several interesting conclusions can be indicated from these results.

- In this table, FPMC reaches the poorest performance in the location prediction task. This poor accuracy highlights the challenge of capturing spatiotemporal dependencies in vehicle trajectories using a simple structure. Meanwhile, the future location prediction performance of GRU is also unacceptable, indicating that it is unsuitable to model the trajectories as time series directly. On the other hand, almost all the TPP-based baselines outperform these basic methods, and they can achieve comparable performance with advanced deep learning models. It is noticeable that, apart from a GRU layer, these TPP-based methods lack complex spatiotemporal learning modules. In other words, the intensity functions in these TPP-based baselines can help them to achieve accurate predictions, which demonstrates that TPP-based

TABLE II
PERFORMANCE COMPARISONS OF THE TRAJECTORY PREDICTION TASK

	Location prediction				Travel time prediction				NLL
	NLL^l	ACC	F1-score	Recall@5	NLL^t	MAE	MPL	MEC	
FPMC		0.475	0.417	0.827	\	\	\	\	\
GRU	\	0.617	0.603	0.896	\	1.187	\	\	\
DeepMove	\	0.720	0.714	0.966	\	\	\	\	\
HST-LSTM	\	0.724	0.720	0.967	\	\	\	\	\
H-LSTM	\	0.640	0.628	0.916	\	1.186	\	\	\
S-LSTM	\	0.626	0.616	0.906	\	1.068	\	\	\
STAN	\	0.531	0.539	0.920	\	\	\	\	\
RMTPP	0.804	0.723	0.718	0.967	1.577	1.553	0.793	0.358	2.381
Exponential	0.802	0.724	0.718	0.967	1.608	1.408	0.778	0.370	2.409
FullyNN	0.816	0.722	0.716	0.966	1.053	2.024	0.871	0.203	1.869
LogNormMix	0.833	0.720	0.712	0.964	1.004	0.992	0.385	**0.442**	1.837
TrajRMTPP (ours)	0.752	0.745	0.740	0.971	1.580	1.466	0.807	0.372	2.332
TrajExp (ours)	0.751	0.745	0.740	**0.971**	1.589	1.332	0.650	0.375	2.340
TrajFullyNN (ours)	0.755	0.743	0.738	0.970	0.968	2.295	0.953	0.179	1.723
TrajTPP (ours)	**0.751**	**0.746**	**0.741**	0.970	**0.868**	**0.913**	**0.356**	0.431	**1.619**



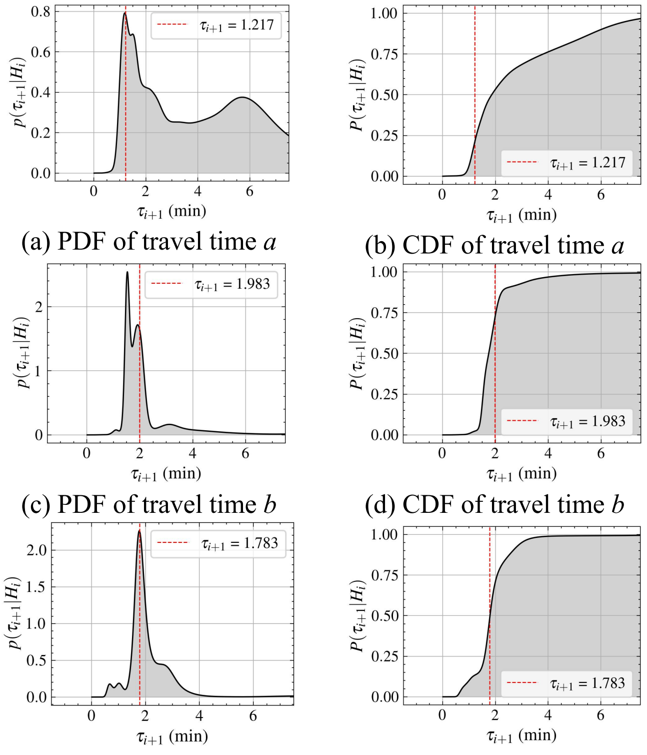
(a) travel times of vehicle *a* (b) travel times of vehicle *b* (c) travel times of vehicle *c*

(d) locations of vehicle *a* (e) locations of vehicle *b* (f) locations of vehicle *c*

Fig.7. Prediction results in LPR dataset with TrajTPP.

models are capable for the trajectory prediction task.

- It is worth noting that after fusing with spatiotemporal learning modules proposed in this study, the prediction accuracies of all the TPP-based methods (e.g., TrajRMTPP) are significantly improved. Specifically, the improvements of ACC and F1-score in these methods approximately exceeded 3%, outperforming all the other baselines. Meanwhile, the total NLLs mainly decrease by 0.1, with the largest decrease exceeding 0.2. This phenomenon demonstrates the effectiveness of our proposed model in sequence dependency extraction.
- Among all the baselines, TrajTPP expresses superior performance, especially in travel time prediction. It is the only model with NLL^t less than 0.90, and this metric on TrajTPP is much lower than all the other methods. Previous studies have identified the uncertainty in travel times [16], increasing the difficulties in achieving accurate prediction. As illustrated in Fig. 7, TrajTPP can effectively reduce this impact by probabilistic forecasting. Although time-varying patterns of travel times are highly dynamic, the prediction intervals can accurately capture these relationships. Furthermore, we also evaluate the performance of TrajTPP on the latest next-location prediction benchmark [26]. The prediction results are summarized in the Appendix, and the proposed TrajTPP model still expresses superiority over the latest baselines in all the metrics. We share the details about the experiment settings and results at https://github.com/SunderlandAJ-1130/TrajTPP.



(a) PDF of travel time *a*

(b) CDF of travel time *a*

(c) PDF of travel time *b*

(d) CDF of travel time *b*

(e) PDF of travel time *c*

(f) CDF of travel time *c*

Fig. 8. The estimated PDF and CDF distributions with TrajTPP.

Afterwards, it is noted that travel time prediction is a component of trajectory prediction, and its performance is also highly dependent on the accuracy of the other prediction task, i.e., location prediction. For instance, supposing we make an extremely accurate prediction for the next travel time via a point prediction, this accurate prediction result also seems to be useless if the predicted location is wrong. Instead, as illustrated in Fig. 8, if we implement probabilistic forecasting, the estimated PDF distribution always consists of several peaks. In these figures, CDF refers to the cumulative distribution function, and the red line denotes the ground truth travel time. Therefore, the predicted results can cover more potential information than a single value, making the predicted results more reliable. Overall, considering the uncertainty [16] in travel times, probabilistic forecasting may be a better solution for the trajectory prediction task.

D. Ablation Analysis

According to the definition in Section III, the proposed TrajTPP model consists of several crucial components. In the above subsections, we have demonstrated its superiority over advanced baselines, but the importance of each element is still unclear. Thus, we perform an extensive ablation experiment for further analysis. Here, we remove each critical component from TrajTPP to evaluate the influence of each module. Below is a detailed description of these components.

- *w/o GAU*. This model removes all the attention mechanisms in TrajTPP.
- *w/o GRAN*. This model drops the external feature learning module in TrajTPP.
- *w/o STGRU*. In this model, we replace the STGRU layer with the conventional GRU model.
- *w/o PTP*. This mode removes the prior transition probability (PTP) fusion module and only involves the learned context vector for next-location prediction.

TABLE III

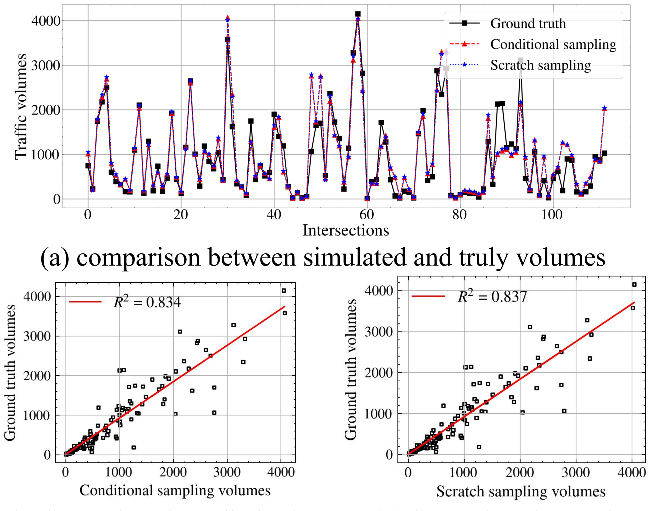
PERFORMANCE COMPARISONS OF ABLATION STUDY

	NLL^l	NLL^t	NLL
TrajTPP	**0.751**	**0.868**	**1.619**
w/o GAU	0.762	0.887	1.649
w/o GRAN	0.809	0.990	1.799
w/o STGRU	0.759	0.884	1.643
w/o PTP	0.756	0.878	1.634
-SA	0.772	0.902	1.674
-MHA	0.759	0.879	1.638
-IMHA	0.767	0.906	1.673

In addition, to further evaluate the performance of different attention mechanisms in the trajectory prediction task, we also employ the widely-used self-attention mechanism, multi-head attention mechanism, and the Interpretable multi-head attention mechanism [37] to replace the GAU model. To distinguish these methods, we name them as *-SA*, *-MHA*, *-IMHA*, respectively. In the *-MHA* and *-IMHA* models, the number of heads is set as 8.

Table III presents the experiment results of the ablation study from various aspects. After dropping each component, the prediction performance of all the sub-models experience decrease. It means that all the components are useful to improve the prediction performance of TrajTPP. Notably, the absence of the GRAN module exhibited the most significant reduction in both location and travel time prediction, emphasizing the critical role of external feature fusion modules in TrajTPP. Meanwhile, this phenomenon also indicates that the involved static factors can effectively reflect travel behaviors in the urban road network and provide more prior knowledge to the TrajTPP model, thereby improving the prediction performance. Afterwards, from the comparison between the remaining models, all the sub-models have experienced a decrease in their prediction accuracy, with *w/o STGRU* showing a larger decline. This phenomenon shows that the proposed STGRU model is capable of capturing the correlations between previous trajectory and location sequences. Meanwhile, this result also indicates that involving historical travel patterns is also effective in enhancing trajectory prediction performance.

Furthermore, in the lower half of Table III, we utilize three widely-used attention mechanisms to replace the GAU in TrajTPP. However, the prediction performance shows that all of these alternative attention mechanisms will lead to a decrease compared to GAU, especially for the naive self-attention and Interpretable multi-head attention mechanism. Moreover, it is noted that prediction accuracies of these two models are even lower than the TrajTPP without any attention



(a) comparison between simulated and truly volumes

(b) fit results of conditaionl sampling

(c) fit results of scratch sampling

Fig. 9. Comparisons of intersection traffic volumes between ground truth and sampling trajectories.

mechanism (i.e., *w/o GAU* in this table). Therefore, although there are numerous novel types of attention mechanisms, variances exist in the learning capabilities across different methods, so it is necessary to select a suitable method.

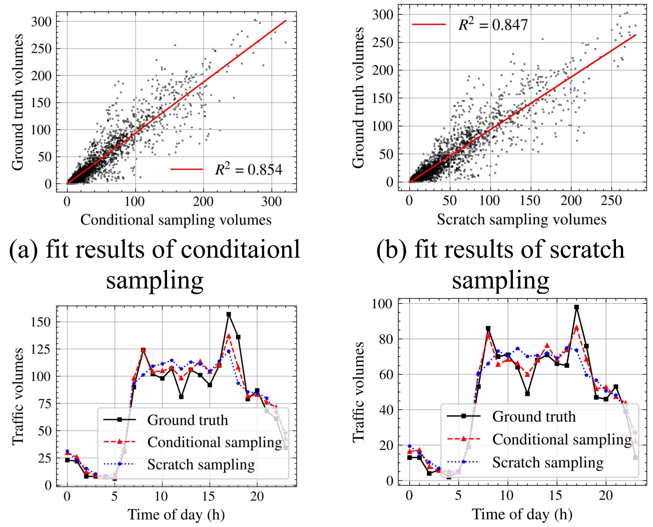
Overall, among all the combined models, TrajTPP consistently achieves superiority, suggesting that integrating these factors into trajectory prediction is useful.

E. Sampling Analysis

A primary advantage of TrajTPP over conventional trajectory prediction methods is that it can simulate evolution patterns of future vehicle trajectories via sampling. To evaluate the accuracy and reliability, we further conduct a sampling experiment on the test set. Here, we adopt both the conditional sampling and scratch sampling strategy to generate new trajectory sequences. Since the collection times on the test set are located in the whole month, the ground truth and simulation results are sparsely distributed in the temporal dimension. For simplification, we accumulate all the results into a single day for better visualization.

For the conditional sampling, we directly use the external features (e.g., vehicle type), initial location, and first collection time of each trajectory as prior information for sequence generation. On the contrary, for scratch sampling where prior information cannot be involved, we begin by estimating the categorical distributions of each external feature from historical trajectories on the training and validation sets. Then, during the simulation process, these categorical distributions are used to randomly sample the prior information as external features. Therefore, conditional sampling is applicable for implementing traffic control measures based on real-time vehicle information, and the latter is more useful in long-term traffic planning and policy evaluation. For instance, we can modify the percentage of small and heavy vehicles to explore travel behaviors under different penetration rates of trucks.

Following these settings, we simulate trajectories 100 times and employ average traffic volumes as the evaluation metric. Here, the estimated traffic volumes are calculated by the generated arrival times and locations. Fig. 9a illustrates the traffic flow comparison by TrajTPP. Each data point in this figure denotes traffic volumes at a specific intersection. This result shows that, for the majority of intersections, the simulated volumes can closely align with ground truths, indicating that these simulated trajectories can accurately capture the actual evolution patterns of network-scale traffic flows. Moreover, according to Fig. 9b and Fig. 9c, the R^2 for these two predicted lines are 0.834 and 0.837, respectively, and traffic volumes from scratch sampling are relatively closer to the true traffic volumes. This is because scratch sampling utilizes the abundant prior information to generate static travel information, making it a more comprehensive understanding of the actual patterns of vehicle mobility behaviors.



(a) fit results of conditaionl sampling

(b) fit results of scratch sampling

(c) intersection *a*

(d) intersection *b*

Fig. 10. Comparisons of hourly traffic volumes between ground truth and sampling trajectories.

Furthermore, we also conduct comparisons on time-varying characteristics of simulated traffic volumes at urban intersections. Here, we employ hourly traffic volumes as examples for illustration and display the fitting results among all the intersections in Fig. 10a and Fig. 10b, where R^2 indicates that the conditional sampling can better capture the temporal evolution patterns of vehicle mobility. Meanwhile, as shown in Fig. 10c and Fig. 10d, we select two intersections as examples, and they show that the simulated volumes can accurately follow the time-varying trends of ground truth, especially during non-peak hours. However, since no real-time information is involved in scratch sampling, the simulation performance is relatively poorer than conditional sampling. Moreover, traffic volumes from sampled trajectories are smoother compared with the ground truth. The reason is that we sample trajectories 100 times, so the generated traffic volumes can be regarded as the average of simulations to enhance reliability. In conclusion, these results show that the sampling strategy is both effective and accurate in exploring future varying patterns in the urban road network.

TABLE IV
PREDICTION COMPARISONS ON THE GEOLIFE BENCHMARK

	Acc@1	Acc@5	Acc@10	F1	MRR	NDCG@10
1-MMC	24.1	38.1	39.5	22.7	30.5	32.7
FPMC	24.0±0.6	53.7±2.0	57.8±0.4	13.5±0.6	35.5±0.7	40.8±0.6
LSTM	28.4±0.8	55.8±1.3	59.1±0.7	19.3±0.8	40.2±1.1	44.7±0.6
LSTM attn	29.8±0.7	54.6±1.5	58.2±1.7	21.3±0.8	40.7±0.4	45.0±0.7
DeepMove	26.1±0.8	54.2±0.8	58.7±0.6	18.9±0.4	38.2±0.2	42.6±0.5
MobTeast	29.5±0.6	51.3±0.7	56.2±1.0	17.3±0.6	39.3±0.4	43.4±0.9
MHSA	31.4±0.9	56.4±0.4	60.8±0.8	21.8±1.0	42.5±0.7	46.5±0.3
TrajTPP	**33.8±0.6**	**58.3±0.5**	**62.2±1.1**	**24.3±0.8**	**44.3±0.7**	**48.7±0.7**
Improvements	+7.64%	+3.37%	+2.30%	+11.47%	+4.24%	+4.73%

V. CONCLUSION

This study focuses on the next location and travel time prediction of individual vehicles in the urban road network. To adapt to the spatiotemporal patterns of vehicle trajectories, we combine deep learning models with intensity-free TPP and propose a TrajTPP framework to make probabilistic forecasting for the next trajectory. In this model, we design spatial and temporal mechanisms to explore the hidden patterns in previous trajectories. Then, a STGRU layer is further proposed to integrate the spatial and temporal features and enhance the autoregressive capabilities. Furthermore, we also propose a fusion module named GRAN to incorporate static information with dynamic features (i.e., previous locations and travel times). Afterwards, an intensity-free learning TPP is employed to model time-varying patterns of vehicle trajectories, and we define a prior transition probability to enhance next-location prediction performance. According to extensive experiments from the LPR dataset in Changsha China, TrajTPP performs superior over advanced deep learning methods and neural TPPs. Beyond the conventional prediction task, we design a sampling-based simulation experiment and find that the simulated results can effectively capture the spatiotemporal mobility patterns of traffic volumes in the urban road network.

Although our TrajTPP outperforms advanced baselines, there still remain several potential improvements in future studies. Firstly, compared with conventional deep learning methods, TPP has stronger interpretability, which induces several explainable applications, such as latent network discovery [57], Granger causality [58], etc. Therefore, future studies can explore this property in analyzing spatiotemporal dependencies among different intersections, which is essential in network-scale traffic prediction and control. Secondly, data-driven simulation [47] has gradually become a critical topic in human mobility modeling. This paper has demonstrated TrajTPP model can cheaply achieve accurate simulation. How to further enhance simulation performance is an intriguing topic for the following studies. Furthermore, relying on the powerful capability in discrete modeling, it is worth applying TPP-based methods to other types of traffic events modeling, such as traffic congestion, traffic accidents, etc.

APPENDIX

To further demonstrate the generalizability of TrajTPP in the classical mobility prediction task, we also employ the latest benchmark [26] on the Geolife dataset [59] for next-location prediction. To enhance fairness, we directly introduce the datasets and experiment results from the latest research [26] for evaluation. In this dataset, since there are more than 1000 candidate locations, and the spatial correlations in this dataset are relatively sparse and weak, the prior information learning module is not involved in this experiment. Meanwhile, because there are only 45 users in this dataset, we utilize an embedding layer to represent the user preference and integrate it with TrajTPP via the proposed GRAN module. Performance comparison of TrajTPP and baselines (i.e., 1-MMC [60], FPMC [50], LSTM, LSTM attn, DeepMove [51], MobTeast [61], and MHSA [26]) are shown in Table IV. More details of datasets and experiment settings on this benchmark can be found in Hong et al. [26] and our Github via https://github.com/SunderlandAJ-1130/TrajTPP.

From this table, we can observe that our proposed TrajTPP model can achieve significant superiority over the latest baselines. Specifically, compared to the most advanced MHSA model, our model can achieve 7.64% (i.e., from 31.4 to 33.8) and 11.47% (i.e., from 21.8 to 24.3) rises in Acc@1 and F1, respectively. These experiment results demonstrate that relying on its well-designed combination of TPP and deep learning model, TrajTPP is more capable than conventional deep learning models in mobility prediction.

REFERENCES

[1] J. Ding, H. Liu, L. T. Yang, T. Yao, and W. Zuo, “Multiuser Multivariate Multiorder Markov-Based Multimodal User Mobility Pattern Prediction,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4519–4531, 2020.

[2] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell, “Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior,” *UbiComp 2008 - Proc. 10th Int. Conf. Ubiquitous Comput.*, pp. 322–331, 2008.

[3] J. Liang, J. Tang, F. Liu, and Y. Wang, “Combining Individual Travel Preferences Into Destination Prediction: A Multi-Module Deep Learning Network,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–12, 2021.

[4] Q. Liu, S. Wu, L. Wang, and T. Tan, “Predicting the next location: A recurrent model with spatial and temporal contexts,” *30th AAAI Conf. Artif. Intell. AAAI 2016*, pp. 194–200, 2016.

[5] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, “Spatial-Aware Hierarchical Collaborative Deep Learning for POI Recommendation,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 11, pp. 2537–2551, 2017.

[6] M. Chen, Y. Zuo, X. Jia, Y. Liu, X. Yu, and K. Zheng, “CEM: A Convolutional Embedding Model for Predicting Next Locations,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3349–3358, 2021.

[7] J. Dai, B. Yang, C. Guo, and Z. Ding, “Personalized route recommendation using big trajectory data,” *Proc. - Int. Conf. Data Eng.*, vol. 2015-May, pp. 543–554, 2015.

[8] M. Chen, X. Yu, and Y. Liu, “MPE: a mobility pattern embedding model for predicting next locations,” *World Wide Web*, vol. 22, no. 6, pp. 2901–2920, 2019.

[9] P. Rathore, D. Kumar, S. Rajasegarar, M. Palaniswami, and J. C. Bezdek, “A Scalable Framework for Trajectory Prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3860–3874, 2019.

[10] D. Ashbrook and T. Starner, “Learning significant locations and predicting user movement with GPS,” *Proc. - Int. Symp. Wearable Comput. ISWC*, vol. 2002-Janua, pp. 101–108, 2002.

[11] J. Sun and J. Kim, “Joint prediction of next location and travel time from urban vehicle trajectories using long short-term memory neural networks,” *Transp. Res. Part C Emerg. Technol.*, vol. 128, no. March, p. 103114, Jul. 2021.

[12] Y. Liang and Z. Zhao, “NetTraj: A Network-Based Vehicle Trajectory Prediction Model with Directional Representation and Spatiotemporal Attention Mechanisms,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14470–14481, 2022.

[13] B. Yan, G. Zhao, L. Song, Y. Yu, and J. Dong, “PreCLN: Pretrained-based contrastive learning network for vehicle trajectory prediction,” *World Wide Web*, vol. 26, no. 4, pp. 1853–1875, 2023.

[14] G. Gidófalvi and F. Dong, “When and where next: Individual mobility prediction,” *Proc. 1st ACM SIGSPATIAL Int. Work. Mob. Geogr. Inf. Syst. MobiGIS 2012 - Conjunction with 20th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst. GIS 2012*, pp. 57–64, 2012.

[15] K. Krishna, D. Jain, S. V. Mehta, and S. Choudhary, “An LSTM Based System for Prediction of Human Activities with Durations,” *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–31, 2018.

[16] Z. Zang, X. Xu, K. Qu, R. Chen, and A. Chen, “Travel time reliability in transportation networks: A review of methodological developments,” *Transp. Res. Part C Emerg. Technol.*, vol. 143, no. February, p. 103866, 2022.

[17] A. B. Adege, H. P. Lin, and L. C. Wang, “Mobility Predictions for IoT Devices Using Gated Recurrent Unit Network,” *IEEE Internet Things J.*, vol. 7, no. 1, pp. 505–517, 2020.

[18] A. Asahara, K. Maruyama, A. Sato, and K. Seto, “Pedestrian-movement prediction based on mixed Markov-chain model,” *GIS Proc. ACM Int. Symp. Adv. Geogr. Inf. Syst.*, pp. 25–33, 2011.

[19] R. Simmons, B. Browning, Y. Zhang, and V. Sadekar, “Learning to predict driver route and destination intent,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 127–132, 2006.

[20] P. N. Pathirana, A. V. Savkin, and S. Jha, “Mobility modelling and trajectory prediction for cellular networks with mobile base stations,” *Proc. Int. Symp. Mob. Ad Hoc Netw. Comput.*, pp. 213–221, 2003.

[21] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, “WhereNext: A location predictor on trajectory pattern mining,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 637–645, 2009.

[22] K. Feng, J. C. Ji, Y. Zhang, Q. Ni, Z. Liu, and M. Beer, “Digital twin-driven intelligent assessment of gear surface degradation,” *Mech. Syst. Signal Process.*, vol. 186, no. November 2022, pp. 1–23, 2023.

[23] Q. Ni, J. C. Ji, and K. Feng, “Data-Driven Prognostic Scheme for Bearings Based on a Novel Health Indicator and Gated Recurrent Unit Network,” *IEEE Trans. Ind. Informatics*, vol. 19, no. 2, pp. 1301–1311, 2023.

[24] K. Feng, J. C. Ji, Q. Ni, and M. Beer, “A review of vibration-based gear wear monitoring and prediction techniques,” *Mech. Syst. Signal Process.*, vol. 182, no. January 2022, 2023.

[25] Y. Chen, N. Xie, H. Xu, X. Chen, and D. H. Lee, “A Multi-Context Aware Human Mobility Prediction Model Based on Motif-Preserving Travel Preference Learning,” *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 2, pp. 2139–2152, 2023.

[26] Y. Hong, Y. Zhang, K. Schindler, and M. Raubal, “Context-aware multi-head self-attentional neural network model for next location prediction,” *Transp. Res. Part C Emerg. Technol.*, vol. 156, no. August, p. 104315, Nov. 2023.

[27] Z. Zhao, H. N. Koutsopoulos, and J. Zhao, “Individual mobility prediction using transit smart card data,” *Transp. Res. Part C Emerg. Technol.*, vol. 89, no. December 2017, pp. 19–34, 2018.

[28] B. Mo, Z. Zhao, H. N. Koutsopoulos, and J. Zhao, “Individual Mobility Prediction in Mass Transit Systems Using Smart Card Data: An Interpretable Activity-Based Hidden Markov Approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12014–12026, 2022.

[29] G. Jin, L. Liu, F. Li, and J. Huang, “Spatio-Temporal Graph Neural Point Process for Traffic Congestion Event Prediction,” *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 14268–14276, 2023.

[30] S. Zhu, R. Ding, M. Zhang, P. Van Hentenryck, and Y. Xie, “Spatio-Temporal Point Processes With Attention for Traffic Congestion Event Modeling,” *IEEE Trans. Intell. Transp. Syst.*, no. 1, pp. 1–12, 2021.

[31] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, “Recurrent marked temporal point processes: Embedding event history to vector,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, vol. 13-17-Augu, pp. 1555–1564.

[32] G. Yang, Y. Cai, and C. K. Reddy, “Recurrent spatio-temporal point process for check-in time prediction,” *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 2203–2212, 2018.

[33] Y. Wu, Z. Cheng, and L. Sun, “Individual Mobility Prediction via Attentive Marked Temporal Point Processes,” *arXiv Prepr.*, Sep. 2021.

[34] O. Shchur, A. C. Türkmen, T. Januschowski, and S. Günnemann, “Neural Temporal Point Processes: A Review,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021, pp. 4585–4593.

[35] A. Vaswani *et al.*, “Attention Is All You Need,” *Adv. Neural Inf. Process. Syst.*, pp. 5998–6008, Jun. 2017.

[36] H. Ren *et al.*, “Combiner: Full Attention Transformer with Sparse Computation Cost,” *Adv. Neural Inf. Process. Syst.*, vol. 27, no. NeurIPS, pp. 22470–22482, 2021.

[37] B. Lim, S. Arık, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting,” *Int. J. Forecast.*, vol. 37, no. 4, pp. 1748–1764, 2021.

[38] W. Hua, Z. Dai, H. Liu, and Q. V. Le, “Transformer Quality in Linear Time,” in *Proceedings of the 39th International Conference on Machine Learning, PMLR*, 2022.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” *arXiv Prepr.*, 2016.

[40] D. R. So, W. Ma, H. Liu, Z. Dai, N. Shazeer, and Q. V Le, “Primer : Searching for Efficient Transformers for Language Modeling,” no. LM, pp. 1–34, 2021.

[41] C. Zheng, X. Fan, C. Wang, and J. Qi, “GMAN: A Graph Multi-Attention Network for Traffic Prediction,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, pp. 1234–1241, Apr. 2020.

[42] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, pp. 1–14, 2016.

[43] Z. Zhou, D. S. Matteson, D. B. Woodard, S. G. Henderson, and A. C. Micheas, “A Spatio-Temporal Point Process Model for Ambulance Demand,” *J. Am. Stat. Assoc.*, vol. 110, no. 509, pp. 6–15, 2015.

[44] K. Dascher-Cousineau, O. Shchur, E. E. Brodsky, and S. Günnemann, “Using Deep Learning for Flexible and Scalable Earthquake Forecasting,” *Geophys. Res. Lett.*, vol. 50, no. 17, pp. 1–9, 2023.

[45] O. Shchur, M. Biloš, and S. Günnemann, “Intensity-Free Learning of Temporal Point Processes,” in *Iclr*, 2019, no. 2013, pp. 1–9.

[46] G. Zou, Z. Lai, C. Ma, Y. Li, and T. Wang, “A novel spatio-temporal generative inference network for predicting the long-term highway traffic speed,” *Transp. Res. Part C Emerg. Technol.*, vol. 154, no. July, 2023.

[47] J. Sun and J. Kim, “Toward Data-Driven Simulation of Network-Wide Traffic: A Multi-Agent Imitation Learning Approach Using Urban Vehicle Trajectory Data,” *IEEE Trans. Intell. Transp. Syst.*,

vol. PP, pp. 1–13, 2023.

[48] O. Shchur, "Modeling Continuous-time Event Data with Neural Temporal Point Processes," *PHD thesis*, 2022.

[49] H. Chen, C. Yang, and X. Xu, "Clustering Vehicle Temporal and Spatial Travel Behavior Using License Plate Recognition Data," *J. Adv. Transp.*, vol. 2017, pp. 1–14, 2017.

[50] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," *Proc. 19th Int. Conf. World Wide Web, WWW '10*, pp. 811–820, 2010.

[51] J. Feng *et al.*, "DeepMove: Predicting human mobility with attentional recurrent networks," *Web Conf. 2018 - Proc. World Wide Web Conf. WWW 2018*, vol. 2, pp. 1459–1468, 2018.

[52] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 2341–2347, 2018.

[53] Y. Luo, Q. Liu, and Z. Liu, "STAN: Spatio-temporal attention network for next location recommendation," *Web Conf. 2021 - Proc. World Wide Web Conf. WWW 2021*, pp. 2177–2185, 2021.

[54] U. Upadhyay, A. De, and M. Gomez-Rodrizuez, "Deep reinforcement learning of marked temporal point processes," *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, pp. 3168–3178, 2018.

[55] T. Omi, N. Ueda, and K. Aihara, "Fully neural network based model for general temporal point processes," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. 2016, 2019.

[56] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao, "LibCity: An Open Library for Traffic Prediction," *GIS Proc. ACM Int. Symp. Adv. Geogr. Inf. Syst.*, pp. 145–148, 2021.

[57] S. W. Linderman and R. P. Adams, "Discovering latent network structure in point process data," *31st Int. Conf. Mach. Learn. ICML 2014*, vol. 4, pp. 3268–3281, 2014.

[58] W. Zhang, T. K. Panum, S. Jha, P. Chalasani, and D. Page, "CAUSE: Learning granger causality from event sequences using attribution methods," *37th Int. Conf. Mach. Learn. ICML 2020*, vol. PartF16814, pp. 11171–11181, 2020.

[59] Y. Zheng, Y. Chen, X. Xie, and W. Y. Ma, "GeoLife2.0: A location-based social networking service," *Proc. - IEEE Int. Conf. Mob. Data Manag.*, no. 49, pp. 357–358, 2009.

[60] S. Gambs, M. O. Killijian, and M. N. Del Prado Cortez, "Next place prediction using mobility Markov chains," *Proc. 1st Work. Meas. Privacy, Mobility, MPM'12*, pp. 0–5, 2012.

[61] H. Xue, F. D. Salim, Y. Ren, and N. Oliver, "MobTCast: Leveraging Auxiliary Trajectory Forecasting for Human Mobility Prediction," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 30380–30391, 2021.

Jie Zeng received the B.S. and M.S. degree in transportation engineering from Central South University in 2021 and 2024, respectively. He is currently working toward the PhD degree at the Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology. His current research interests include travel demand management, spatiotemporal data mining, and intelligent transportation systems. He serves as a reviewer for the journals of IEEE Transactions on Intelligent Transportation Systems, ACM Computing Surveys, Cities, etc.

Chenxi Xiao received the B.S. degree in transportation engineering from Central South University in 2023. She is currently working toward the Ph.D. degree in the School of Traffic and Transportation Engineering, Smart Transport Key Laboratory of Hunan Province, Central South University, Changsha, China. Her research interests include travel chain research, multimodal transportation, travel destination prediction, network traffic modeling and control, and so on.

Jinjun Tang received his Ph.D. (2016) degree in transportation engineering at the Harbin Institute of Technology, Harbin, China. He is currently a Professor with the School of Traffic and Transportation Engineering, Central South University, Changsha, China. From 2014 to 2016, he was a visiting scholar at the Smart Transportation Applications and Research Laboratory (STAR Lab) of the University of Washington, Seattle, WA, USA. He published more than 60 technical papers in the journal as the first author and corresponding coauthor. His research interests include traffic flow prediction, data mining in the transportation systems, intelligent transportation systems, and transportation modeling.

Cheng Hu received the B.S. degree in transportation engineering from China University of Mining and Technology in 2014; the M.S. degree in transportation engineering from Central South University, in 2018. He is currently working toward the Ph.D. degree in the School of Traffic and Transportation Engineering, Smart Transport Key Laboratory of Hunan Province, Central South University, Changsha, China. His research interests include traffic safety modeling, road network optimization, dynamic traffic assignment, traffic flow theory, network traffic modeling and control, and so on.