Synthesizing Human Trajectories Based on Variational Point Processes

Huandong Wang, *Member, IEEE*, Qizhong Zhang, Yuchen Wu, Depeng Jin, *Member, IEEE*, Xing Wang, Lin Zhu, and Li Yu

***Abstract*—Synthesized human trajectories are instrumental for a large number of applications. However, existing trajectory synthesizing models are limited in either modeling variable-length trajectories with continuous temporal distribution or incorporating multi-dimensional context information. In this paper, we propose a novel probabilistic model based on the variational temporal point process to synthesize human trajectories. This model combines the classical temporal point process with the novel neural variational inference framework, leading to its strong ability to model human trajectories with continuous temporal distribution, variable length, and multi-dimensional context information. Extensive experimental results on two real-world trajectory datasets show that our proposed model can synthesize trajectories most similar to real-world human trajectories compared with four representative baseline algorithms in terms of a number of usability metrics, demonstrating its effectiveness.**

***Index Terms*—Generative models, mobility trajectory, temporal point process, variational auto-encoder.**

I. Introduction

HUMAN trajectory synthesizing models are crucial for many applications. For example, in mobile ad hoc networks or cellular networks, based on synthesized trajectories, we can simulate the detailed process of movement and communication of network users to have a reliable performance analysis of the network [1]. Similarly, we can simulate the traffic congestion based on synthesized trajectories, which are also instrumental for urban planning [2].

Manuscript received 27 May 2021; revised 26 April 2022; accepted 18 June 2023. Date of publication 5 September 2023; date of current version 8 March 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFA0711403, in part by the National Natural Science Foundation of China under Grants 62171260, U22B2057, and 61972223, and in part by Young Elite Scientists Sponsorship Program by CIC under Grant 2021QNRC001. Recommended for acceptance by Jae-Gil Lee. *(Huandong Wang and Qizhong Zhang contributed equally to this work.) (Corresponding author: Huandong Wang.)*

Huandong Wang and Depeng Jin are with Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: wanghuandong@mail.tsinghua.edu.cn; jindp@tsinghua.edu.cn).

Qizhong Zhang is with the Department of Statistics, University of Chicago, Chicago, IL 60637 USA (e-mail: zhangqz1922@uchicago.edu).

Yuchen Wu is with the Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: yuchenwu@andrew.cmu.edu).

Xing Wang, Lin Zhu, and Li Yu are with China Mobile Research, Tsinghua University-China Mobile Communications Group Company Ltd. Joint Institute, Beijing 100084, China (e-mail: wangxing@chinamobile.com; zhulinyj@chinamobile.com; yuliyf@chinamobile.com).

The code and datasets are available at https://github.com/tsinghua-fib-lab/TrajSynVAE.

Digital Object Identifier 10.1109/TKDE.2023.3312209

Existing trajectory synthesizing methods can be divided into two categories, i.e., model-based methods and data-based methods. Model-based methods usually make strong assumptions about human movement behavior, leading to their weakness in incorporating multi-dimensional external factors that may influence the decision-making process of human mobility. For example, the TimeGeo model [3] assumes users' mobility states only depend on whether they are at home. Users' future movements are further assumed to be determined by their current states, the current time, and the number of visited locations. Other factors (e.g., more fine-grained semantic location context) are ignored and hard to be considered under the assumptions of this model.

On the other hand, data-based models adopt powerful deep learning techniques to model users' mobility behavior. Giving up building mobility models with explicit physical meaning, these methods seek to directly learn from data to achieve better performance. However, the data-irregular nature of human trajectories causes big challenges for them. For example, the waiting time of users at each location is distributed in a continuous space, with the range from tens of minutes to tens of hours. In addition, the number of location records of different trajectories is also diverse for different users. Most existing deep learning based trajectory synthesizing methods adopt Generative Adversarial Networks (GAN), of which the discriminator requires inputs with discrete temporal distribution and fixed lengths. In order to solve this problem, existing studies compromise to preprocess human trajectories by discretizing the timestamp and cutting the whole trajectories into many fixed-length sub-trajectories [4], [5], leading to their disadvantage in terms of modeling the fine-grained behavior and long-term correlation of the human mobility.

In order to overcome the limitations of existing methods, we propose a novel probabilistic model based on the variational temporal point process to synthesize human trajectories. This model combines the classical temporal point process with the deep neural network based on the novel variational inference framework, where the temporal point process is utilized as the bridge between uninterpretable neurons in the neural network and real-world human trajectories. Specifically, in our proposed model, human trajectories are generated from probabilistic distributions with explicit physical meaning, whose parameters are the output of the neural network. In this way, our proposed model retains the strong interpretability of the classical temporal point process model. In addition, various context information is considered in calculating these parameters. Specifically, in terms of the spatial context, semantic information of the visited location is considered by incorporating their point of interests (PoI) distribution. In terms of the temporal context, the joint impact of the visiting time and waiting time on the mobility behavior is considered. Finally, the variational inference framework is utilized for training the proposed model, which defines a loss function by comparing the probability distribution of the latent codes describing users' decision tendencies in the future movements obtained from the generated trajectories and the real trajectories. Thus, our proposed model does not require fixed-length trajectories with discrete timestamps as in the GAN-based methods, leading to its strong ability to model the fine-grained behavior and long-term correlation of human mobility. In summary, our paper makes the following contributions:

- We propose a trajectory synthesizing framework based on the variational temporal point process, which combines the advantages of model-based methods and data-based methods. Specifically, it inherits the strong interpretability of model-based methods and also takes full advantage of the data-based deep neural network to directly learn from data.
- We propose a novel interaction mechanism between our proposed model and trajectory data combining classical probabilistic distributions in the temporal point process and powerful encoding techniques in neural networks. Specifically, we propose a Fourier-based positional encoder to extract fine-grained temporal information in the continuous domain. The spatial distribution is also modeled in a multi-view manner by incorporating semantic location context.
- We conduct extensive experiments on four real-world mobility datasets. Results show that compared with five representative baseline algorithms, our model is able to synthesize trajectories most similar to real trajectories in terms of a number of key usability metrics. Specifically, the difference measured by JSD is reduced by 56.5% on average.

II. Preliminaries and Problem Overview

In this section, we first define the notations used in this paper. Then, based on these notations, we present an overview of the trajectory synthesizing system and formally define the trajectory synthesizing problem.

A. Mathematical Setup

We define $\mathcal{U}$ as the set of all users. The trajectory of each user is defined by a set of mobility records R^u. Each mobility record is defined as a three-tuple $r_i^u = (t_i^u, l_i^u, \tau_i^u)$, indicating user u visits the location l_i^u at time t_i^u and stay there for the time of τ_i^u. Further, we define N_u as the number of mobility records of user u. Thus, we have $R^u = \{r_1^u, \ldots, r_{N_u}^u\}$.

Specifically, the visiting time t_i^u and waiting time τ_i^u are all defined on real positive number space. In terms of the location l_i^u, in order to prevent the leakage of sensitive information of users and process the trajectory data and external datasets (e.g., semantic information of locations) in a cohesive manner, locations are divided into geographic regions. We further define $\mathcal{L}$ as the set of all regions. For each region $l \in \mathcal{L}$, its semantic information is characterized by the point of interests (PoIs) located in this region. Specifically, a PoI represents a location point with a certain function such as a restaurant or a shopping mall. We define $\mathcal{C}$ as all the PoI categories (functions). Then, the semantic information of l is characterized by a $|\mathcal{C}|$-sized vector $\boldsymbol{I}_l$, where each element $I_l(c)$ represents the number of PoIs of category $c \in \mathcal{C}$ located in the region l.

For readability, we summarize the major notations used throughout the paper in Table I.

TABLE I
List of Commonly Used Notations

Notation	Description
$\mathcal{U}$	The set of all users.
$\mathcal{L}$	The set of geographical regions.
$\mathcal{C}$	The set of PoI categories.
$\boldsymbol{I}_l(c)$	The number of PoIs of category $c \in \mathcal{C}$ at region $l \in \mathcal{L}$.
$\boldsymbol{I}$	The global PoI matrix with size $|\boldsymbol{C}| \times |\boldsymbol{L}|$.
r_i^u	The i-th records for user u.
t_i^u	The timestamp of r_i^u.
l_i^u	The location of r_i^u.
τ_i^u	The waiting time of r_i^u.
z_i^u	Latent code describing the decision tendency of user u corresponding to r_i^u.
R^u	The trajectories of user u.
N^u	The number of records for user u.
θ	The set of parameters for the generation process of our proposed model.
ϕ	The set of parameters for the inference process of our proposed model.
T	The time span of synthesized trajectories.

B. Problem Definition

Based on the above mathematical setup, the trajectory synthesizing problem investigated in this paper can be formally defined as follows.

Trajectory Synthesizing Problem

Given: The trajectories R^u of user $u \in \mathcal{U}$, and the PoI distribution $\boldsymbol{I}_l(c)$ of all regions $l \in \mathcal{L}$.

Problem: Synthesizing new trajectories that are different from the original trajectories and preserve the original trajectory dataset's usability, which indicates that the synthesized trajectories should statistically resemble the original trajectories.

The trajectory synthesizing problem has a number of unique characteristics and challenges compared with modeling common sequence data. Specifically, the waiting time or staying time of users at different locations is an important aspect of human trajectories, which common sequence data do not have. In addition, there are hidden semantics behind the locations visited by users. For example, the urban function of the locations is able to characterize the intention of users to a large extent. On the other hand, most existing trajectory modeling approaches are based on deterministic models (e.g., standard RNN). That is, given a fixed input, the output is the same every time we run the network. Thus, these approaches fail to capture the uncertainty of the mobility behavior [6], [7], [8]. Overall, in order to overcome these unique challenges of trajectory modeling and synthesizing, we propose a novel probabilistic model based on the variational temporal point process. This model combines the classical temporal point process with the deep neural network based on the novel variational inference framework, where the temporal point process is utilized as the bridge between uninterpretable neurons in the neural network and real-world human trajectories. Before introducing our proposed model in detail, we first present the necessary preliminary knowledge about the variational auto-encoder and temporal point process in the following part of this section.

C. Variational Auto-Encoder

Auto-Encoder (AE) refers to a category of unsupervised deep learning methods that are used to find the corresponding embedding vector z of the given target object x (e.g., images, goods) to efficiently characterize their features. AE methods are mainly composed of two components, namely encoder and decoder. The encoder is used to map the target object x into the embedding vector z, while the decoder is used to reconstruct the target object based on the embedding vector z. By denoting the reconstructed object as x', the neural network of AE is learned by minimizing the reconstruction error between x and x'. However, AE methods are limited in two aspects. First, these methods cannot be applied to format-irregular datasets, e.g., the trajectory datasets. Specifically, the number of mobility records is diverse for different trajectories. Thus, it is not easy to define a reasonable "reconstruction error" between trajectories. Second, most AE methods are deterministic models. The lack of randomness indicates these methods are not suitable for the trajectory synthesizing tasks.

As a variant of AE methods, Variational Auto-Encoder (VAE) overcomes these limitations by combining the neural network with the Bayesian probabilistic framework through variational inference techniques. First, the encoder and decoder in VAE are all defined in the form of probabilistic distributions described by neural networks. Specifically, we denote them as $q_\phi(z|x)$ and $p_\theta(x|z)$, where θ and ϕ are their corresponding parameters, i.e., neural network weights. In addition, a prior distribution $p_\theta(z)$ is introduced to the latent code z. In this way, randomness is introduced to the model, which is able to solve the deterministic limitation of the AE models. Second, VAE is not trained by minimizing the reconstruction error. Instead, it minimizes the distance between the distributions of the latent code z obtained based on the encoder and the decoder, which is expressed by $\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))$. Specifically, $q_\phi(z|x)$ has already been modeled by the encoder network. As for $p_\theta(z|x)$, based on the Bayes formula, we have $p_\theta(z|x) = p_\theta(x|z)p_\theta(z)/p_\theta(x)$. Further, we can derive the following equation:

$$\begin{aligned}&\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))\\&\quad=\log p_\theta(x)+\int q_\phi(z|x)\log\frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|x)}dz,\end{aligned} \tag{1}$$

where KL is the Kullback-Leibler divergence, which is larger than 0. As we can observe, $\log p_\theta(x)$ does not change with the neural network weights. Thus, we can convert the optimization target of minimizing $\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))$ to maximizing the second term on the right hand of (1). Specifically, we denote it as $\mathcal{L}(\theta, \phi)$, which also can be represented as follows:

$$\mathcal{L}(\theta,\phi) = E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p_\theta(z)). \tag{2}$$

Then, we can optimize the VAE model by maximizing $\mathcal{L}$. Further, a reparameterization trick is proposed to update the neural network weights [9] to sample from the probability distributions and optimize parameters θ and ϕ in the process of maximizing $\mathcal{L}$. In this way, the reconstruction error is not required anymore. Thus, the VAE method has a stronger ability to model the format-irregular trajectory data with variable length and continuous temporal distribution.

D. Temporal Point Process

The temporal point process is a category of stochastic processes that are utilized to describe time-tagged sequential data points with different types [10]. Two well-known temporal point processes are the Poisson process and Hawkes process [11].

In the temporal point process, the intensity function $\lambda(t)$ is utilized to describe the probability density of a data point occurring around time t. Specifically, in the Poisson process, the inter-point time gap follows the exponential distribution with a constant parameter λ_0, of which the intensity function can be expressed as follows:

$$\lambda(t) = \lambda_0. \tag{3}$$

Differently, in the Hawkes process an occurred data point will influence the intensity function of future data points, which can be expressed as follows:

$$\lambda(t) = \mu_0 + \sum_{t_i<t} \alpha \cdot \phi(t - t_i), \tag{4}$$

where t_i is the happening time of the past data points before t, and μ_0 is the basic intensity. ϕ is a kernel function that describes the impact of the occurred data points on the current value of the intensity function.

In general, the intensity function of the temporal point process at time t is only affected by the past data points due to the law of causality. Thus, we denote it as $\lambda(t|x_{1:n-1})$ for $t \in [t_{n-1}, \infty)$, where $x_{1:n-1}$ represents the past $n-1$ data points. Then, the probability distribution function of t_n can be expressed as follows:

$$p(t_n|x_{1:n-1}) = \lambda(t_n|x_{1:n-1})e^{-\int_{t_{n-1}}^{t_n}\lambda(t|x_{1:n-1})dt}. \tag{5}$$

The Poisson process and Hawkes process are all special cases of it with different intensity functions as shown in (3) and (4). However, it is not easy to obtain an accurate expression of the intensity function. For example, in Hawkes process, ϕ is usually set to be an exponential decay function as $\phi(t - t_i) = e^{\beta(t-t_i)}$. However, whether the intensity functions of these forms can well fit the real sequential data cannot be guaranteed. In addition, the complicated impact of the multiple categories of data points on the intensity function further exacerbates the difficulties in

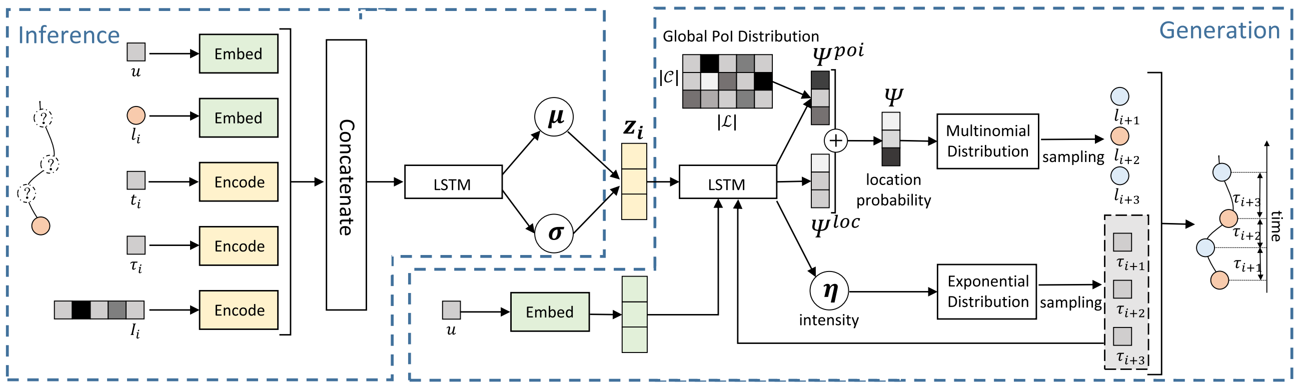


Fig. 1. Framework of the proposed variational trajectory synthesizing model.

accurately modeling them. Instead, in the trajectory synthesizing problem, we introduce the powerful deep recurrent neural networks to estimate the intensity function, which has been proven to have a strong modeling ability in a number of practical applications [12], [13], [14], [15], [16], [17], [18], [19].

III. Trajectory Synthesizing Model Based on Variational Temporal Point Process

In order to overcome the limitation of existing trajectory synthesizing methods in terms of variable-length trajectory with continuous temporal distribution and incorporating multi-dimensional context information, we propose a probabilistic model based on the variational temporal point process. This model combines the classical temporal point process with the deep neural network, which inherits the strong ability of the temporal point process to model continuous temporal distribution and also takes full advantage of the strong modeling ability of the deep neural network.

We show the framework of our proposed variational trajectory synthesizing model in Fig. 1. As we can observe, our proposed trajectory synthesizing model can be divided into two parts, i.e., generation and inference, with the latent code z_i as the bridge between them, which encodes the stochastic decision-making features of users. For example, after work, a commuter may go home immediately, or go to a bar with a certain probability. We aim to encode his/her decision tendency in the latent code z_i, which is able to model the randomness of human mobility. Then, the inference module describes the process of estimating the probability distribution of z_i based on historical mobility records $r_{1:i}$, while the generation module describes the process of synthesizing new mobility records given historical latent codes $z_{1:i}$ and historical mobility records $r_{1:i-1}$.

Specifically, two separate Long Short Term Memory (LSTM) networks are utilized to model the sequence of users' mobility records and latent codes in two parts, respectively. Further, we elaborately design embedding and encoding mechanisms in the inference module of our model to extract sufficient information from the mobility records in terms of spatial, temporal, and semantic dimensions to learn the latent code z_i. Specifically, we propose a novel Fourier-based positional encoder to extract fine-grained temporal information in the continuous domain. Then, in the generation module of our model, based on the concept of the temporal point process, we design a probability-based generation process to synthesize mobility trajectories with explicit physical meaning in a multi-view manner. Overall, by combining the neural networks and the temporal point process, our proposed method is able to incorporate multi-dimensional information and synthesize trajectories with continuous temporal distribution and arbitrary trajectory length, which overcomes the limitations of existing model-based and data-based methods.

In the following part of this section, we will introduce the generation and inference process of our proposed model, respectively. Then, we introduce how to learn parameters in our model. Note that our model is designed to synthesize several continuous trajectories with arbitrary length or stop conditions for one given user each time. In addition, different users share the same neural network weight in our model. For the sake of simplicity, we omit the superscript u in this section.

A. Inference

The goal of the inference process is to obtain the probability distribution of latent codes z_i based on historical mobility records $r_{1:i}$. That is, learning an approximate distribution $q(z_i|r_{1:i})$ to estimate the posterior distribution of z_i directly based on observed mobility record $r_{1:i}$. For the sake of simplicity, we denote the set of all neural network parameters in the inference module as ϕ.

In order to extract sufficient information from $r_{1:i}$, the user ID u, waiting time τ_i, visiting time t_i, visited location l_i, and the PoI distribution I_i are all considered. First, the visited location l_i and the user ID u are embedded into representative vectors l_i^e and u^e based on embedding modules, respectively. Next, the visiting time t_i is encoded based on the positional encoding techniques [20], [21] to extract the fine-grained temporal features. However, the classical positional encoding does not well capture the periodic behavior of human mobility. Thus, we propose a Fourier-based positional encoder $\mathbf{PE}(\cdot)$ as follows:

$$\begin{cases} \mathrm{PE_{2i}(t)} = \sin(2\pi \mathrm{it}/\gamma), \\ \mathrm{PE}_{2i+1}(t) = \cos(2\pi it/\gamma), \end{cases} \tag{6}$$

where $\mathrm{PE}_i(t)$ represents the ith element of the positional encoding of the absolute time or time difference t. γ is the fundamental frequency. In our model, the same time of different days is expected to be encoded into similar vectors. Thus, we set γ as one day in our model. In this way, the periodic behavior of human mobility is well captured. Specifically, we denote the positional encodings of t_i as t_i^e.

As for the waiting time τ_i, we encode it by applying a logarithmic transformation followed by a linear layer, i.e., $\tau_i^e = \text{Linear}(\log \tau_i)$. Then, PoI distribution $\boldsymbol{I}_i$ is a $|\mathcal{C}|$-sized vector with each element $I_i(c) = I_{l_i}(c)$ representing the number of PoIs of category c in region l_i. We transform it into an embedding vector $\boldsymbol{I}_i^e$ where each element $I_i^e(c)$ has values ranging from 0 to 1. This is achieved by transforming each element $I(c)$ into the percentage of regions with fewer than $I(c)$ PoIs of category c. Specifically, we denote this transformation as:

$$\boldsymbol{I}_i^e = \mathrm{F}(\boldsymbol{I}_i). \tag{7}$$

Finally, t_i^e, τ_i^e, l_i^e, u^e, and $\boldsymbol{I}_i^e$ are concatenated to obtain the embedding of mobility record r_i, which is denoted as r_i^e. Then, r_i^e is fed into an LSTM as follows:

$$\begin{cases} r_i^e = [t_i^e; \tau_i^e; l_i^e; u^e], \\ h_i = \text{LSTM}_\phi(h_{i-1}, r_i^e). \end{cases} \tag{8}$$

Note that in this process, the visiting time t_i and waiting time τ_i are redundant with each other. Specifically, with a given initial time, we can reconstruct $t_{1:i}$ from $\tau_{1:i}$ and vice versa. However, we still feed both of them into the neural network. The main reason is that we cannot guarantee that the neural network can correctly learn the features in terms of both t_i and τ_i only based on one of them. Thus, we alternatively utilize the redundancy between t_i and τ_i to better encode their features in the hidden state variable h_i, which paves the way for modeling their dependency between the mobility feature z_i. Specifically, the information of mobility records $r_{1:i}$ are all encoded into the current state vector h_i. Thus, our model utilizes h_i to calculated the estimated posterior distribution of z_i as follows:

$$\begin{cases} [\mu_\phi; \sigma_\phi] = \text{MLP}_\phi(h_i), \\ q_\phi(z_i|r_{1:i}) = \mathcal{N}(\mu_\phi, \sigma_\phi^2). \end{cases} \tag{9}$$

B. Generation

The generation process in our model describes the process of synthesizing the next mobility record r_i given historical latent codes $z_{1:i}$ and historical mobility records $r_{1:i-1}$. For the sake of simplicity, we denote the set of all neural network parameters in the generation module as θ.

In the three components of r_i, we only need to generate τ_i and l_i, since t_i can be obtained by $t_i = t_{i-1} + \tau_{i-1}$. Inspired by the temporal Poisson process (e.g., Poisson process, Hawkes process), we model the probability distribution of τ_i and l_i as follows:

$$\begin{cases} p(\tau_i|z_{1:i}, r_{1:i-1}) = \lambda(z_{1:i}, r_{1:i-1}) \cdot e^{-\lambda(z_{1:i}, r_{1:i-1})\tau_i}, \\ p(l_i|z_{1:i}, r_{1:i-1}) = \Psi_{l_i}(z_{1:i}, r_{1:i-1}), \end{cases} \tag{10}$$

where τ_i and l_i are modeled by the exponential distribution and the multinomial distribution, respectively. In addition, $\Psi_l(z_{1:i}, r_{1:i-1})$ indicates the visiting probability of each location $l \in \mathcal{L}$. Thus, we further have $\sum_{l\in\mathcal{L}} \Psi_l(z_{1:i}, r_{1:i-1}) = 1$. At the same time, $\lambda(z_{1:i}, r_{1:i-1})$ represents the intensity of the next movement.

In order to model the long-term complicated correlation between the next mobility record and $(z_{1:i}, r_{1:i-1})$, we utilize another LSTM network parameterized by θ. Specifically, the current mobility feature z_i is concatenated with the embedding of the last mobility record r_{i-1}^e. Then, they are fed into the LSTM network as follows:

$$\begin{cases} r_{i-1}^e = [\mathbf{PE}(t_{i-1} + \tau_{i-1}); u^e], \\ h_i = \text{LSTM}_\theta(h_{i-1}, z_i, r_{i-1}^e), \end{cases} \tag{11}$$

where h_{i-1} and h_i represents the recurrent hidden state of the LSTM. Note that different from (8) of the encoder network, the waiting time τ_{i-1} is directly added into the visiting time t_{i-1} without been considered separately. Specifically, in (8) we aim to model the correlation between users' current mobility record r_i and current latent code z_i through h_i, which are strongly correlated. However, in (11), we only need to model the correlation between r_{i-1} and r_i through h_i, where the waiting time τ_{i-1} plays an relatively unimportant role. Thus, we ignore it in (11).

Then, the current hidden state h_i is utilized to generate τ_i and l_i. Specifically, our model feeds h_i into a multi-layer perceptron to obtain a scalar η, an $|\mathcal{L}|$-sized vector ξ^{loc}, and a $|\mathcal{C}|$-sized vector ξ^{poi} as follows:

$$[\eta, \Psi^{loc}, \xi^{poi}] = \text{MLP}_\theta(h_i). \tag{12}$$

In terms of generating waiting time τ_n, η is used as the intensive of this temporal point process. That is,

$$\lambda(z_{1:i}, r_{1:i-1}) = \eta. \tag{13}$$

Then, the waiting time τ is drawn from (10).

In terms of generating locations, ξ^{loc} is used to directly characterize the probability of visiting different locations. ξ^{poi} represents the "intention" of the movement, i.e., the probability of visiting different PoI categories. Then, the global PoI distribution embedding matrix $\boldsymbol{I}^e = [\boldsymbol{I}_1^e, \boldsymbol{I}_2^e, \ldots, \boldsymbol{I}_{|\mathcal{L}|}^e] \in R^{|\mathcal{C}|\times|\mathcal{L}|}$ is used to map the "intention" to concrete locations as follows:

$$\Psi^{poi} = (\xi^{poi} \odot W)^T \cdot \boldsymbol{I}^{\mathbf{e}}, \tag{14}$$

where $\odot$ is the Hadamard product operator, and $W \in R^{|\mathcal{C}|}$ is a learnable weight vector. Then, the obtained Ψ^{poi} is an $|\mathcal{L}|$-sized column vector.

Finally, the weighted sum of Ψ^{poi} and Ψ^{loc} is used as the final probability distribution of l_i based on (10), which can be expressed as follows:

$$\Psi(z_{1:i}, r_{1:i-1}) = \alpha\Psi^{loc} + \beta\Psi^{poi}. \tag{15}$$

Note that the inference process and the generation process are not symmetric with each other. Specifically, in the inference process, z_i is estimated only based on $r_{1:i}$, while in the generation process, r_i is not only dependent on $z_{1:i}$ but also dependent on $r_{1:i-1}$, which constitutes a feedback loop in the generation network. That is, the output of the ith recurrent computation, i.e., r_i will be used as the input for the $(i+1)$th recurrent computation. We stress that the feedback is important. Since $z_{1:i}$ only contains the information about users' decision tendency,

Algorithm 1: Training Process of the Proposed Model.

```
Input: ${R^u}_{u\in\mathcal{U}}$
Initialize: Randomly initialise $\theta$, $\phi$.
for $epoch \in N_{epoch}$ do
    for $u \in \mathcal{U}$ do
        $R^u_M \leftarrow$ Random minibatch of $M$ continuous records drawn from $R^u$;
        $\boldsymbol{\epsilon}^u_M \leftarrow$ Sampled from $\mathcal{N}(0,1)$;
        $g \leftarrow \nabla_{\theta,\phi}\widetilde{L}^M(\theta,\phi,R^u_M,\boldsymbol{\epsilon}^u_M,u)$;
        $\theta$,$\phi \leftarrow$ Update parameters using gradients $g$;
Output: $\theta$, $\phi$
```

Algorithm 2: Synthesizing Trajectories Based on the Proposed Model.

```
Input: $\theta$, $\Phi$, $\mathcal{U}$
for $u \in \mathcal{U}$ do
    $i \leftarrow 1$;
    $\widetilde{R}^u \leftarrow \emptyset$;
    $\tilde{t}^u_{i-1} \leftarrow 0$;
    $\tilde{\tau}^u_{i-1} \leftarrow$ Sampled from $p^u(t)$;
    $\tilde{t}^u_i = \tilde{t}^u_{i-1} + \tilde{\tau}^u_{i-1}$;
    while $\tilde{t}^u_i < T$ do
        $z^u_i \leftarrow$ Sampled from $p_\theta(z)$
        Calculate $\Psi^u_i$,$\lambda^u_i$ based on (6)-(11);
        $\tilde{l}^u_i \sim$Multinormial$(\cdot|\Psi^u_i)$;
        $\tilde{\tau}^u_i \sim$Exponential$(\cdot|\lambda^u_i)$;
        $\tilde{t}^u_{i+1} = \tilde{t}^u_i + \tilde{\tau}^u_i$;
        $\widetilde{R}^u \leftarrow \widetilde{R}^u \cup \{(\tilde{t}^u_i, \tilde{l}^u_i, \tilde{\tau}^u_i)\}$;
        $i \leftarrow i+1$;
Output: ${\widetilde{R}^u}_{u\in\mathcal{U}}$
```

while the final decision including waiting time and staying time can be only reflected by $r_{1:i-1}$, which has a critical impact on the future mobility behavior. Thus, the feedback loop is important for the performance of the trajectory synthesizing model, which will be evaluated in the experiment section.

C. Parameter Learning

Following the learning strategy in the variational inference framework [9], we train our proposed model by maximizing the evidence lower bound of the trajectory, which can be represented as follows:

$$\sum_{u\in\mathcal{U}}\mathcal{L}_{\theta,\phi}(r^u_{1:N_u}) = \sum_{u\in\mathcal{U}}\sum_{i=1}^{N_u} E_{q_\phi(z^u_i|r^u_{1:i})}[\log p_\theta(r^u_i|z^u_{1:i})] - \mathrm{KL}(q_\phi(z^u_i|r_{1:i})||p_\theta(z)). \tag{16}$$

However, calculating the integration in (16) is intractable in practice. In order to solve this problem, following the common solution of variational inference based methods, we draw samples from $q_\phi(z^u_i|r^u_{1:i})$, and then approximate (16) based on the following equation:

$$\sum_{u\in\mathcal{U}}\mathcal{L}_{\theta,\phi}(r^u_{1:N_u}) \approx \sum_{u\in\mathcal{U}}\sum_{i=1}^{N_u}[\log p_\theta(r^u_i|\widetilde{z}^u_{1:i})p_\theta(\widetilde{z}^u_i) - \log(q_\phi(\widetilde{z}^u_i|r^u_{1:i}))], \tag{17}$$

where $\widetilde{z}^u_i$ is the sample of z^u_i drawn from $q_\phi(z^u_i|r^u_{1:i})$.

Further, the reparameterization trick [9] is utilized to decouple the parameter ϕ from $\widetilde{z}^u_i \sim q_\phi(z^u_i|r^u_{1:i})$ for better parameter estimation based on (9) as follows:

$$\begin{cases}\epsilon^u_i \leftarrow \mathcal{N}(0,1),\\ \widetilde{z}^u_i = \epsilon^u_i\cdot\sigma_\phi(h^u_i) + u_\phi(h^u_i).\end{cases} \tag{18}$$

We denote the approximation of the loss function (16) in the right hand of (17) as $\widetilde{L}(\theta,\phi,R^u,\boldsymbol{\epsilon}^u,u)$. Note that the reparameterization trick has been adopted in calculating $\widetilde{L}(\theta,\phi,R^u,\boldsymbol{\epsilon}^u,u)$.

Further, in each iteration, we randomly sample a sub-trajectory R^u_M with M continuous mobility records from R^u for each user $u\in\mathcal{U}$. Then, we calculate the loss function based on the sub-trajectories as follows:

$$\begin{aligned}\sum_{u\in\mathcal{U}}\mathcal{L}_{\theta,\phi}(r^u_{1:N_u}) &\approx \sum_{u\in\mathcal{U}}\widetilde{L}(\theta,\phi,R^u,\boldsymbol{\epsilon}^u,u),\\ &\approx \sum_{u\in\mathcal{U}}\frac{N^u}{M}\widetilde{L}(\theta,\phi,R^u_M,\boldsymbol{\epsilon}^u_M,u),\\ &= \sum_{u\in\mathcal{U}}\widetilde{L}^M(\theta,\phi,R^u_M,\boldsymbol{\epsilon}^u_M,u),\end{aligned} \tag{19}$$

where ϵ^u_M is the set of all random samples drawn from (18) corresponding to sub-trajectory R^u_M. Then, our proposed algorithm iteratively updates parameter θ and ϕ based on the gradient of $\widetilde{L}^M$.

Finally, we present the detailed process of training our proposed model in Algorithm 1. As we can observe, in each iteration, this algorithm first randomly samples one sub-trajectory R^u_M with M continuous mobility records from R^u for each user $u\in\mathcal{U}$. Then, the corresponding $\boldsymbol{\epsilon}^u_M$ is sampled from the noise distribution $\mathcal{N}(0,1)$. Based on them, $\widetilde{L}^M$ is calculated as the loss function, and this algorithm updates θ and ϕ based on the gradient of $\widetilde{L}^M$.

We also present the pseudocode of synthesizing trajectories based on our proposed model in Algorithm 2. Specifically, the first visiting time is sampled from $p^u(t)$, which is the marginal distribution of the first visiting time in each trajectory, and a threshold of the time span of synthesized trajectories is defined as T. Then, based on the neural network weight θ and ϕ learned from Algorithm 1, this algorithm synthesizes users' mobility records one by one based on (10)–(15) until the time span of synthesized trajectories exceeds T.

TABLE II
STATISTIC INFORMATION OF DATASETS

Datasets	**ISP**	**GeoLife**	**FSQ-NYC**	**FSQ-TKY**
Duration	7 days	64 months	11 months	11 months
#Users	100,000	182	1,083	2,293
#Records/User	261	453	210	250
#Locations	9,000	5,669	38,333	61,858

IV. PERFORMANCE EVALUATION

A. Experimental Settings

Dataset: We conduct our experiments on three real-world trajectory datasets, including the ISP dataset, Geolife dataset, and Foursquare dataset. Their information is introduced in detail as follows:

- *ISP Dataset:* The ISP dataset is collected by collaborating with a major Internet service provider (ISP) in China, of which the duration is from April 19 to April 26, 2016, covering the whole metropolitan area of Shanghai. Each record in this dataset is characterized by an anonymized userID, timestamp, the cellular base station. There are over 2,140,327 user trajectories involved in this dataset.
- *Geolife Dataset:* This dataset is collected by MSRA Geolife project [22], which contains mobility trajectories of 178 users with the duration of over four years (from April 2007 to October 2011). There are 17,621 trajectories involved in this dataset. Each mobility record in this dataset is characterized by latitude, longitude, altitude, and timestamp.
- *Foursquare Datasets:* This dataset is a publicly available dataset collected by Yang et al. [23], which contains check-ins of two cities including New York City and Tokyo from April 12, 2012 to February 16, 2013. There are 227,428 check-ins in New York City and 573,703 check-ins in Tokyo. Each record in the dataset includes the corresponding user ID, timestamp, and PoI.

For better comparing and understanding different datasets, we show their statistic information in Table II. Specifically, we can observe that the ISP dataset has the densest trajectories, whose average number of records per user is similar to other datasets while the duration is much shorter. On the other hand, the GeoLife dataset has the sparsest trajectories.

The raw human trajectory data may be full of noise, especially communication data with the base stations (e.g., ISP dataset) or the GPS trajectories (e.g., GeoLife dataset). Specifically, there may exist inaccurate locations in trajectories, and it is also possible to have a large number of redundant records in a short period of time in the trajectories. What's more, the time interval between adjacent records can be very large, e.g., from several hours to several days [23]. In order to eliminate the noise, we pre-process the trajectories by merging mobility records with the same location and close time, and filtering out outlier locations. For the check-in datasets, the noise in terms of redundant records and inaccurate locations is less. However, the interval between adjacent records is prolonged. Thus, all trajectory datasets are pre-processed by breaking sparse trajectories into dense sub-trajectories if the time interval between adjacent records exceeds a predefined threshold.

Compared Algorithms: We compare our proposed trajectory synthesizing algorithm with the following algorithms.

- *TimeGEO:* As a model-based trajectory synthesizing method, TimeGEO [3] models users' state only based on whether they are at home. Then, users' further movements are modeled based on the explore and preferential return (EPR) model [24].
- *Semi-Markov:* In semi-Markov process, the waiting time is modeled by the exponential distribution [25]. Dirichlet prior and gamma prior are used to model the transition matrix and the intensity of the waiting time to implement a Bayesian inference.
- *Hawkes:* Hawkes process [26] is a widely used classical temporal point process, where an occurred data point will influence the intensity function of future points.
- *LSTM:* This model directly predicts the next locations and waiting time based on the LSTM network, and the prediction results are utilized as the synthesized trajectories [27].
- *MoveSim:* This model is recently proposed to synthesize human trajectories based on GAN, which introduces prior knowledge and physical regularities to the SeqGAN model [5].

The implementation code of TimeGEO, Semi-Markov, Hawkes, and LSTM can be found in our Github repository. As for MoveSim, we use the official implementation provided in https://github.com/FIBLAB/MoveSim. All compared algorithms are trained on the three real-world datasets, and then utilized to synthesize trajectories. Specifically, TimeGEO, Semi-Markov, and Hawkes are representative model-based methods for trajectory synthesizing, while LSTM and MoveSim are the representative data-based methods for trajectory synthesizing. In addition, among the data-based models, MoveSim has been the state-of-the-art algorithm with one of the best performance. Thus, by comparing these selected algorithms covering the most representative model-based and data-based methods, the performance of our proposed algorithm can be credibly evaluated.

Metrics: As the problem of trajectory synthesizing is defined in the previous section, the synthesized trajectories should be evaluated in terms of two aspects. In the first aspect, the synthesized trajectories should preserve the usability of the original trajectory dataset. That is, two datasets should have similar statistical properties. Thus, we elaborately select six metrics to evaluate both the spatial statistical property and the temporal statistical property of synthesized trajectories compared with real-world ground-truth trajectories of users, which are defined as follows:

- *Distance:* The traveling distance between adjacent mobility records in users' trajectories, and is a metric in terms of the spatial perspective.
- *Radius:* Radius of gyration, which is defined as the root mean square of the distance of each location point in the one trajectory to its center of mass [28].
- *G-rank:* The top visited frequency to different locations with respect to all users, which is a metric in terms of the spatial perspective.
- *Duration:* The waiting time of users at different locations, which is a metric in terms of the temporal perspective.

- *Move:* The visiting time of each mobility record, which is a metric in terms of the temporal perspective.
- *Stay:* The average waiting time as a function of the visiting time, which is a metric in terms of the temporal perspective.

Specifically, the metrics of Distance, Location, G-rank, Duration, and Move are expressed by probability distributions defined on one-dimensional spaces. In order to compare the synthesized trajectories and real trajectories in a more intuitive way, we utilize the Jensen-Shannon divergence (JSD) to measure their difference. Specifically, for two distribution $\boldsymbol{p}$ and $\boldsymbol{q}$, the JSD between them can defined as:

$$\mathrm{JSD}(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{2}\mathrm{KL}\left(\boldsymbol{p}||\frac{\boldsymbol{p}+\boldsymbol{q}}{2}\right) + \frac{1}{2}\mathrm{KL}\left(\boldsymbol{q}||\frac{\boldsymbol{p}+\boldsymbol{q}}{2}\right), \quad (20)$$

where $\mathrm{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence [29]. On the other hand, the distribution of Stay can be characterized from two dimensions, including the visiting time and the waiting time. Therefore, unlike other metrics, we compute the Jensen-Shannon divergence between the Stay distributions of real data and generated data in this two-dimensional space.

In terms of the second aspect, the synthesized trajectories should be different from the original trajectories. We use two metrics including overlapping ratio and symmetrized segment-path distance (SSPD) [30] to evaluate the performance in terms of this perspective. Specifically, we define the overlapping ratio as the ratio of the number of identical spatio-temproal points between the synthesized trajectories and the original trajectories, where the identical spatio-temproal points are defined based on discrete time bins (1 min in our experiments). On the other hand, SSPD provides a meaningful evaluation method of distance between high-dimensional trajectories.

Then, multiple original trajectories belonging to the same users are ranked based on the overlapping ratio and SSPD, and the trajectory with the highest overlapping ratio or the smallest SSPD is selected. The results can also be extended by considering more similar trajectories, including the top-2 and top-3 similar real trajectories.

Parameter Settings: As the model shown in Fig. 1, the embedding sizes of location and user ID are 256 and 128 respectively, and the encoding sizes of the time stamp and waiting time are both 256. And the output size of LSTM_ϕ is set to 512. In addition, MLP_ϕ contains two identical 2-layer MLPs, with Leaky-ReLu activation after the first layers and the size of the hidden layers being set to 256. The latent variable z is set to be a 512-sized vector. In terms of the generation, the embedding layer of user ID as well as the positional encoding are the same as those in the encoder, and the output size of LSTM_θ is also 512. MLP_θ is composed of two components for temporal parameters and spatial parameters, respectively. The temporal parameter η is obtained from a 3-layer MLP with Leaky-ReLu activation function after the first two layers, and an exponential non-linearity applied after the third layer to ensure the obtained η is positive. The spatial parameter Ψ^{loc} and Ψ^{poi} are calculated based on the output of a 3-layer MLP with Leaky-ReLu activation after the first two layers. Ψ^{loc} is the soft-max of the output, while Ψ^{poi}, or more precisely ξ^{poi}, is obtained by activating the output with Leaky-ReLu and then passing through another linear layer to ensure the correct vector size, the size of POI, for computation. The size of the hidden layers in MLP_θ is set to 128. The weights α, β are set to 0.9 and 0.1, respectively. In addition, the LSTMs are initialized using Xavier's method and the linear layers are initialized using KaiMing's Method. The threshold of the time span T is set to 24 hours, i.e., one day by default. As for the spatial divisions, we utilize the default spatial divisions, i.e., cellular base station for the ISP dataset, and we utilize the spatial grids of 1 km $\times$ 1 km as the spatial divisions for the GeoLife dataset and two Foursquare datasets. Unless otherwise stated, 90% of trajectories in each dataset are randomly selected as the training set, and the left 10% of trajectories are used for testing.

In addition, all baselines except MoveSim and TimeGEO are able to generate trajectories with a continuous temporal distribution, which can be compared with trajectories synthesized by our model in the same setting directly. However, MoveSim and TimeGEO are proposed to synthesize trajectories with discrete time bins. What's more, MoveSim further requires fixed-length trajectories with discrete temporal distribution as input. In the original paper of MoveSim [5], all baselines use fixed-length discrete trajectories as input and generate trajectories with the same form to be compared with the discretized and interpolated real-world trajectories. Differently, in our experiments, the trajectories with a continuous temporal distribution in the training set are discretized and interpolated to obtain the fixed-length trajectories to train the MoveSim model. Then, in the process of testing, the synthesized trajectories of MoveSim, which are discrete and fixed-length, are post-processed by merging records with identical locations and adjacent time bins to produce time-continuous trajectories with the same form as introduced in Section II-A, and then compared with the original variable-length trajectories with continuous temporal distribution in the test set. We implement similar operations to the trajectories generated by TimeGEO.

B. Experimental Results

Performance in Terms of Usability: We first show the performance of different algorithms in terms of the selected metrics measured by JSD. We first show the experimental results based on the ISP dataset in Table III, where we repeat our experiments for 5 times and exhibit the means and standard deviations of different metrics. We can observe that all metrics of trajectories synthesized by different algorithms are stable. Specifically, the average ratio of the standard deviation to the mean of different metrics of different algorithms is only 25.1%, while the performance gap between our proposed algorithm and each compared algorithm is 56.5%. In addition, we can observe that the performance of MoveSim obviously decreases compared with the results shown in [5]. The main reason is that MoveSim is designed for synthesizing trajectories with discrete and successive time slots. Thus, an interpolation method is implemented on different trajectory datasets to fill up users' locations at missing time slots. At the same time, MoveSim requires fixed-length trajectories as the input. Thus, the trajectories in the test set are also discretized and interpolated in [5] to obtain the fixed-length trajectories. Different from them, we mainly consider

TABLE III
PERFORMANCE COMPARISONS ON THE ISP DATASET, WHERE BOLD DENOTES BEST (LOWEST) RESULTS AND UNDERLINE DENOTES THE SECOND BEST RESULTS

Metric / Algorithm	**Distance** (JSD)	**Radius** (JSD)	**G-Rank** (JSD)	**Duration** (JSD)	**Move** (JSD)	**Stay** (JSD)
Semi-Markov	0.1217±0.0044	0.0126±0.0003	0.0987±0.0002	0.0142±0.0002	0.0686±0.0004	0.1010±0.0004
TimeGEO	0.4321±0.0025	0.2752±0.0035	0.1370±0.0022	0.1987±0.0020	0.2193±0.0007	0.4067±0.0014
Hawkes	0.0262±0.0010	0.0104±0.0008	0.1438±0.0014	0.0444±0.0008	0.1053±0.0050	0.1315±0.0010
LSTM	0.0476±0.0013	0.0131±0.0005	**0.0962±0.0004**	0.0740±0.0004	0.0592±0.0003	0.1481±0.0002
MoveSim	0.0278±0.0120	0.0551±0.0393	0.2612±0.0315	0.3077±0.0060	0.1243±0.0217	0.3911±0.0178
Our	**0.0193±0.0007**	**0.0053±0.0008**	0.1410±0.0002	**0.01294±0.0006**	**0.0133±0.0005**	**0.0542±0.0005**

TABLE IV
PERFORMANCE COMPARISONS ON THE GEOLIFE DATASET, WHERE BOLD DENOTES BEST (LOWEST) RESULTS AND UNDERLINE DENOTES THE SECOND BEST RESULTS

Metric / Algorithm	**Distance** (JSD)	**Radius** (JSD)	**G-Rank** (JSD)	**Duration** (JSD)	**Move** (JSD)	**Stay** (JSD)
Semi-Markov	0.2926±0.0099	0.1838±0.0037	0.2509±0.0012	0.1789±0.0012	0.0462±0.0012	0.2300±0.0008
TimeGEO	0.2660±0.0161	0.2920±0.0173	0.2600±0.0024	0.1940±0.0082	0.1625±0.0051	0.4493±0.0074
Hawkes	**0.1157±0.0134**	0.1353±0.0051	0.2539±0.0025	0.0581±0.0016	0.0614±0.0108	0.1922±0.0055
LSTM	0.1696±0.0045	0.1045±0.0024	0.1997±0.0028	0.0923±0.0021	0.0410±0.0017	0.1988±0.0032
MoveSim	0.1523±0.0373	0.1068±0.0440	0.2981±0.0435	0.2750±0.0028	0.1548±0.0802	0.4238±0.0519
Our	0.1503±0.0095	**0.0911±0.0079**	**0.1996±0.0024**	**0.0415±0.0018**	**0.0233±0.0028**	**0.1301±0.0023**

TABLE V
PERFORMANCE COMPARISONS ON THE FOURSQUARE DATASET OF NYC, WHERE BOLD DENOTES BEST (LOWEST) RESULTS AND UNDERLINE DENOTES THE SECOND BEST RESULTS

Metric / Algorithm	**Distance** (JSD)	**Radius** (JSD)	**G-Rank** (JSD)	**Duration** (JSD)	**Move** (JSD)	**Stay** (JSD)
Semi-Markov	0.0808±0.0093	0.0511±0.0049	0.1398±0.0004	0.0400±0.0009	0.0319±0.0015	0.1383±0.0013
TimeGEO	0.2726±0.0120	0.3540±0.0074	0.1693±0.0043	0.0985±0.0035	0.1277±0.0036	0.3307±0.0033
Hawkes	0.0615±0.0072	0.0601±0.0061	0.1635±0.0010	0.0238±0.0019	0.0304±0.0067	0.1719±0.0025
LSTM	0.0419±0.0020	0.0420±0.0023	0.1244±0.0013	0.0439±0.0018	0.0236±0.0008	0.1590±0.0041
MoveSim	0.0421±0.0032	0.0676±0.0035	0.3331±0.0683	0.1376±0.0024	0.1269±0.0172	0.3284±0.0148
Our	**0.0320±0.0023**	**0.0313±0.0048**	**0.1234±0.0022**	**0.0202±0.0006**	**0.0133±0.0007**	**0.1318±0.0026**

modeling variable-length trajectories in our paper. In the process of training, the trajectories in the training set are still discretized and interpolated to obtain the fixed-length trajectories to train the MoveSim model. However, in the process of testing, the synthesized trajectories of MoveSim, which are discrete and fixed-length, are compared with the original variable-length trajectories with continuous temporal distribution, leading to its performance degradation. We can observe that our model achieves the best performance in most usability metrics.

Specifically, the performance gap between our proposed algorithm and the best compared algorithm is the largest in terms of Move, indicating the strong ability of our proposed model in terms of modeling the visiting time of users' mobility behavior. In addition, we can observe that LSTM achieves the second best performance in terms of Move, since it models the sequential dependencies of human movement in a similar way [27]. Compared with it, our model is able to automatically learn these characteristics of human mobility from data.

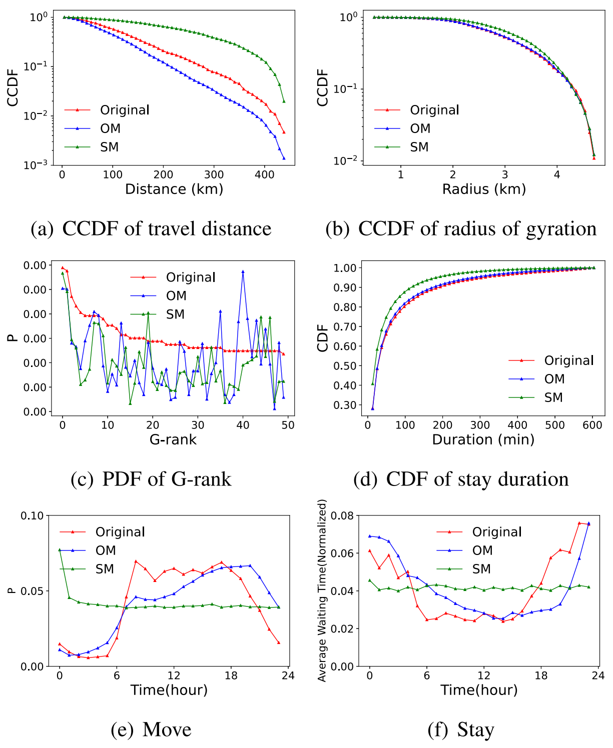
Further, we show the performance of different algorithms on the Geolife dataset and Foursquare datasets of New York city and Tokyo in Tables IV, V, and VI, respectively. From the experimental results, we can observe a similar trend with the results shown in Table III. That is, our model achieves the best performance in terms of the most usability metrics, demonstrating the superiority of our method in synthesizing human trajectories.

Usability Visualization: For better comparison, we also visualize the metrics of usability. Specifically, the probability density function (PDF), the cumulative distribution function (CDF), and the time curve are selected for different metrics and shown in Figs. 2 and 3. As we can observe from Fig. 2(e) and (f), we can observe our proposed model well captures the daily rhythm of humans, i.e., the trajectories based on our proposed model have higher travel frequency at daytime and longer average waiting time at night. Overall, compared with the performance of the best baselines, trajectories synthesized by our proposed algorithm are more similar to the real trajectories in terms of most metrics, indicating its effectiveness.

Performance in Terms of Uniqueness: We evaluate whether the synthesized trajectories are different from the original trajectories, i.e., the uniqueness of the synthesized trajectories. We select two key metrics to measure their difference.

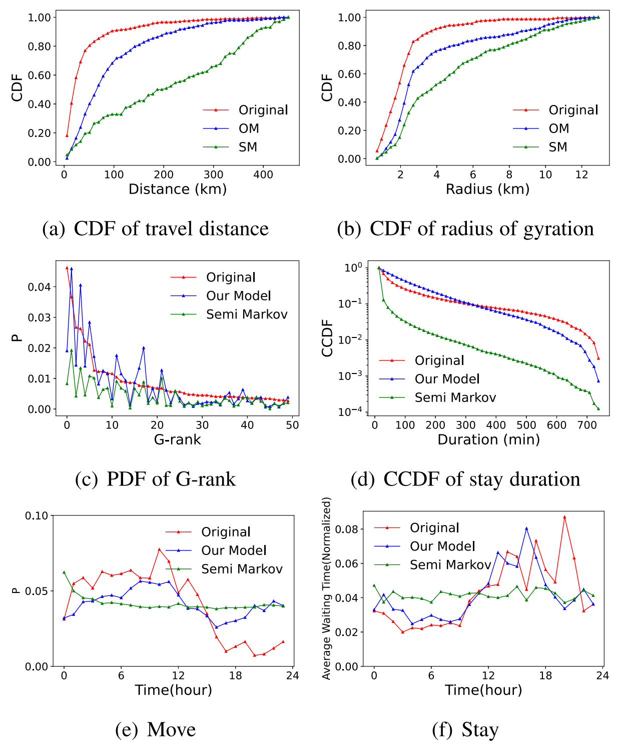
TABLE VI
PERFORMANCE COMPARISONS ON THE FOURSQUARE DATASET OF TOKYO, WHERE BOLD DENOTES BEST (LOWEST) RESULTS AND UNDERLINE DENOTES THE SECOND BEST RESULTS

Metric / Algorithm	**Distance** (JSD)	**Radius** (JSD)	**G-Rank** (JSD)	**Duration** (JSD)	**Move** (JSD)	**Stay** (JSD)
Semi-Markov	0.1319±0.0139	0.0464±0.0012	0.0425±0.0003	0.0664±0.0003	0.1057±0.0011	0.1915±0.0008
TimeGEO	0.2361±0.0072	0.3238±0.0061	0.0525±0.0013	0.1378±0.0026	0.2606±0.0020	0.4936±0.0042
Hawkes	**0.0279±0.0040**	<u>0.0293±0.0015</u>	0.0542±0.0008	<u>0.0491±0.0013</u>	<u>0.0941±0.0039</u>	<u>0.1845±0.0018</u>
LSTM	0.0644±0.0039	0.0316±0.0021	<u>0.0333±0.0004</u>	0.0909±0.0013	0.1109±0.0009	0.2439±0.0022
MoveSim	0.0634±0.0270	0.0397±0.0118	0.1987±0.0310	0.1121±0.0271	0.2819±0.0404	0.4827±0.0524
Our	<u>0.0583±0.0021</u>	**0.0287±0.0006**	**0.0329±0.0007**	**0.0489±0.0008**	**0.0140±0.0005**	**0.1048±0.0019**



(a) CCDF of travel distance (b) CCDF of radius of gyration (c) PDF of G-rank (d) CDF of stay duration (e) Move (f) Stay

Fig. 2. Performance visualization on ISP Dataset.



(a) CDF of travel distance (b) CDF of radius of gyration (c) PDF of G-rank (d) CCDF of stay duration (e) Move (f) Stay

Fig. 3. Performance visualization on the Geolife Dataset.

The first metric is the overlapping ratio. Specifically, for two trajectories for comparison, we align them in the time dimension one by one based on a pre-defined temporal threshold and determine whether the locations at the corresponding time points are exactly the same. Then, the overlapping ratio is defined as the ratio of the number of identical locations to the total trajectory length. Based on the concept of the overlapping ratio, the metric of uniqueness examines the overlapping ratio of top-k most similar original trajectories for each synthesized trajectory. For a given synthesized trajectory, if the overlapping ratio between it and its top-k most similar original trajectory is high, it represents that the synthesized trajectory is possibly contained in the original trajectory dataset, indicating a poor performance of the model to generate trajectories different from the original trajectory dataset. In turn, a small overlapping ratio indicates the good performance of the model to generate trajectories different from the original trajectory dataset. The CDF of the overlapping ratio of top-1, top-2, and top-3 most similar original trajectories for each synthesized trajectory of the two datasets is shown in Fig. 5(a) and (b). Take the results of the Geolife dataset for example. We can observe that in terms of the top-1 most similar trajectory, over 80% synthesized trajectories have an overlapping ratio of less than 40%. In terms of the top-3 most similar trajectories, we can further observe that over 80% synthesized trajectories have an overlapping ratio of less than 20%. The small overlapping ratio between synthesized trajectories and original trajectories indicates that the synthesized trajectories are significantly different from the original trajectories, indicating the effectiveness of the synthesized trajectories.

The second metric is the Symmetrized Segment-Path Distance (SSPD) [30]. Specifically, the Segment-Path distance $\mathrm{SPD}(T_1, T_2)$ between two arbitrary trajectories T_1 and T_2 is defined as the mean of all distances from records composing T_1 to the trajectory T_2, which is asymmetric in terms of T_1 and T_2. Then, SSPD is defined as the mean of $\mathrm{SPD}(T_1, T_2)$ and $\mathrm{SPD}(T_2, T_1)$, which is symmetric in terms of T_1 and T_2.

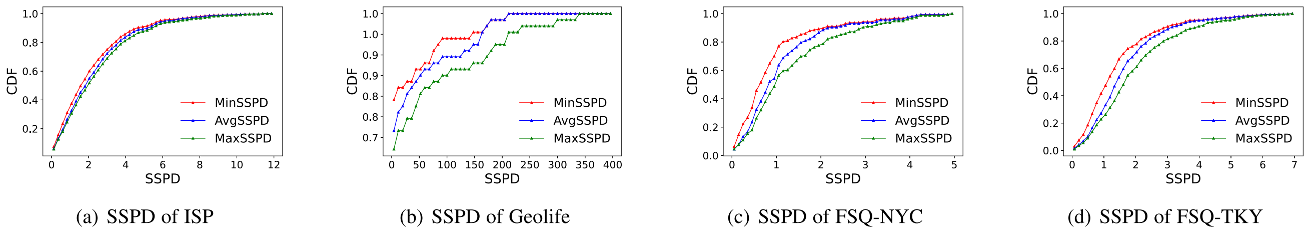


Fig. 4. Evaluation in terms of SSPD between synthesized trajectories and original trajectories.

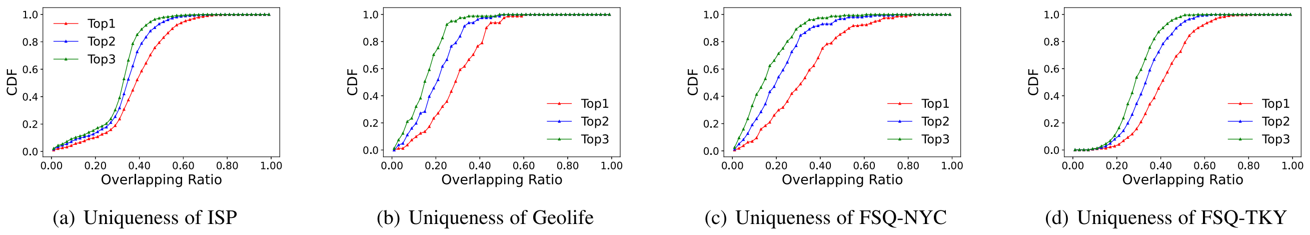


Fig. 5. Evaluation in terms of Uniqueness between synthesized trajectories and original trajectories.

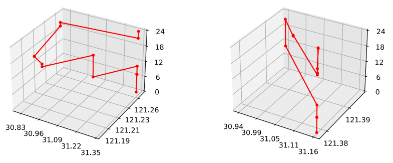


Fig. 6. Visualization of individual synthesized trajectories.

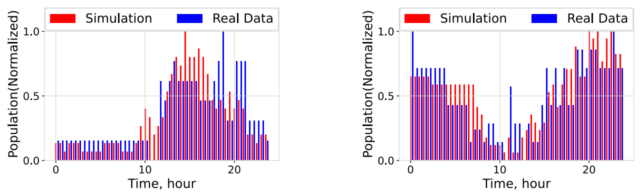


Fig. 7. Visualization of aggregate synthesized trajectories.

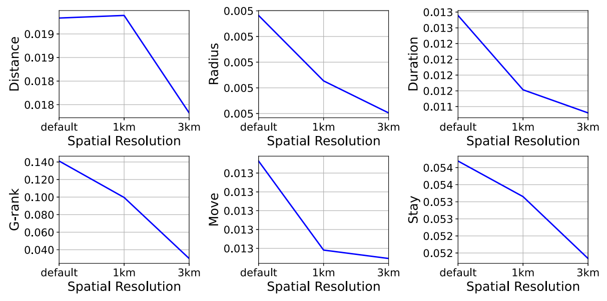


Fig. 8. Performance of our proposed algorithm versus spatial resolution.

Then, the SSPD between the synthesized trajectories and the original trajectories is calculated to evaluate the ability of our proposed model in terms of synthesizing trajectories different from the original trajectory dataset. The experimental results of the four datasets are shown in Fig. 4, where we show the distribution of the minimum SSPD, maximum SSPD, and average SSPD, respectively. We can observe that the SSPD between the synthesized trajectories and the original trajectories is also large. Specifically, the average SSPD is 2.53 km, 25.81 km, 1.13 km, and 1.78 km on the ISP, GeoLife, FSQ-NYC, and FSQ-TKY datasets, respectively. Such a large distance in terms of SSPD suggests that users' true locations are not leaked from the synthesized trajectories, indicating the good performance of our proposed model.

Case Study: We further present two case studies of the synthesized trajectories to evaluate their quality. We first show the visualization of two individual synthesized trajectories in Fig. 6. We can observe the strong circadian rhythm of the synthesized trajectories. Then, we show the visualization of the aggregate synthesized trajectories in Fig. 7, where the aggregate population of two representative regions is exhibited. We can observe that our synthesized trajectories resemble real data by well capturing its trend, indicating the effectiveness of our proposed model.

Performance With Different Data Quality: We then evaluate the performance of our proposed method with different data quality. Specifically, we consider two key factors including the spatial resolutions and the training set size. Without loss of generality, we only show the experimental results based on the ISP dataset in this section.

We show the performance of our proposed model in terms of different usability metrics as the function of spatial resolutions. Specifically, we consider three different spatial resolutions, which include the default geographical division based on the cellular base stations and the geographical division based on the spatial grids of 1 km×1 km and 3 km×3 km, respectively. The results are shown in Fig. 8. As we can observe, with lower spatial resolutions, the usability metrics in terms of JSD, however, decrease. The main reason is that the synthesized trajectories are also evaluated based on low-resolution test datasets, which

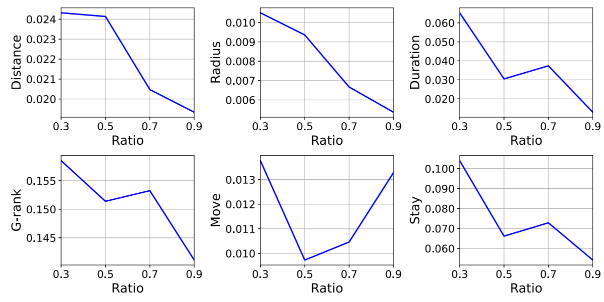


Fig. 9. Performance of our proposed algorithm versus training set size on the ISP dataset, where the training set size is represented by the ratio of records used as the training set.

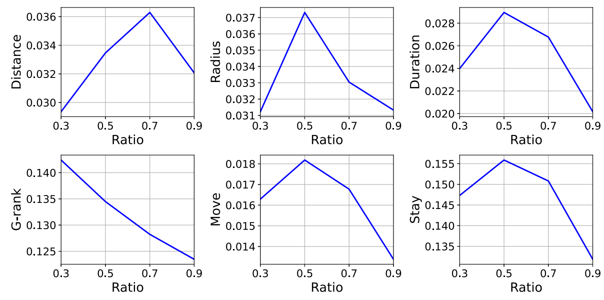


Fig. 10. Performance of our proposed algorithm versus training set size on the FSQ-NYC dataset, where the training set size is represented by the ratio of records used as the training set.

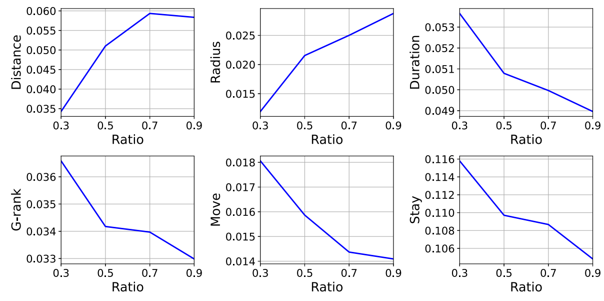


Fig. 11. Performance of our proposed algorithm versus training set size on the FSQ-TKY dataset, where the training set size is represented by the ratio of records used as the training set.

is an easier task. Overall, the performance in terms of all usability metrics is stable with changes in the spatial resolutions, indicating the robustness of our proposed algorithms.

Then, we show the performance of our proposed model in terms of different training set size. Specifically, we utilize 30%, 50%, 70%, and 90% trajectories as the training set, respectively, and show the performance of our proposed model in terms of different usability metrics as the function of the ratio of mobility trajectories used as the training set. The experimental results on the ISP dataset are shown in Fig. 9. In addition, the corresponding results on the FSQ-NYC and FSQ-TKY datasets are shown in Figs. 10 and 11, respectively. We can observe that in most cases, the performance in terms of different metrics increases with the size of the training set. However, we can also observe a small number of metrics with contrary tendencies. That is, the performance decreases with the size of the training set, especially in the FSQ-NYC and FSQ-TKY datasets. A potential reason is that these two Foursquare datasets are sparse. That is, the time interval between two adjacent records is very long. Thus, in our pre-processing steps in terms of breaking trajectories into sub-trajectories if the time interval between adjacent records is prolonged, the Foursquare trajectories are broken into too many short trajectories, leading to the difficulty in terms of modeling their distribution. Thus, the performance in terms of some metrics is not stable on the Foursquare datasets.

Ablation Study: In order to evaluate the importance of the different components proposed in our work, we conduct an ablation study on two mobility datasets, which is performed by removing each component from the complete model. The results are also evaluated by the selected six metrics. Without loss of generality, we only show the results on the ISP dataset in Table VII. Specifically, Our Best represents the complete version of our proposed model, - Fourier replaces the Fourier-based positional encoder by the traditional positional encoder, - Ψ^{poi} removes the global PoI distribution matrix, - I_i removes the PoI distribution I_i in the input of the inference module, - τ_i removes the waiting time τ_i in the input of the inference module, and - Feedback removes the feedback of the generation module. In addition, we define $\Delta = (\xi' - \xi)/\xi$ as the relative improvement of our proposed model compared with these variants, where ξ is the performance indicator of Our Best and ξ' is the performance indicator of a certain variant. In addition, we define $\bar{\Delta}$ as the average relative improvement in terms of all the six metrics. As we can observe, the absence of τ_i and the alternation of the positional encoding would both cause a significant drop of the performance in terms of Duration. Another observation is that - Feedback has the largest performance gap in terms of most metrics, indicating the importance of the feedback loop in the generation module. That is, in the generation module, except for $z_{1:i}$, we further need $r_{1:i-1}$ to accurately model the distribution of r_i, which is consistent with our discussion in Section III-B. Overall, our proposed method outperforms all variants in terms of the six metrics in most cases, indicating the effectiveness of our proposed modules including the Fourier-based positional encoding mechanism, feedback mechanism in the general module, etc.

V. Related Work

Applications Based on Mobility Trajectories: Mobility trajectories are instrumental for a number of applications including location recommendation [31], [32], [33], [34], [35], [36], intelligent transportation [2], [37], [38], etc. Ye et al. [31] consider the social and geographical characteristics of users and locations in their location recommendation system. Zheng et al. [32] recommend locations to users based on their trajectories and comments. Yuan et al. [33] recommend the point of interest (PoI) to users by using the collaborative filtering method and incorporating temporal information. Liu et al. [34] propose a geographical probabilistic factor analysis framework for PoI recommendations, which strategically takes multiple factors influencing the user check-in decision process into consideration. Wang et al. [35] recommend spatial items by modeling and

TABLE VII
RESULTS OF THE ABLATION STUDY IN TERMS OF DIFFERENT METRICS

Metrics	**Distance**		**Radius**		**G-Rank**		**Duration**		**Move**		**Stay**		$\bar{\Delta}$
	JSD	Δ	JSD	Δ	JSD	Δ	JSD	Δ	JSD	Δ	JSD	Δ	
Our Best	<u>0.0193</u>	-	**0.0053**	-	0.1410	-	**0.0129**	-	**0.0132**	-	**0.0541**	-	-
- Fourier	0.0246	27.5%	0.0061	15.1%	<u>0.1132</u>	-19.7%	0.0547	324.0%	0.0143	8.3%	0.0951	75.8%	71.8%
- Ψ^{poi}	**0.0156**	-19.1%	<u>0.0056</u>	5.6%	0.1705	20.9%	<u>0.0186</u>	44.2%	0.0206	56.1%	0.0706	30.5%	23.0%
- I_i	0.0234	21.2%	0.0078	47.1%	0.1502	6.5%	0.0202	56.6%	0.0153	15.9%	<u>0.0638</u>	17.9%	27.5%
- τ_i	0.0271	40.4%	0.0063	18.9%	0.1265	-10.1%	0.0387	200.0%	<u>0.0136</u>	3.0%	0.0777	43.6%	49.3%
- Feedback	0.0297	53.9%	0.0108	103.7%	**0.0865**	-38.6%	0.0529	310.1%	0.0809	512.9%	0.1572	190.6%	188.8%

Bold denotes the best (lowest) results and underline denotes the second best results.

fusing the sequential influence, cyclic patterns, and personal interests. Yuan et al. [36] propose a context-aware location recommendation system that considers user, spatial, temporal, and activity aspects simultaneously. Liu et al. [2] propose a localized transportation mode choice model for bus routing optimization based on location trajectories of taxicabs. Biczok et al. [37] find strong logical ties between different locations based on human trajectories obtained from a live indoor/outdoor positioning and navigation system deployed at a large university campus in Norway. Wu et al. [38] interpret traffic dynamics based on user trajectories obtained from geo-tagged contents published in social networks. The large number of applications based on mobility trajectories further indicates our strong motivations to develop a power trajectory synthesizing model to overcome the privacy issues of utilizing real-world trajectory data.

Temporal Point Process: The temporal point process has been widely used in many applications, including earthquake modelling [39], computational finance [26], and human behavior modelling [40]. In recent years, the temporal point process has been combined with the novel deep neural network techniques [14], [15], [17], [18], [19], and has exhibited outstanding performance in terms of the prediction of the time-stamped sequential data. Du et al. [18] propose the Recurrent Marked Temporal Point Process (RMTPP), which utilizes the Recurrent Neural Network (RNN) to model the intensity function of the temporal point process. Wu et al. [19] utilize the RMTPP to model time-stamped sequences with multiple markers based on the factorial model, of which the idea is inspired by the factorial hidden Markov models [41]. However, these models are deterministic, leading to their disability in capturing the uncertainty of sequential data. At the same time, other studies seek to combine the neural point process with variational inference techniques to capture the uncertainty [12], [13], [16], which also improves their ability to synthesize sequential data. Mehrasa et al. [12] propose the Action Point Process VAE (APP-VAE), which utilizes VAE combined with RNN for modeling action sequences from videos. Pan et al. [16] combine VAE with the temporal point process to model sequential data. Mehrasa et al. [13] further propose an intensity-free framework for variational temporal point processes that directly models the point process distribution by utilizing normalizing flows. Simini et al. [42] propose a deep gravity model to generate mobility flows that utilize deep learning to exploit features (e.g., land use, road network, transport, food, and health facilities) and discover non-linear relationships between them. At the same time, a number of existing studies combine deep learning techniques with ordinary differential equations (ODE) [43], [44], which is referred to as ODENet. Chen et al. [44] utilize ODENet for continuous-depth modeling and continuous-time modeling. Jia et al. [43] further combine ODENet with temporal point processes, which focuses on predicting future time series rather than data generation. Thus, rather than the variational inference framework, this method selects to directly use the log probability density as the target function. Different from them, our paper focuses on the human trajectory synthesizing problem, which has a number of unique characteristics and challenges compared with the common sequential data synthesizing problem, including hidden semantics and complicated joint correlation with both visiting time and waiting time.

Trajectory Synthesizing: A number of different approaches have been proposed to synthesize human trajectories [45], [46], [47], which can be divided into model-based methods and data-based methods. Model-based methods make strong assumptions about human mobility and synthesize trajectories based on probabilistic models or statistical models [3], [6], [24], [48], [49], [50], [51]. Isaacman et al. [48] synthesize trajectories based on a statistical probability model in terms of the distribution of commute distances, probability of a call at each time bin, etc. Lin et al. [6] generate mobility trajectories based on Input-Output Hidden Markov Model (IO-HMM). Bindschaedler et al. [49] focus on preserving the privacy of users, and they synthesize trajectories by replacing each location with locations with similar semantics. On the other hand, data-based methods seek to model users' mobility behavior by adopting deep learning techniques including CNN or RNN combined with GAN [4], [5], [52], [53]. Ouyang et al. [4] synthesize fixed-length mobility trajectories with continuous-time gap based on the CNN model. Feng et al. [5] synthesize fixed-length trajectories based on GAN. However, these methods all have a number of different limitations. Due to the strong assumption of model-based methods [3], [48], they are hard to incorporate multi-dimensional external information. Data-based deep neural methods [4], [5] can only model and synthesize fixed-length trajectories limited by their loss function in the training process. Different from them, our proposed method utilizes the temporal point process as the bridge between deep neural networks and users' mobility behavior based on the novel variational inference framework, and is able to capture the hidden semantics and uncertainty of human trajectories, and model the continuous temporal distribution of variable-length trajectories, which overcomes the limitation of existing approaches.

VI. Conclusion

In this paper, we propose a novel human trajectory synthesizing model, which utilizes the temporal point process as the bridge between deep neural networks and users' mobility behavior based on the variational inference framework. This model maintains the strong modeling ability of the neural network to incorporate multi-dimensional external information. In addition, the Bayesian-based variational inference framework much improves the interpretability of the proposed model and the ability to model variable-length trajectory with continuous temporal distribution. We believe this work paves the way toward a systematic understanding, synthesizing, and simulating human mobility trajectories in the real world. There are several limitations of this work. First, we do not consider complicated privacy criteria such as differential privacy, because they require modification of the loss function, which will influence the performance of the proposed algorithm. In addition, other formats of the intensity function in the adopted exponential function possibly improve the performance of the proposed algorithm, which we leave for future work.

References

[1] A. Hess, K. A. Hummel, W. N. Gansterer, and G. Haring, "Data-driven human mobility modeling: A survey and engineering guidance for mobile networking," *ACM Comput. Surveys*, vol. 48, no. 3, pp. 1–39, 2015.

[2] Y. Liu et al., "Exploiting heterogeneous human mobility patterns for intelligent bus routing," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 360–369.

[3] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. González, "The timegeo modeling framework for urban mobility without travel surveys," *PNAS*, vol. 113, no. 37, pp. E5370–E5378, 2016.

[4] K. Ouyang, R. Shokri, D. S. Rosenblum, and W. Yang, "A non-parametric generative model for human trajectories," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3812–3817.

[5] J. Feng, Z. Yang, F. Xu, H. Yu, M. Wang, and Y. Li, "Learning to simulate human mobility," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 3426–3433.

[6] Z. Lin, M. Yin, S. Feygin, M. Sheehan, J.-F. Paiement, and A. Pozdnoukhov, "Deep generative models of urban mobility," *IEEE Trans. Intell. Transp. Syst.*, 2017.

[7] X. Song, H. Kanasugi, and R. Shibasaki, "Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2618–2624.

[8] J. Feng et al., "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proc. World Wide Web Conf.*, 2018, pp. 1459–1468.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*.

[10] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Berlin, Germany: Springer, 2007.

[11] M.-A. Rizoiu, Y. Lee, S. Mishra, and L. Xie, "A tutorial on hawkes processes for events in social media," 2017, *arXiv: 1708.06401*.

[12] N. Mehrasa, A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, "A variational auto-encoder model for stochastic point processes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3160–3169.

[13] N. Mehrasa et al., "Point process flows," 2019, *arXiv: 1910.08281*.

[14] Q. Cao, H. Shen, K. Cen, W. Ouyang, and X. Cheng, "DeepHawkes: Bridging the gap between prediction and understanding of information cascades," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1149–1158.

[15] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, "Cascade dynamics modeling with attention-based recurrent neural network," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2985–2991.

[16] Z. Pan, Z. Huang, D. Lian, and E. Chen, "A variational point process model for social event sequences," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 173–180.

[17] H. Mei and J. M. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6754–6764.

[18] 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 *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1555–1564.

[19] W. Wu, J. Yan, X. Yang, and H. Zha, "Decoupled learning for factorial marked temporal point processes," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2516–2525.

[20] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv: 1810.04805*.

[22] Y. Zheng et al., "Geolife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, Feb. 2010.

[23] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[24] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modelling the scaling properties of human mobility," *Nature Phys.*, vol. 6, no. 10, pp. 818–823, 2010.

[25] L. A. Maglaras and D. Katsaros, "Social clustering of vehicles based on semi-markov processes," *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 318–332, Jan. 2016.

[26] E. Bacry, T. Jaisson, and J. Muzy, "Estimation of slowly decreasing hawkes kernels: Application to high-frequency order book dynamics," *Quantitative Finance*, vol. 16, pp. 1–23, 2016.

[27] Y. Abu Farha, A. Richard, and J. Gall, "When will you do what?-anticipating temporal occurrences of activities," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5343–5352.

[28] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[29] J. A. Thomas and T. M. Cover, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.

[30] P. Besse, B. Guillouet, J. M. Loubes, and R. Franois, "Review and perspective for distance based trajectory clustering," *Comput. Sci.*, vol. 47, no. 2, pp. 169–179, 2015.

[31] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 458–461.

[32] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with GPS history data," in *Proc. Int. Conf. World Wide Web*, 2010, pp. 1029–1038.

[33] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 363–372.

[34] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1043–1051.

[35] W. Wang, H. Yin, X. Du, Q. V. H. Nguyen, and X. Zhou, "TPM: A temporal personalized model for spatial item recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 9, pp. 1–25, Nov. 2018.

[36] Q. Yuan, G. Cong, K. Zhao, Z. Ma, and A. Sun, "Who, where, when, and what: A nonparametric Bayesian approach to context-aware recommendation and search for twitter users," *ACM Trans. Inf. Syst.*, vol. 33, no. 1, pp. 1–33, 2015.

[37] G. Biczok, S. D. Martínez, T. Jelle, and J. Krogstie, "Navigating mazemap: Indoor human mobility, spatio-logical ties and future potential," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2014, pp. 266–271.

[38] F. Wu, H. Wang, and Z. Li, "Interpreting traffic dynamics using ubiquitous urban data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, Art. no. 69.

[39] Y. Ogata, "Space-time point-process models for earthquake occurrences," *Ann. Inst. Stat. Math.*, vol. 50, no. 2, pp. 379–402, 1998.
[40] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015.
[41] Z. Ghahramani, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, 1997.
[42] F. Simini, G. Barlacchi, M. Luca, and L. Pappalardo, "A deep gravity model for mobility flows generation," *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 6576.
[43] J. Jia and A. R. Benson, "Neural jump stochastic differential equations," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 32, 2019.
[44] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 31, 2018.
[45] E. Toch, B. Lerner, E. Ben-Zion, and I. Ben-Gal, "Analyzing large-scale human mobility data: A survey of machine learning methods and applications," *Knowl. Inf. Syst.*, vol. 58, pp. 501–523, 2019.
[46] M. Luca, G. Barlacchi, B. Lepri, and L. Pappalardo, "A survey on deep learning for human mobility," *ACM Comput. Surv.*, vol. 55, no. 1, pp. 1–44, 2021.
[47] H. Barbosa et al., "Human mobility: Models and applications," *Phys. Rep.*, vol. 734, pp. 1–74, 2018.
[48] S. Isaacman et al., "Human mobility modeling at metropolitan scales," in *Proc. 10th Int. Conf. Mobile Syst.*, 2012, pp. 239–252.
[49] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 546–563.
[50] P. Zhao, H. Jiang, J. Li, F. Zeng, and G. Zhang, "Synthesizing privacy preserving traces: Enhancing plausibility with social networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2391–2404, Dec. 2019.
[51] L. Pappalardo and F. Simini, "Data-driven generation of spatio-temporal routines in human mobility," *Data Mining Knowl. Discov.*, vol. 32, no. 1, pp. 787–829, 2017.
[52] X. Liu, H. Chen, and C. Andris, "trajGANs : Using generative adversarial networks for geo-privacy protection of trajectory data (vision paper)," *Future Gener. Comput. Syst.*, vol. 142, pp. 25–40, 2023.
[53] H. Y. Song, M. S. Baek, and M. Sung, "Generating human mobility route based on generative adversarial network," in *Proc. FedCSIS*, 2019.

Huandong Wang (Member, IEEE) received the BS degree in electronic engineering and his 2nd BS degree in mathematical sciences from Tsinghua University, Beijing, China, in 2014 and 2015, respectively, and the PhD degree in electronic engineering from Tsinghua University, Beijing, China, in 2019. He is currently a research associate with the Department of Electronic Engineering, Tsinghua University. His research interests include urban computing, urban simulation, mobile Big Data mining, and machine learning.

Qizhong Zhang received the BS degree in mathematics and applied mathematics from Tsinghua University, Beijing, China, in 2022. He is currently working toward the master's degree with the Department of Statistics of the University of Chicago. His research interests include Bayesian Inference, generative models, and machine learning.

Yuchen Wu received the BE degree in communication engineering from UESTC, in 2020. He is currently working toward the master's degree with the Department of Electrical and Computer Engineering of Carnegie Mellon University. His research interests focus on mobile Big Data, mobility prediction and deep leaerning.

Depeng Jin (Member, IEEE) received the BS and PhD degrees from Tsinghua University, Beijing, China, in 1995 and 1999, respectively, both in electronics engineering. Now he is an associate professor with Tsinghua University and vice chair of Department of Electronic Engineering. He was awarded National Scientific and Technological Innovation Prize (Second Class) in 2002. His research fields include telecommunications, high-speed networks, ASIC design and future internet architecture.

Xing Wang received the MS degree from the Beijing Normal University, Beijing, China, in 2018. She is currently a technical researcher with the Artificial Intelligence and Intelligent Operation Center of China Mobile Research Institute. Her research interests include network intelligence, network topology optimization, spatial-temporal data mining, reinforcement learning, and deep learning.

Lin Zhu received the PhD degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2008. She is currently senior technical researcher in artificial intelligence algorithm of China Mobile Research Institute. Her research interests include intelligent network, intelligent pipeline and communication network. She has applied for more than 30 national invention patents in related filed.

Li Yu is the deputy general manager and senior engineer of artificial intelligence and intelligent operation center of China Mobile Research Institute. The main research direction is cutting-edge mobile communication technology, network intelligence, Big Data and IT technology.