Simulating Human Mobility with a Generative Trajectory Generation Model

Chen Chu[a,b], Hengcai Zhang[a,b*] and Feng Lu[a,b]
a State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, China; b University of Chinese Academy of Sciences, Beijing, China

Chen Chu is a Master student at State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences. His research interests focus on interpretable spatiotemporal prediction, human mobility, and spatiotemporal data mining. chuchen0411@igsnrr.ac.cn.

Hengcai Zhang is currently an Associate Professor at the Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences. He received his Ph.D. degree from the Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences in 2013. His research interest mainly focuses on moving objects database, urban computing and spatial-temporal data mining. zhanghc@lreis.ac.cn.

Feng Lu is a Professor with the Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences. His research interests cover trajectory data mining, computational transportation science and location-based services. luf@lreis.ac.cn.

Simulating Human Mobility with a Generative Trajectory Generation Model

Abstract: Most of the current mobility modeling methods are specially designed to solve one specific task, which leads to questions regarding generalizability. Inspired by the bloom of foundation models, we proposed a generalized Trajectory Generation framework based on Diffusion Model (TrajGDM) to capture the universal mobility pattern in a trajectory dataset by learning the trajectory-generation process. The process is modeled as a step-by-step uncertainty reducing process, which estimates uncertainty with a trajectory generator network. We compared the proposed trajectory generation method with five benchmarks on two public trajectory datasets. The result showed that the similarity between generated and real trajectory movements measured by Jensen-Shannon Divergence improved by at least 50.3% on both datasets. Moreover, we applied zero-shot inferences to two basic trajectory tasks: trajectory prediction and trajectory reconstruction. The zero-shot prediction accuracy of our model is up to 23.4% higher than the benchmark, and the reconstruction accuracy improves by a maximum of 25.6%. The universal mobility pattern that suits for solving multiple trajectory tasks is verified by the zero-shot multi-tasks inferring ability of our model. At last, the study gave insights into the generation of trajectories by exploring the way the model maps trajectory's representation in the latent space into reality.
Keywords: Human mobility; Trajectory generation; Diffusion model; Geo-Foundation model

Introduction

Human mobility modeling has a wide range of tasks, including transportation management(Song et al., 2016), trajectory prediction(Li et al., 2020), human mobility pattern mining(Ji et al., 2023), urban planning(Li et al., 2021), epidemic spread simulation(Feng et al., 2020), privacy protection(Rao et al., 2020), etc. Modeling the mobility pattern properly is essential to all trajectory tasks. Currently, the most common way to solve different human mobility modeling tasks is to build various models, with each model learning a part of the mobility pattern. For example, in a trajectory prediction model, the model aims at capturing the mobility pattern between the observed trajectory and the predicted point. Despite the fact that the mobility pattern is the same between every two points in the observed trajectory, the model is not able to learn from the observed part because of the limitation of its learning object. Even though the human mobility pattern is universal in the entire dataset, it cannot be learned and used in various tasks. To capture the universal mobility pattern in a trajectory dataset, a model is required to set a corresponding learning object. Simulating human mobility is one way to achieve that, it aims at generating a synthesized human mobility dataset based on the mobility pattern shown in the real one(Jiao et al., 2022). Accurately simulation of human mobility requires a model to learn the generation process of trajectories, which directly reflects the universal mobility pattern. Thus, learning the universal pattern through simulating the generation process of a trajectory could provide a solution for solving most of the human mobility modeling tasks.

Plenty of trajectory generation studies model human mobility with mechanistic modeling methods(Isaacman et al., 2012; Jiang et al., 2016; Simini et al., 2021). These studies model human mobility based on researchers' prior knowledge in the human mobility pattern and assume every movement of people has an explicit purpose. For example, researchers assume people's activity is affected by their distance to home and that everyone has a fixed commute distance from home to work(Isaacman et al., 2012). These kinds of strong assumptions limit models' ability in generating detailed trajectories and also make the model fail in modeling the randomness of human mobility. Moreover, to map the generated individual activity into geography distribution, most of the mechanistic models employ external location features, such as land use, points of interest, or divided living and working places(Yin et al., 2018). The introduction of external features also constrains the transfer ability of mechanistic models.

On the other hand, with the deep generative model achieves great success in many generation tasks, deep learning based generative models like Variational Autoencoder (VAE)(Kingma & Welling, 2013), Generative Adversarial Network (GAN)(Goodfellow et al., 2020) have been widely used in crowd trajectory generation. The generative model raised the concept of latent space(Doersch, 2016). By sampling from the continuous latent distribution, a deep generative model can generate numerous new trajectories that are different from any trajectory in the training dataset, which promises the diversity of generation. Diversity is of vital importance for trajectory generation, as the movement activity of each moving object is unique and inherently stochastic(Song et al., 2010). Moreover, different from the mechanistic model, deep generative models do not rely on the assumption to human movement. They extract the mobility pattern by learning the probabilistic distribution of each movement decision from the real dataset directly. Therefore, rather than just modeling residents' regular work-to-home behavior, these model-free methods can extract the hidden pattern of human mobility and can be easily applied to many scenarios. However, there still are some limitations in current generative models in trajectory generation. VAE failed to generate qualified trajectories because the noise adding operation is fuzzy for a trajectory with explicit meaning(Chen et al., 2021), which is unacceptable for the downstream utility of generated trajectory. As the most well-known structure for the deep generative model, GAN is notoriously difficult to train(Karnewar & Wang, 2020; Kodali et al., 2017). Moreover, instead of sampling from the latent space to get the latent representation of the upcoming generated trajectory, most of the GAN based trajectory generation models abandon the usage of latent space(Feng et al., 2020; Jiang et al., 2023; Yu et al., 2017), and start 'generating' a trajectory with a fixed start token. With the absence of latent space, these models lose the ability to generate diverse trajectories. The variation of their output is fixed once the model is finished training. They can no longer be called as a generative model. We will explain this statement further in the trajectory generation section.

Recently, Denoising Diffusion Probabilistic Model (DDPM) has bloomed in many realms, including image generation(Dhariwal & Nichol, 2021; Ho et al., 2020), text generation(Gong et al., 2022; X. Li et al., 2022), time series modeling(Tashiro et al., 2021) and trajectory prediction(Gu et al., 2022; Mao et al., 2023), etc. The diffusion model aims at modeling the generation process of a sample step by step, and the process starts from a latent representation sampled from the latent space.

In this study, we proposed a generative Trajectory Generation Framework based on the Diffusion Model named TrajGDM, which aims at modeling the universal human mobility pattern by simulating the generation process of the trajectory. The basic conception of our model is shown in figure 1.

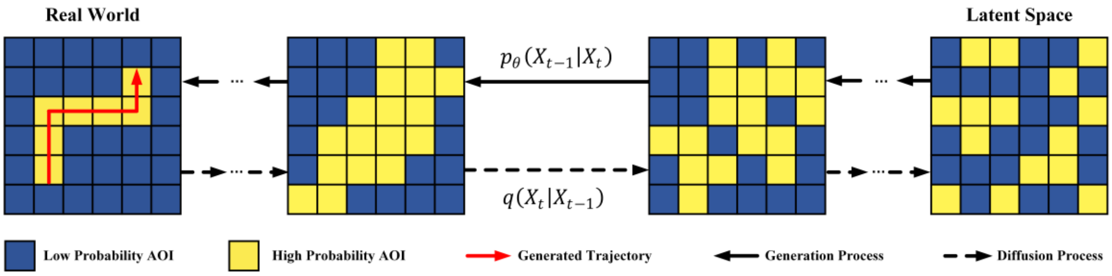


Figure 1. The intuition of the TrajGDM.

According to figure 1, the trajectory generation process is modeled to be an uncertainty reducing process. X_t denotes the trajectory at step t of the trajectory generating process p, and diffusion process q. We employed a deep learning network with parameter θ to estimate the uncertainty in X_t based on X_{t-1}. A detailed trajectory is generated after T steps of uncertainty reduction. To generate a realistic trajectory, the trajectory generator with parameter θ is required to capture the universal mobility pattern suits the whole trajectory dataset. While the human mobility pattern plays an important role in all trajectory tasks, including trajectory prediction, trajectory reconstruction, etc. By learning the universal mobility pattern in a trajectory dataset, the trajectory generation model is expected to be able to solve multiple trajectory problems without extra training. Therefore, to verify that it is possible to solve multiple trajectory tasks with the universal mobility pattern, we performed zero-shot trajectory prediction and reconstruction experiments using our model. Our contributions could be summarized as follows:

- We proposed a novel human mobility modeling method named TrajGDM, which models trajectory generation as a process the uncertainty in the trajectory is gradually removed. A trajectory generator is proposed to predict the uncertainty in a trajectory. Moreover, we defined a trajectory diffusion process and a trajectory generation process to train the trajectory generator and generate a realistic dataset with that.
- We compared the performance of our human mobility modeling method with five strong baselines in two datasets. Our model achieves great improvement in simulating individual mobility and other metrics while promising the diversity of generated trajectories. Furthermore, by visualizing the trajectory generation process and exploring the latent space of the model, a new perspective on the trajectory generation process is provided.
- We conducted zero-shot inferences on two basic trajectory tasks: trajectory prediction and trajectory reconstruction. The zero-shot inferring ability of our model verifies the utility of the universal mobility pattern captured through learning the generation process of a trajectory. It also demonstrates the potential of our method in serving as a foundation model in human mobility modeling.

The rest of the paper is organized as follows. Section 2 introduces related works in modeling human mobility. In section 3, we proposed our TrajGDM framework. In section 4, we provide detailed information for the experiment datasets and model implementation. We introduce 5 baselines used for comparison, and analyze the experiment result of trajectory generation, prediction and reconstruction, then we interpret the latent space of trajectory. Eventually, we conclude our work in section 5.

Related Work

Modeling human mobility has always been the most challenging job in trajectory data mining. Most of the existing methods extract the human mobility pattern by building a sequential model. The Recurrent neural networks and the attention mechanism are often used to model the temporal relationship in a trajectory(Bao et al., 2021; Feng et al., 2020; L. Li et al., 2022). As for the spatial relationship between locations, generally, a normal embedding function is employed in most models(Feng et al., 2018). Because it is well adapted to all task scenarios, moreover, to achieve better performance, spatial information is necessary to be provided to models as prior knowledge. Graph structure is often used (Salzmann et al., 2020). Moreover, location encoding is another way to represent the spatial relationship(Mai, Janowicz, et al., 2022). Representing a location by its neighbor fuzzy locations(Li et al., 2020) or Markov transition matrix(Wang et al., 2019) can be regarded as types of fixed location encoding methods. On the other hand, methods like Space2Vec(Mai et al., 2020) and Sphere2Vec(Mai, Xuan, et al., 2022) focus on trainable encoding based on downstream tasks. While most of the human mobility models were built to solve a specific problem, like human movement prediction(Bao et al., 2021), trajectory reconstruction(Li et al., 2019; Liu et al., 2018; Qi et al., 2020), they failed to capture the universal mobility pattern that applied to the entire trajectory.

Trajectory generation aims at simulating human mobility by generating a synthesized trajectory dataset that shows the same mobility pattern as the trained dataset. Current generation methods are mostly based on a GAN structure with a trajectory prediction model. SeqGAN employs a gated recurrent unit (GRU) network as its generator and starts generating with a trainable start token. It designed a discriminator that uses the reward signal in reinforcement learning to guide the generator(Yu et al., 2017). Movesim follows the combination of GAN and reinforcement learning. It starts generating by sampling from a historical origin matrix and a trajectory prediction model is employed as its generator to generate the rest part of a trajectory. A discriminator is designed to return rewards(Feng et al., 2020). TS-TrajGen also based on the GAN structure, it matches trajectory points with the road network and starts generating by sampling a pair of origin and destination (OD) from the historical matrix. It trained a generator to output the parameter of the A* algorithm between the OD. Then it can output a continuous trajectory using the path search algorithm(Jiang et al., 2023). However, these GAN based trajectory generating methods abandoned the usage of the latent space, which is important for a generative model to generate diverse samples. Only a few trajectory generation models are designed to generate trajectories from latent vectors. TrajGAN converts trajectory points into pixels in a picture of the whole region and employs the standard CNN based GAN structure to generate trajectory(Ouyang et al., 2018). But the point to pixel conversion is not suitable for all tasks. SVAE combines VAE and sequence to sequence model and generates trajectory by sampling from the designed latent space(Huang et al., 2019). In this study, we proposed a novel trajectory generation model, the model is able to generate diverse and realistic trajectories.

Preliminary

In this section, we define the formation of trajectory and trajectory dataset. We also define the trajectory generation and other related tasks.

Definition 1. Trajectory. A trajectory X is formed by a serial of trajectory points $loc = (x, y)$, where x, y are the coordinates of the trajectory point. n denotes the length of a trajectory.

$$X = [loc_1, loc_2, \dots, loc_n]$$

Definition 2. Trajectory dataset. Numbers of trajectories record the human mobility activity in a region forms a human mobility dataset $\mathbb{D}$. N denotes the number of trajectories recorded in the dataset.

$$\mathbb{D} = \{X^1, X^2, \dots, X^N\}$$

Definition 6. Trajectory latent space. The latent space of a trajectory dataset is a high dimension feature space $\mathcal{Z}$, where each trajectory in the dataset $\mathbb{D}$ has its corresponding representation in it. Generally, we need to define a mapping function $\mathcal{M}$ to map all trajectories in $\mathbb{D}$ into their corresponding locations in $\mathcal{Z}$. The function can be defined as:

$$\mathcal{Z}_{\mathbb{D}} = \mathcal{M}(\mathbb{D})$$

where $\mathcal{Z}_{\mathbb{D}}$ denotes the latent representation of dataset $\mathbb{D}$.

Definition 3. Trajectory generation. The target of trajectory generation is to generate a realistic dataset $\widehat{\mathbb{D}} = \{\hat{X}^1, \hat{X}^2, \dots, \hat{X}^N\}$, by sampling trajectories form the latent space $\mathcal{Z}_{\mathbb{D}}$. It can be formulated as following:

$$\widehat{\mathbb{D}} = \mathcal{F}_\theta(\mathcal{Z}_{\mathbb{D}})$$

where the trajectory generation model $\mathcal{F}$ learn the mobility pattern in the trajectory dataset $\mathbb{D}$ by its trainable parameter θ. Then it generates a synthesized dataset $\widehat{\mathbb{D}}$, which is expected to show similar trajectory performance with $\mathbb{D}$ from all aspects. The basic measurement formula is as following:

$$\min_\theta JSD\,[M(\mathbb{D})||M(\widehat{\mathbb{D}})]$$

JSD is the Jensen-Shannon Divergence (JSD), which is generally used to measure the difference between two distributions. M could be different metrics used to measure the trajectory performance in a dataset. The similarity could be evaluated from many aspects, like the similarity of individual movements, or geography distributions of all trajectory points. We will introduce the evaluation method more specifically in the evaluation metrics section.

Definition 4. Trajectory prediction. Trajectory prediction aims at predicting the future movement of a moving object. It can be defined as following:

$$loc_{n-l}, \dots, loc_n = \mathcal{P}(loc_1, \dots, loc_{n-l})$$

where trajectory points $loc_{1:n-l}$ are the previous location points from time 1 to time $n - l$, which have been observed at the moment. $loc_{n-l:n}$ denotes the future trajectory points, and l is the length of points to be predicted. $\mathcal{P}$ is the model used for prediction.

Definition 5. Trajectory reconstruction. Trajectory reconstruction aims to fill up the missing points in a trajectory. The definition of trajectory reconstruction can be formulated as following:

$$\bar{u}_i, \dots, \bar{u}_{i+1} = \mathcal{R}(\dots, loc_{i-1}, u_i, \dots, u_{i+l}, loc_{i+l+1}, \dots)$$

where u_i denotes the missing trajectory point i and $\bar{u}_i$ denotes the corresponding trajectory points reconstructed by model $\mathcal{R}$. l denotes the length of missing trajectory points to be reconstructed.

Trajectory generation Framework

Human mobility has natural uncertainty(Liu et al., 2022; Yu et al., 2023). We model the generation of a trajectory as a process that the uncertainty in the trajectory is gradually removed. And in order to train a model with the ability to predict and remove the uncertainty in human mobility, a trajectory diffusion process is constructed to simulate the uncertainty adding process. The general architecture of our TrajGDM framework is shown in figure 2.

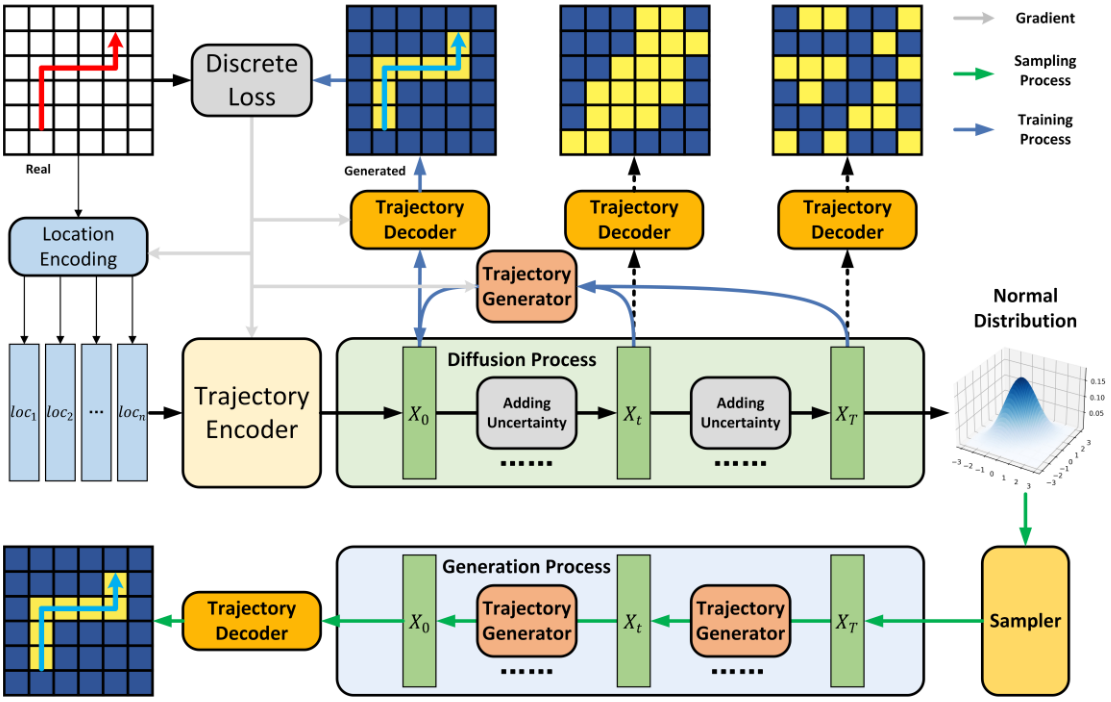


Figure 2. Structure of the TrajGDM framework

In figure 2, there are two important parts in the framework, the diffusion process and the generation process. The generation process assumes the generation of a trajectory as an uncertainty removing process. More specifically, in the initial stage of human mobility, only a few essential factors are formed by the person, such as destination and origin. These factors are represented by a latent vector. And the latent representation of all trajectories in the dataset is assumed to follow a normal distribution. At this stage, the entire trajectory is full of uncertainty, which means the detailed route that satisfies all key factors is unknown. Then according to these key factors, the person gradually decides its each movement. We designed a deep learning network named trajectory generator to estimate the uncertainty in the trajectory X_t, the network is denoted as $\mathcal{G}$. The trajectory generator predicts and removes the uncertainty in the current trajectory. Through T times of iterations in the trajectory generation process, the uncertainty in the trajectory is basically removed, and a detailed trajectory that corresponds with the key factors in the representation is generated. To train this trajectory generator $\mathcal{G}$, we construct a diffusion process based on a Markov chain. In the diffusion process, gaussian noise is added step by step in T steps in total to simulate the uncertainty in a trajectory. And we make our trajectory generator to learn to estimate noise added and recover the original real trajectory X_0. In the rest of the section, we will first introduce how we encode a trajectory into a hidden representation using the location encoding function $\mathcal{P}$ and trajectory encoder $\mathcal{E}$. Then we will introduce specific definitions of the trajectory diffusion process and the trajectory generation process. And we will introduce our training process, which combines diffusion and generation together to train the trajectory generator. After that, the sampling process is proposed to generate trajectory data by sampling latent vectors from the latent space. Eventually, the structure of the trajectory decoder $\mathcal{D}$ and trajectory generator network $\mathcal{G}$ will be presented.

Trajectory Encoder

In this work, we model human mobility in a format of semantic trajectory represented by the geography grid, which makes the model applicable to most of the trajectory datasets. To encode a trajectory discrete represented into a continuous feature space X_0, we proposed a trajectory encoder $\mathcal{E}$, The formula of the trajectory encoding function $\mathcal{E}$ is as following:

$$X_0 = \mathcal{E}[\mathcal{P}(X)]$$

$$\mathcal{E}(\mathcal{P}(X)) = LSTM[\mathcal{P}(x_1, y_1), \mathcal{P}(x_2, y_2), \dots, \mathcal{P}(x_n, y_n)]$$

where we encode the trajectory with a $LSTM$ network, and take the output of the $LSTM$ network as the representation of the trajectory, which promises the serial relationship between trajectory points is also encoded into the feature space X_0 of corresponding trajectory. We also proposed a trainable location encoding function $\mathcal{P}$ to provide the model with the awareness of the spatial relationship between locations. The location encoding function to map point (x, y) is formulated as following:

$$\mathcal{P}(x, y) = W_{\mathcal{P}} * Concat(E(x, y) * \gamma, E(x-1, y), E(x, y-1), E(x+1, y), E(x, y+1))$$

where $W_{\mathcal{P}}$ is the parameter of the location encoding function, $Concat$ is the concatenate function, E is the general embedding function that embeds the index of a location into feature space, and γ is the hyperparameter of the location encoding function. Considering the movement of a trajectory is continuous in real-world space, it is important to provide the adjacent relationship to the model so that the model learns from people's prior knowledge in the spatial relationship, rather than starts learning the relationship between locations from nothing. Therefore, instead of representing a location just by its embedding vector, the proposed location embedding function combines the adjacent locations of loc. The function employs a hyperparameter γ to emphasize the actual location loc and employs a trainable weight matrix $W_{\mathcal{P}}$ to learn complex spatial information from the adjacent relationship. After encoding each location in a trajectory into a feature space, the trajectory encoder $\mathcal{E}$ is employed to construct the representation X_0 of the entire trajectory X. The trajectory encoder is of vital importance to generate a continuous trajectory, because it integrates the representation of points into the representation of a trajectory.

Trajectory Diffusion Process

We construct the trajectory diffusion process to gradually add uncertainty to a trajectory. To promise the trajectory can be reconstructed after adding the uncertainty, we build a T steps Markov chain. In each step, the uncertainty is added in the form of random Gaussian noise, which is very small. After T times of diffusion, the encoded trajectory representation X_0 is mapped into a latent representation X_T, which follows a latent distribution $q(X_{1:T}|X_0)$. The stepwise diffusion process from X_0 to X_T can be formulated as following:

$$q(X_{1:T}|X_0) = \prod_{t=1}^{T} q(X_t|X_{t-1})$$

$$q(X_t|X_{t-1}) = N(X_t; \sqrt{1-\beta_t}X_0, \beta_t\mathbf{I})$$

where $q(X_t|X_{t-1})$ is one step in diffusion process, scheduler $\beta_t \in (0,1)$ is variance of the sample distribution and it is used to control the scale of the uncertainty added, it is a varied schedule differed from β_1 to β_T. To simplify the calculation, according to the notable property of the diffusion process, we can sample trajectory X_t at an arbitrary timestamp t in a closed form as following.

$$q(X_t|X_0) = N(X_t; \sqrt{\bar{\alpha}_t}X_0, (1-\bar{\alpha}_t)\mathbf{I})$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. According to the formula, we can directly get the noised trajectory representation X_t by sampling from a X_0 based Gaussian distribution. The formula of sampling is as following,

$$X_t = \sqrt{\bar{\alpha}_t}X_0 + (1-\bar{\alpha}_t)\epsilon$$

where ϵ is a random Gaussian noise, and it is regarded as the uncertainty added on the trajectory X_0. Considering $\alpha_t < 1$, when the diffusion step t is large enough, the $\bar{\alpha}$ will be sufficiently close to 0, which leads to the $q(X_t|X_0)$ converge to a standard Gaussian distribution. This property ensures all trajectory representations in the training dataset $\mathbb{D}$ will be projected to the same latent distribution. And the good property of normal distribution also ensures that we can easily sample from it in a controllable way. This attribute is important for many downstream tasks of human mobility modeling.

Trajectory Generation Process

After the uncertainty was added, all trajectories were mapped into a latent distribution following a normal distribution. The trajectory generation process is employed to generate a trajectory from a latent vector representation sampled from the latent distribution. Considering the diffusion process is a parameter fixed Markov process, which adds a small bit of uncertainty to the trajectory in each step, the trajectory generator only needs to learn to estimate the uncertainty added in each step. The generation process is also built as a T steps process, so the model only needs to estimate a small part of the uncertainty in each step. And after T times of iterations, the model distribution $p_\theta(X_0)$ is formed. The generating process can be formulated as following:

$$p_\theta(X_{0:T}) = p(X_T) \prod_{t=1}^{T} p_\theta(X_{t-1}|X_t)$$

$$p_\theta(X_{t-1}|X_t) = N(X_{t-1};\ \mu_\theta(X_t,t), \sum\nolimits_\theta(X_t,t))$$

where $p(X_T) = N(\mathbf{0}, \mathbf{I})$ is the latent distribution of all trajectories in the training dataset, $p_\theta(X_{t-1}|X_t)$ is the estimation of x_{t-1} given x_t by model $\mathcal{G}$ with the parameter θ. The variance term of the Gaussian transition is manually set as $\sum_\theta(X_t,t) = \tilde{\beta}_t\mathbf{I}$ and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$, which has been proved to have good performance in practice(Ho et al., 2020).

Training Process

Now we have defined the trajectory diffusion process to add uncertainty to a trajectory and the trajectory generation process to remove that. For the next step, we need to know how to combine two processes together, and define a training object. So, a trajectory generator could learn from the added noise in the diffusion process and use the learned pattern to generate a trajectory through generating process. To ensure the generated trajectory follows the real human mobility pattern in the real dataset, the training objective is defined to minimize the difference between the trajectory generated from the model distribution $p_\theta(X_0)$ and the data distribution $q(X_0)$ reflected by real trajectory data. Therefore, our training object is to maximize their variational lower bound as following,

$$\max_\theta \mathbb{E}_{q(x_0)}[\log p_\theta(x_0)] \le \max_\theta \mathbb{E}_{q(x_0,\ldots,x_T)}[\log p_\theta(x_{0:T}) - \log q(x_{1:T}|x_0)]$$

As we have modeled the trajectory generation process as Gaussians with trainable mean functions and fixed variances, the objective above can be simplified as following.

$$\min_\theta \mathbb{E}_{x_0,\epsilon}[\frac{\beta_t^2}{2\tilde{\beta}_t\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \mathcal{G}_\theta\left(\sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2]$$

where $\mathcal{G}_\theta(X_t, t)$ is the uncertainty predicted by trajectory generator $\mathcal{G}$, at generation step t to approximate ϵ. The objective can be simply interpreted as minimizing the difference between the uncertainty ϵ added to the observed trajectory X_0 in each step of the trajectory diffusion process with the uncertainty $\mathcal{G}_\theta$ estimated by the trajectory generator $\mathcal{G}$.

However, unlike the color of pixels in a picture, which is naturally distributed in a continuous feature space, the representation of geographic locations generally is originally discrete (such as AOI (Area of Interests), cellular network, grids, etc.). In a discrete feature space, it is more important to model the relationships and the differences between locations rather than their absolute position in feature space. The original training loss of the diffusion model derives gradient from minimizing the difference between the estimation of the added noise ϵ_θ in each step with the truly add one ϵ, which has been proved to be efficient in modeling the continuous feature space(Rombach et al., 2022). While the discrete state space needs a more distinct training loss to direct the optimization of the relationship between locations in feature space. Therefore, besides of using location encoding to map the discrete location representation into a continuous feature space, we also modify the training process, so that the model can focus on learning the relationship of locations through the trajectory. First, we directly calculate the estimation of X_0 according to the $q(X_t|X_0)$ formula in the trajectory diffusion process.

$$\bar{X}_0 = \frac{1}{\sqrt{\bar{\sigma}_t}}X_t - (\sqrt{\frac{1}{\bar{\sigma}_t} - 1})\mathcal{G}_\theta(X_t, t)$$

Then we decode the estimated $\bar{X}_0$ with a trajectory decoder $\mathcal{D}$, so that the hidden representation is decoded to the probabilistic of each discrete location. Then we take the gradient from a SoftMax Cross-Entropy loss to train the network.

$$\min_\theta \mathbb{E}_{X,\epsilon\sim N(0,1),t}[-X * \log \mathcal{D}(\bar{X}_0)]$$

Therefore, our training algorithm can be summarized as algorithm 1.

Algorithm 1 Training algorithm

repeat:

$X_t \sim q(X_t | \mathcal{E}[P(X)])$

$t \sim Uniform(\{1,2,\ldots,T\})$

$\epsilon \sim N(\mathbf{0}, \mathbf{I})$

Take gradient from

$$\nabla_\theta || - X * \log \mathcal{D}\left(\frac{1}{\sqrt{\bar{\sigma}_t}} X_t - (\sqrt{\frac{1}{\bar{\sigma}_t} - 1}) \mathcal{G}_\theta(X_t, t)\right) ||$$

until converged

Sampling Process

As the requirement of the optimization function $\min_\theta \mathbb{E}_{x_0,\epsilon}$, $\mu_\theta(X_t, t)$ in trajectory generation process has to predict under the following form.

$$\mu_\theta(X_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(X_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \mathcal{G}_\theta(X_t, t))$$

Based on the formula, we can sample from $p_\theta(X_{t-1}|X_t)$ with the optimization goal that after t times of denoising the model distribution $p_\theta(X_0)$ is approximate to the data distribution $q(X_0)$. The formula for the trajectory generation process to infer X_{t-1} from X_t is as follows.

$$X_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(X_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \mathcal{G}_\theta(X_t, t)\right) + \tilde{\beta}_t z$$

where z is a random variable sampled from standard Gaussian distribution. According to the formula, once all the parameterized models were trained through the training process, we can estimate $\mathcal{G}_\theta(X_t, t)$ as the estimation of the uncertainty in X_t and generate a trajectory through the generation algorithm in algorithm 2.

Algorithm 2 Generating algorithm

$X_T \sim N(\mathbf{0}, \mathbf{I})$

for $t = T, \ldots, 1$ **do**

$\boldsymbol{z} \sim N(\mathbf{0}, \mathbf{I})$ **if** $t > 1$, **else** $\boldsymbol{z} = 0$

$X_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \mathcal{G}_\theta(X_t, t)\right) + \tilde{\beta}_t \boldsymbol{z}$

end for

$\hat{X} = \mathcal{D}(X_0)$

return $\hat{X}$

Trajectory generator

Now we have defined the diffusion process and the generation process, and we have proposed a training process to point out an optimization object to train the trajectory generator and a sampling process to use the trajectory generator to generate a new trajectory. Here comes the most important part of our human mobility modeling framework, a trajectory generator neural network is proposed to generate a trajectory by reducing its uncertainty. We designed a Transformer based trajectory generation network to capture the spatial-temporal relationship in a trajectory. The structure of the trajectory generator is shown in figure 3.

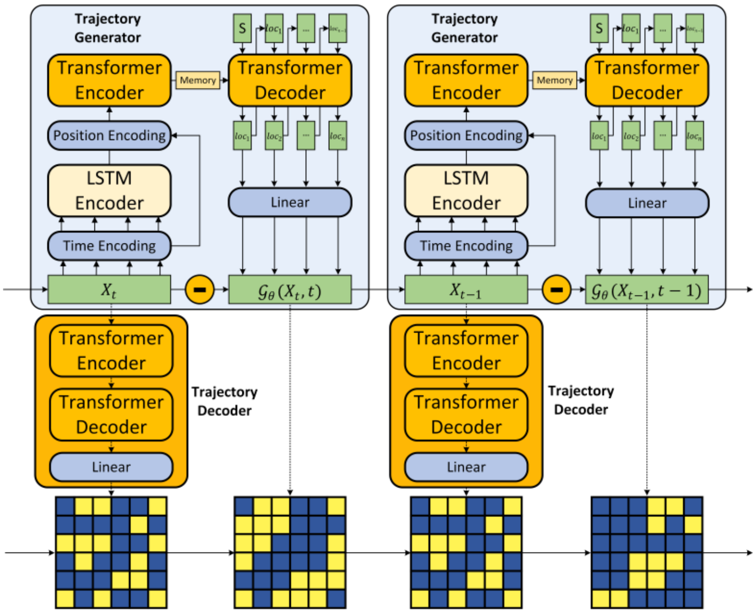


Figure 3. Structure of the trajectory generator and the trajectory decoder

Trajectory generation has a strict requirement for the serial pattern in the generated trajectory. The generated trajectory must be continuous and follow the mobility pattern in a real dataset. Therefore, we proposed our trajectory generator based on the sequence to sequence structure. As it has shown in figure 3, the trajectory generator takes X_t as input, which denotes the latent representation of the trajectory X at generation step t. A step encoding function is performed to encode the current step t into the latent feature X_t. The step encoding function is formulated as following,

$$TE(X_t, t) = X_t + \begin{cases} \sin\left(\frac{t}{10000^{\frac{2i}{d}}}\right), & if\ i\%2 = 0 \\ \cos\left(\frac{t}{10000^{\frac{2i}{d}}}\right), & if\ i\%2 \neq 0 \end{cases}$$

where d is the dimension of X_t and i is the dimension. Considering the amount of added uncertainty in different steps of the diffusion process is varying, Step encoding injects step information into the X_t, so that the model could learn to estimate how strong the uncertainty might be in the current step. Then a LSTM network and a transformer encoder are employed to model the serial relationship in the trajectory. Through experiment, we found out that the LSTM network is very important for generating a continuous trajectory, and the transformer module plays an important role in capturing the long-range dependency and accelerating the converging process. The encoded representation is passed to the transformer decoder by the memory of the encoder. Then the decoder decodes the uncertainty sequence step by step, it takes the output of the last moment as the input. The decoding process starts with the embedding of a start token, which promises the generated series will perform well in the serial relationship.

The uncertainty predicted by the trajectory generator is used to generate X_{t-1} by sampling from $p_\theta(X_{t-1}|X_t)$, which is represented as a minus symbol in figure 4. Through T steps iteration follows the sampling process, X_0 is predicted. The trajectory decoder $\mathcal{D}$ is used to decode the hidden representation of a trajectory into the possibility distribution of all candidate locations. We employed a transformer encoder and decoder structure, the input of decoder is X_t, the decoder decodes a trajectory from the memory of the encoder and another series of the embedding of the output token. We design the trajectory decoder as simple as possible, so that the model mostly relies on the trajectory generator to model human mobility by estimating the uncertainty in it.

Experiment

We evaluated our method on two public trajectory datasets. The two datasets are in different geographic scales and have distinct human mobility patterns. The density distribution of all trajectory points is shown in figure 4.

T-Drive: The dataset collected the real taxi GPS trajectory in Beijing, China. It contains the trajectory of 10,357 taxis during the period of Feb. 2 to Feb. 8, 2008. The average sample frequency is 2.95 minutes. Considering the data missing problem, we resampled the location of every taxi for every 5 minutes, so all trajectories in the dataset have a fixed time interval. We extracted the positioning points in the six-ring road, which account for 98.2% of all points in the dataset. Then the region in the six-ring road is divided into 27*27 grids by the square with 2000 meters edge length, which was decided by the mobility frequency and averaged moving distance in the dataset. Eventually, there are 169,984 trajectories recorded.

Geo-life: The dataset was collected from 182 people. The GPS trajectories record their mobility activity over 5 years. We also resampled all trajectories into a 5 minutes time interval and extracted points in six-ring road. Considering the mobility activity is relatively weak, the division is set as 500 meters, so there are 110*110 grids in total. At last, there are 79,360 trajectories left.

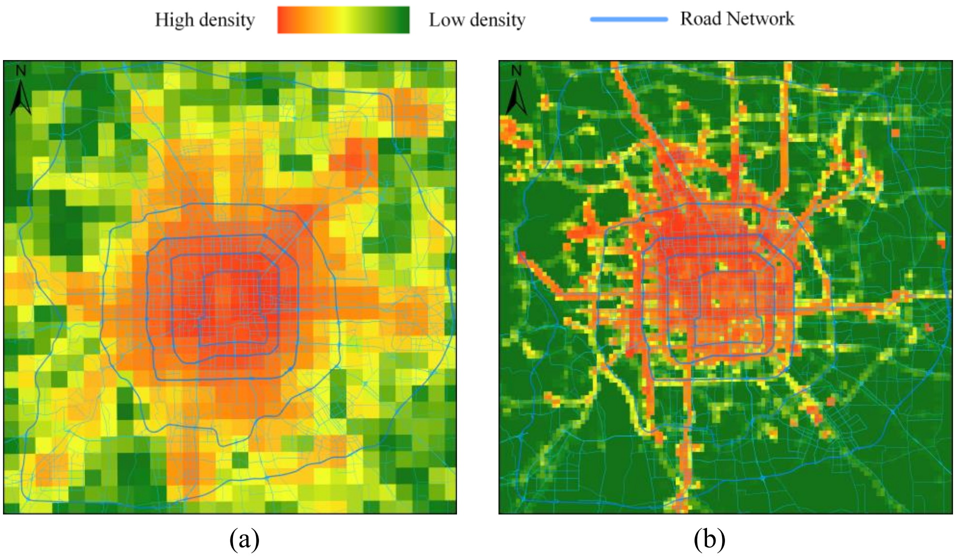


Figure 4. Geography distribution of trajectory points in (a) T-Drive,(b) Geo-life.

All the trajectories in each dataset are resampled to a 5 minutes time interval. We trained the model with the human mobility trajectories in one hour, which includes 12 trajectory points for every single trajectory. The dimension of location embedding is set as 512. The number of layers for trajectory encoder $\mathcal{E}$ is set as 2. And the number of hidden units in all the layers in the model is set as 512. The number of diffusion steps of the model is 1,000. The model was implemented by Pytorch and was trained on a Nvidia Titan V GPU.

Evaluation Metrics

We employed 5 different evaluation metrics to quantitatively evaluate the quality of generation performance and used Jensen-Shannon Divergence (JSD) to measure the similarity of the distribution of these metrics between the generated dataset with the real validation dataset. JSD is commonly used to evaluate the generation quality of a generative model(Feng et al., 2020; Theis et al., 2016). The definition of JSD is as following,

$$JSD(\mathbb{D}_1||\mathbb{D}_2) = \frac{1}{2}KL(\mathbb{D}_1||\frac{\mathbb{D}_1 + \mathbb{D}_2}{2}) + \frac{1}{2}KL(\mathbb{D}_2||\frac{\mathbb{D}_1 + \mathbb{D}_2}{2})$$

where $\mathbb{D}_1$ and $\mathbb{D}_2$ is two different distributions, and function KL measures the Kullback–Leibler divergence between two distributions. It can be formulated as:

$$KL(\mathbb{D}_1||\mathbb{D}_2) = \sum_{x\in\mathcal{X}} \mathbb{D}_1(x)\log\left(\frac{\mathbb{D}_1(x)}{\mathbb{D}_2(x)}\right)$$

where x denotes as a sample point in the discrete sample space $\mathcal{X}$. The smaller the JSD is, the more similar the two datasets are. We measured the JSD between the generated dataset and the real dataset of the following 5 metrics.

Moving distance (Moving): The mobility of a moving object is one of its most important characteristics for evaluating the generation quality of an individual trajectory. The metric measures the moving distance between two adjacent moments in a trajectory. It reflects the individual mobility pattern of all trajectories in a dataset.

Geography distribution (Distribution): Except for simulating the mobility of every single person, it is also important to generate a dataset that follows the same distribution as the real one. This metric evaluates the geography distribution of trajectory points in all generated trajectories. A similar geography distribution to the real one promises the basic utility of the generated dataset in downstream tasks.

Origin distribution (O-Dis) and **Destination distribution (D-Dis)**: Origin-Destination (OD) flow pattern mining is an important research task of trajectory data mining. Therefore, we formed these evaluation metrics to measure the usability of generated trajectory in OD flow pattern mining. The O-Dis measures the location distribution of the start locations in all trajectories in a dataset. The D-Dis measures the distribution of the end locations of all trajectories.

Diversity: From the view of the utility of a forged dataset, the diversity of generated trajectories is also very important. We measured the diversity of a trajectory dataset by counting the ratio of trajectories that appeared more than twice in the dataset. The lower the ratio is, the more diverse the dataset is. The quality of a generated dataset can be judged by whether the value of the metric is close to that in a real dataset.

Baselines

We compared the performance of our method with 5 strong baseline methods. To conduct a fair competition, the dimension of location embedding and the number of hidden units in all networks were set to be the same as that in our model. We also fine-tuned some of the default settings in these baselines to achieve the best performance.

FC-LSTM(Sutskever et al., 2014): FC-LSTM is a well-known discriminative model to deal with sequence to sequence tasks. It starts generating a trajectory by sampling a point from the density distribution of all trajectory points, then the rest part of the trajectory is generated using a step-by-step process. The dimension of location embedding and the number of hidden units in the LSTM were set to be the same as that in our model.

MoveSim(Feng et al., 2020): The model is one of the state-of-the-art methods for human mobility simulation. It is a discriminative model based on a GAN structure. The method samples a trajectory's start point from the population density distribution. Then the rest part of the trajectory is output from a specifically modified trajectory prediction model named SeqNet. The model is trained by the discriminator with a reinforcement learning technique. The dimension of location embedding and the number of hidden units in all the network were set as 512, which are also the same as ours.

SeqGAN(Yu et al., 2017): SeqGAN is the benchmark model for sequence generation. It combines GAN with reinforcement learning to provide the generator with a policy gradient. Two GRU networks are employed as its generator and discriminator. Different from the original GAN starts generating with a representation randomly sampled from the latent space, the model starts generating a trajectory with a trainable embedding of a start token. Therefore, the model only can be regarded as a discriminative model.

TrajVAE(Chen et al., 2021): TrajVAE is one of the few existing real generative models with a latent space for trajectory generation. Variational autoencoder is one of the most typical generative models. The model employed two LSTM networks as its encoder and decoder for VAE. It generates trajectory by decoding a sampled vector from a Gaussian latent space.

Generative SeqGAN: The model is a generative version modified from seqGAN. As we have mentioned previously, seqGAN is regarded as a discriminative model for its deficiency of the latent space. We modified the seqGAN by changing its start token by a latent representation sample from a normal distribution. So, the model generates a trajectory by mapping the latent distribution into reality.

Results

Trajectory Generation

We compared our trajectory generation model with five baseline methods on two datasets. To evaluate the quality of the generated trajectory, we calculated the distributions of five metrics in each generated dataset and compared them with their distribution in the real dataset. We also calculated the metrics' performance of a test dataset, which was randomly sampled from the real trajectory dataset. Considering the variation and complexity of human mobility, differences can be found even between two subsets sampled from the same dataset. The test dataset is used to evaluate this part of the difference. Through the comparison with the training dataset, we can evaluate the quality of trajectory data. The comparison result is shown in table 1.

Table 1. Performance comparison of all models on two datasets.

	T-Drive				
Metrics (JSD)	**Moving**	**Distribution**	**O-Dis**	**D-Dis**	**Diversity**
Test Dataset	0.02949	0.04773	0.05466	0.04458	0.04655

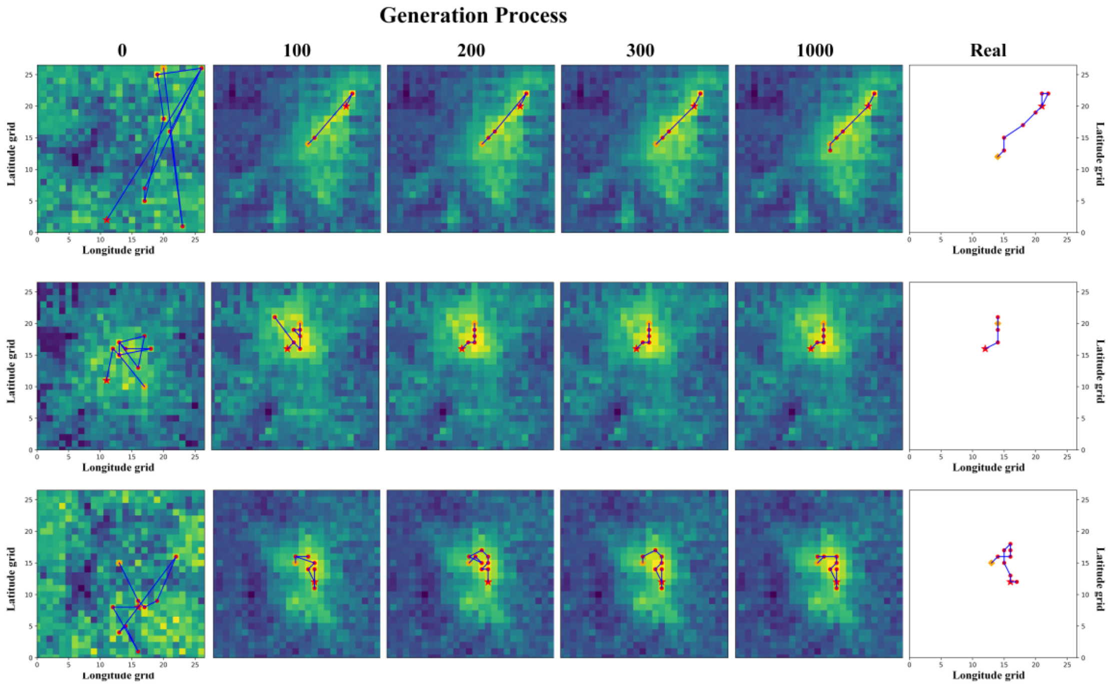
FC-LSTM	0.2227	0.2307	0.1603	0.27594	0.98228
MoveSim	0.3361	0.1763	**0.05449**	0.24498	0.22914
SeqGAN	0.1106	0.1418	0.1398	0.1732	0.1085
TrajVAE	0.3009	0.4557	0.3713	0.4831	0.0
Generative SeqGAN	0.3194	0.1890	0.4932	0.1701	0.0
TrajGDM	**0.05490**	**0.1171**	0.1358	**0.1326**	**0.01367**

	Geo-life				
Metrics (JSD)	**Moving**	**Distribution**	**O-Dis**	**D-Dis**	**Diversity**
Test Dataset	0.04491	0.08189	0.08631	0.09278	0.08621
FC-LSTM	0.2871	0.3565	0.2711	0.3436	0.4597
MoveSim	0.4487	0.1656	**0.06720**	0.2274	0.4225
SeqGAN	0.2640	0.2672	0.2627	0.2787	0.2782
TrajVAE	0.7283	0.2609	0.3804	0.2642	0.0
Generative SeqGAN	0.3812	0.4092	0.4504	0.3759	0.0
TrajGDM	**0.1142**	**0.1226**	0.1231	**0.1386**	**0.02226**

According to whether the model generates data by sampling from a latent space, models in table 1 can be divided into discriminative models and generative models. Models like FC-LSTM, seqGAN and MoveSim are regarded as discriminative models. Although MoveSim and seqGAN are based on GAN structure and output data using a so-called generator, the generator does not learn to map a trajectory from latent distribution. They employ discriminative trajectory prediction networks to generate trajectories by repeating the prediction process. The trajectory prediction network was trained to model the distribution of all predicted points, which can be denoted as $P(x_i|x_1, x_2, \dots, x_{i-1})$, where the emerging possibility of a point x_i is decided by its previous $i-1$ trajectory points. The distribution is theoretically different from the distribution of the entire trajectory points dataset $P(x_i)$. However, when the dataset is big enough and the division of predicted trajectory points x_i and observed trajectory points $x_1, x_2, \dots, x_{i-1}$ is randomly enough, these two distributions are overlapped in some extent. Therefore, it is possible for a discriminative model to output a set of trajectories that is likely to fit in with the distribution of the real trajectory dataset. It is the result of the overlap of two different distributions. Moreover, benefitting from the directly maximizing the probabilistic that each trajectory point emerges by $P_\theta(x_i|x_1, x_2, \dots, x_{i-1})$. The discriminative models can perform well in fitting the general geography distribution. This explains why most discriminative models show better performance in Distribution, O-Dis and D-Dis metrics. In order to start the predicting iteration, models like FC-LSTM and MoveSim start generating a trajectory by sampling a point from the density distribution of all trajectory points. This nonparametric starting method leads to their good performance on the O-Dis metric and further benefits their performance on other metrics that evaluate the general distribution of trajectory points. However, outputting the next trajectory point based on the maximized likelihood of the current observation also limits the diversity of outputs. Theocratically, once a discriminative model finished training, its output probability under every circumstance is fixed, and considering the starting method is also fixed, the model is very likely to output highly repetitive trajectories. This is reflected by the terrible performance of all discriminative models on the diversity metric.

Different from the discriminative models, the generative models, like TrajVAE, Generative SeqGAN, and TrajGDM, model the trajectory with a continuous latent space. The generation starts from sampling the latent space, different sample points in the latent space represent different trajectories. The generative models generate trajectories by mapping a representation vector from the latent space to a trajectory through the generation process. Theoretically, considering the latent space is continuous, a generative model is capable of generating countless variations of trajectory. This explains why all generative models show very low repetition rates in the diversity metric. Especially for our method, it has shown outstanding performance on the diversity metric, which is closest to the repetition rate of the validation trajectory dataset. For TrajVAE and Generative SeqGAN, their repetition rate is zero, this is caused by their poor performance in simulating the individual trajectory. Their generated trajectories are not continuous in space, so generated trajectories have a lower chance of being similar to each other.

While other generative models show bad performance on the moving distance metric, our method performs remarkably well and is better than all baseline methods on both datasets. In the Geo-life dataset, the decline ratio of the Moving metric is at least 57.2%. And the ratio comes to 50.3% in T-Drive. This means the trajectories we generated are more similar to the trajectories in the real world. Moreover, benefitting from directly modeling the trajectory distribution $P_\theta(x_1, x_2, \ldots, x_i)$, our method also shows great improvement on geography distribution metric. As for the OD distribution, besides of Movesim, which generates trajectory origins by sampling from the real distribution, our method still shows better performance than all other parametric methods. To understand how does a trajectory generate, we visualized trajectories in different steps of the generation process. We also picked out several real trajectories that are close to the generated one to make comparisons. The result is shown in figure 5. The background color represents the average generating possibility of all the points in the trajectory.



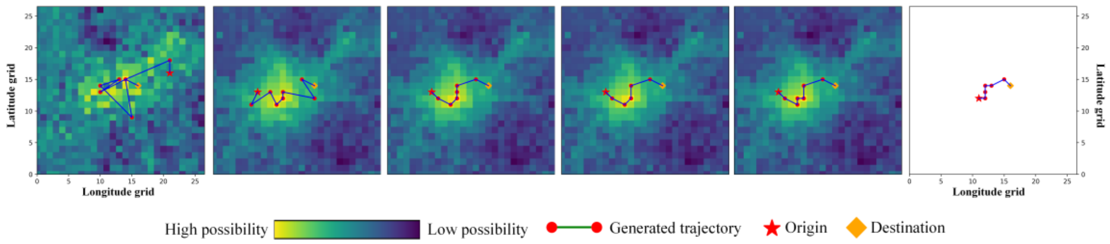


Figure 5. Generated trajectories in different steps of the generation process with similar real trajectories for demonstration.

Takes the generated trajectory in the first column of figure 5 as an example, it starts on the way to the city's airport, which is located in the northeast of the city, and the driver stays at the airport for minutes before going back downtown. The trajectory shows our method succeeds in simulating human mobility with strong moving intensity. The trajectories shown in the third and fourth columns show the simulated human mobility in the downtown area, in this area, cars have more complex moving activities, including turning around, more turns, etc.

Although the number of diffusion steps is set as 1000, the generated trajectories do not change significantly after the first 300 steps. This is because our optimization target is set as directly estimating X_0 from X_t and the trajectory points are nominated according to the output possibility distribution of the model. The possibility distribution is basically fixed after hundreds of steps.

An interesting phenomenon that occurred on all trajectories is that the origin and destination are settled down in the first place. Even though in the trajectory decoder, the generation of a trajectory's destination is partially based on its previous points due to its sequence to sequence structure, and the location points before the destination are still varying, destinations in most of the generated trajectories remained unchanged after the first few steps. The phenomenon demonstrates that important factors are determined in the first place in human movement. Details in a trajectory are specified later based on these key factors by our trajectory generator. This uncertainty reducing generation process is similar to the route planning process of a human being. This also illustrates that the trajectory generator in the generation process plays a more important role in determining the performance of a generated trajectory than the trajectory decoder.

Trajectory General Intelligence

To generate a realistic trajectory, the trajectory generator is required to capture the universal human mobility pattern that suits the whole trajectory dataset. While the human mobility pattern plays an important role in nearly all related trajectory tasks, including trajectory prediction, trajectory reconstruction, etc. It would be helpful if we could directly solve these problems with the universal mobility pattern captured by our trajectory generation model, rather than building numbers of different models, with each of them only capturing part of the pattern mobility and are only able to handle a related problem. We performed zero-shot trajectory prediction and reconstruction experiments using our model. The experiment aims at testing whether our model has captured the universal mobility pattern through learning the generation process of a trajectory. And the result is also expected to verify that it is possible to achieve trajectory general intelligence through modeling the universal human mobility pattern in a trajectory dataset. To the best of our knowledge, there are few trajectory modeling methods that are able to conduct zero-shot inference on multiple tasks.

Trajectory Prediction

Trajectory prediction is one of the most important tasks in trajectory data mining(Zheng, 2015). It requires a model to capture the mobility pattern between the observed trajectory and the predicted one. We employed a finished trained trajectory generation network. To achieve that, instead of putting the latent representation of the entire trajectory to the trajectory generator, the input to the generation network is changed as the concatenation of the location encoding the observed part of the trajectory and the latent representation of the trajectory point to be predicted. The prediction process can be formulated as follows:

$$X_T = sampler[N(0, \mathbf{I})]$$
$$X_t = \mathcal{G}(\mathcal{P}(loc_1), \mathcal{P}(loc_2), \ldots, \mathcal{P}(loc_{n-l}), X_{t+1})[-l]$$
$$Y = \mathcal{D}(X_0)$$

Where $X_t \in R^{l\times d}$, l is the length of trajectory points to be predicted, and d is the dimension of location encoding. $sampler$ is a controllable sampler used to sample from the latent normal distribution N. In trajectory prediction, the sampler sample at 0. This will be further explained in the latent space interpretation section. Considering the generation model was not trained for the prediction task previously, it is believed to be a zero-shot prediction. The trajectory generator was only trained for generating a trajectory sequence according to the spatial-temporal relationship in it. Therefore, by providing the generator with the exact information of the observed part of the trajectory, the model is expected to generate the rest part according to that.

Moreover, we employed an improved sampling method named Denoising Diffusion Implicit Models (DDIM) to accelerate generating by reducing the sampling process to 50 steps(Song et al., 2020). We compared the prediction accuracy with the DeepMove(Feng et al., 2018), which is the benchmark deep learning human mobility prediction method. To make a fair comparison, we abandoned the usage of other features like users' id. We compared the prediction accuracy in different prediction lengths. The comparison result is shown in table 2.

Table 2. Comparison of the prediction accuracy in different predicting lengths between Deepmove and the zero-shot prediction result of TrajGDM

Methods	T-Drive			Geo-life		
	Accuracy @Length 1	Accuracy @Length 2	Accuracy @Length 3	Accuracy @Length 1	Accuracy @Length 2	Accuracy @Length 3
DeepMove	34.86%	16.14%	10.11%	43.75%	25.26%	17.41%
TrajGDM	43.03%	21.13%	14.66%	34.11%	19.98%	14.63%

As it has shown in table 2, the zero-shot prediction accuracy of our model surpasses the accuracy of the DeepMove in T-Drive dataset for 23.4%, and the accuracy is only slightly lower than that of the DeepMove in the Geo-life dataset. This can be explained by the difference of learning objective between two models, DeepMove is trained to maximize the likelihood of $P(loc_{n-l:n}|loc_{1:n-l})$, while the objective of our generative model is to maximize $P(loc_{1:n-l}, loc_{n-l:n})$. The difference can be regarded as that DeepMove tends to output where the moving object most likely to go when it has passed by $loc_{n-l:n}$, whereas our model was trained to output all suitable positions $loc_{1:n-l}$ that make the complemented trajectory $(loc_{1:n-l}, loc_{n-l:n})$ most like a real one. With the predicting length l gets longer, the observed $loc_{1:n-l}$ plays a less important role, the difference between two distributions is getting smaller. This explains why two prediction accuracies are getting closer with the growth of prediction length l. To better understand how a trajectory is generated, we visualized the predicting process in figure 6.

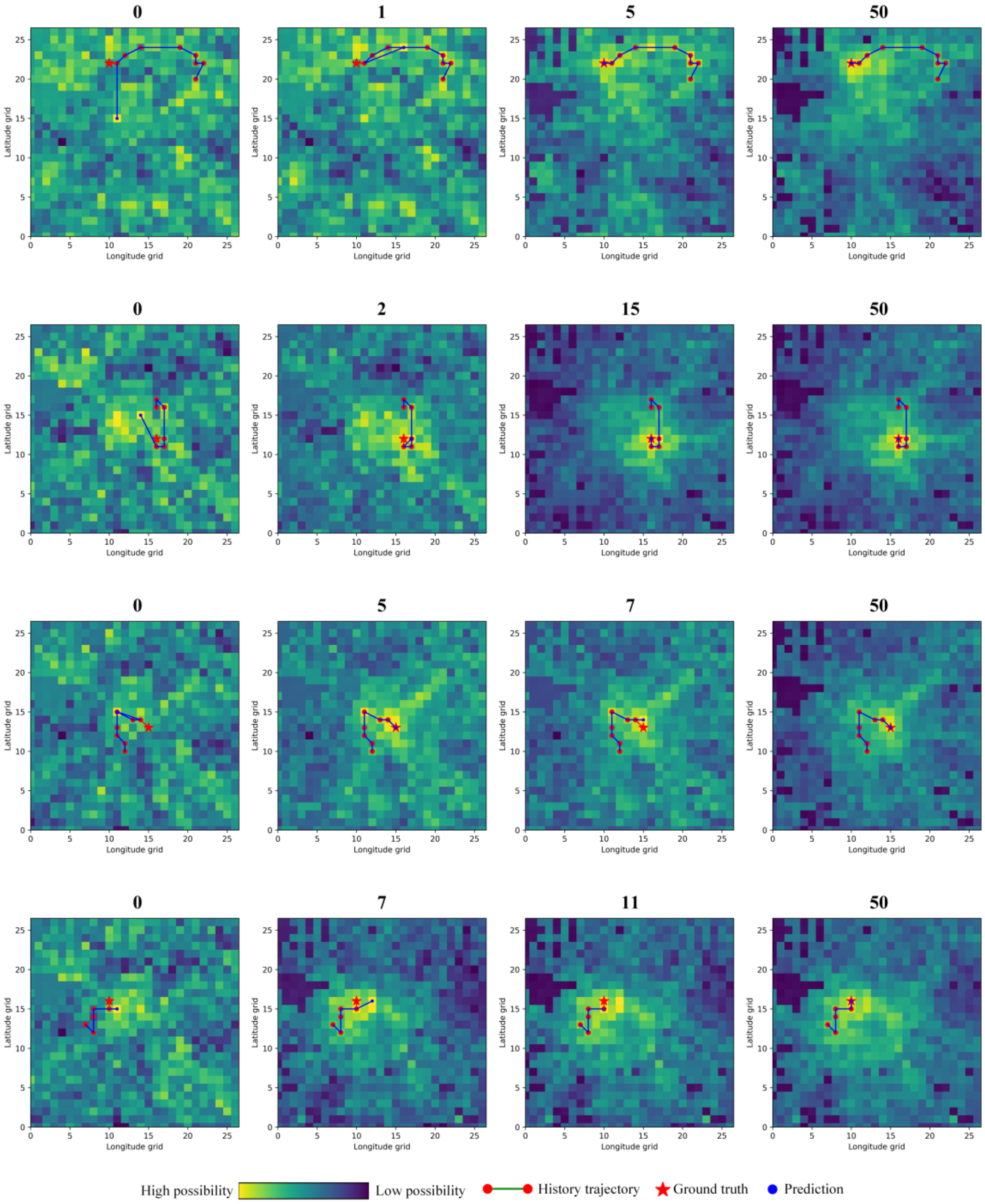


Figure 6. Trajectories in different prediction steps of the generation process of our method

Figure 6 shows the uncertainty reducing process in prediction. The TrajGDM gradually reduced the high possibility space. Taking the trajectory in the first column for example, in the beginning, the cab was predicted to move to the downtown in center of the area according to the general mobility pattern followed by most of the trajectory. Then, in the next step, the model adjusts its prediction result based on the historical trajectory and predicts it to go back to its former route. However, the output still seems unreasonable because the trajectory moves too far in a step. With the uncertainty gradually removed, the model generates later predictions mostly based on the current observation and predicts to move along its way. It also shows the spatial awareness that only nearby locations were predicted to be the high possibility region. The similar pattern can be seen in all the generation processes: the output high possibility region transferred from a global high trajectory density region to one that is primarily based on individual movement characteristics.

Trajectory Reconstruction

Trajectory reconstruction or imputation is another tough problem in human mobility modeling. In reality, limited by the cost of data transmission and storage space, most of the human mobility datasets are collected in a discontinuity form, and the data missing problem is very common(Wang et al., 2022; Zhao et al., 2016). Trajectory reconstruction is a key preprocessing step in many trajectory tasks. Accurate reconstruction requires a model to capture the mobility pattern from both prior and subsequent trajectory points of the missing data. We also employed a trained trajectory TrajGDM model to conduct a zero-shot trajectory reconstruction experiment. The reconstruction process can be formulated as following:

$$X_T = sampler[N(0, \mathbf{I})]$$

$$X_t = \mathcal{G}(\dots, \mathcal{P}(loc_{i-1}), X_{t+1}, \mathcal{P}(loc_{i+l}), \dots)[i: i + l]$$

$$Y = \mathcal{D}(X_0)$$

where $X_t \in R^{l \times d}$, l is the length of trajectory points to be reconstructed. i is the index for the first missing point of the trajectory. We randomly masked part of the points in a trajectory. And conduct the reconstruction follows the formula above. The reconstruction accuracy is compared with several baseline methods. MDP-TR combines multiple machine learning methods and shows great performance in reconstructing call detail record (CDR) trajectory data(Li et al., 2019). Spatial interpolation is one of the most common baseline methods for trajectory reconstructing, the method reconstructs a trajectory by joining each pair of existing points(Hoteit et al., 2014). The comparison result is listed in table 3. MDP-TR is unable to handle continuous missing points, so the corresponding position in the table is masked.

Table 3. Comparison of the reconstruction accuracy in different reconstruction lengths between baselines and the zero-shot reconstruction result of TrajGDM

Methods	T-Drive		Geo-life	
	Accuracy @Length 1	Accuracy @Length 2	Accuracy @Length 1	Accuracy @Length 2
MDP-TR	69.10%	----	37.20%	----
Interpolation	50.27%	17.83%	43.82%	19.77%
TrajGDM	59.12%	25.16%	46.75%	23.67%

In table 3, our method achieves the best reconstruction performance in the Geo-life dataset, where it surpasses both the mechanism method and the learning method. The improvement in accuracy is up to 25.6%. Moreover, another advantage of our model is that, our model is more flexible, it not only can reconstruct continuous missing points, but also can reconstruct a trajectory with multiple discrete missing points at one time.

The zero-shot predicting and reconstructing ability proves that the human mobility pattern captured by our model is universal. And learning human mobility from trajectory generation and applying it to do other trajectory tasks is a possible way to achieve artificial general intelligence in trajectory.

The latent space Interpretation

Understanding the relationship between the latent space and the real-world space can help us interpret how the model learns human mobility. In this section, we try to explore the relationship by controlling the $sampler$ in the sampling process of trajectory prediction. The basic setting is the same as the previous section. It is still a zero-shot prediction based on a trained model. We changed the latent representation of the points to be predicted with different sampling points in the latent space.

The TrajGDM models the human mobility with a Gaussian distribution, which means, in prediction, all appropriate positions x_i that make the trajectory $(x_{1:i-1}, x_i)$ like a real one are assumed to follow a normal distribution in the latent space. At the same time,all x_i candidates are mapped into the real space by the model from this normal distribution. Therefore, by sampling from different places in the latent space and analyzing their generation result, we could get a brief interpretation of how does the model build the connection between the latent space and the real-world space.

We sampled from 5 different places in the latent space $N(u, \sigma^2)$, where μ denotes the mean of the Gaussian distribution and σ is its standard deviation. Point μ owns the highest possibility density in the entire latent space. While the possibility density decreases as the sampled points move away from μ. We conducted an experiment by sampling from different positions in the latent space, and compared the prediction accuracy, possibility distribution and distance from the predicted point to the last point of the observed trajectory. The comparison of prediction accuracy and output distance is shown in table 4.

Table 4. Comparison of prediction accuracy and predicted distance between the prediction results sampled from points with different possibility density values and positions in the latent space

	T-Drive				
Sampled points	-2σ	$-\sigma$	μ	σ	2σ
Possibility density	0.0540	0.2420	0.3989	0.2420	0.0540
Accuracy	31.25%	37.55%	43.03%	40.42%	36.52%
Distance	3.83	1.61	0.88	1.92	4.61

In table 4, the prediction accuracy is the highest at point μ, and decreases with the possibility density is reduced in other sampled points. This means the high possibility density region in the latent space is mapped to the trajectory points with the highest emerge frequency by the trajectory generator. More specifically, when we sampled at point μ, where owns the highest possibility density in the latent space, the generated trajectory point also appears most frequently in the circumstance that $x_{1:i-1}$ is observed. While for those sampled points with lower possibility density, their corresponding generated trajectory point has a lower emerging possibility. Therefore, the prediction accuracy is relatively low when sampled at these regions. Spatially, this correlation was also reflected by the distance metric. In reality, the average moving distance between every two trajectory points is 0.96 grid. Obviously, the moving distance predicted by sampling at μ is the closest to the real average moving distance. With other sample points showing a much larger moving distance, their generated points are positioned in locations where it is less likely to move under condition $x_{1:i-1}$. It is worth noting that, even though the emerging frequency is relatively low at these low possibility density places, they are still sampled in the range of the latent distribution, and the outputs are generated under the guidance of the condition $x_{1:i-1}$. Therefore, their emergence should still be reasonable under condition $x_{1:i-1}$ in some extent. In order to comprehend the relationship between the latent space and real-world space, with different sampled points, we visualized the possibility distribution in geography, the result of several typical trajectories is shown in figure 7.

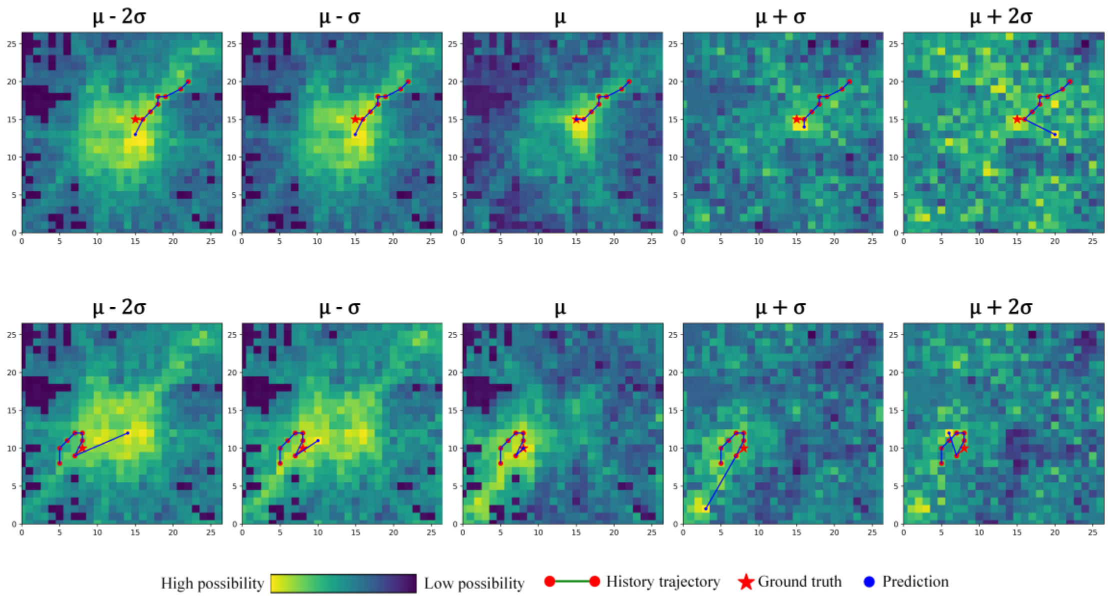


Figure 7. Predicted possibility distribution when sampled from different points in the latent space.

In figure 7, by comparing the variation of high possibility region generated by sampling from different sample points, it is found out that sampling at point μ generates the most suitable possibility distribution for the current trajectory. As we have mentioned previously, points with high possibility density values in the latent space correspond to places with high emergence frequency in reality. The high possibility region generated by sampling at μ contains most of the common locations that the trajectory might move to under current observation. Moreover, when sampled points are smaller than μ, the high possibility region covers a wider area. Interestingly, it is found out that the distribution is overlapped with the global high point density region of the whole dataset. While sampling at points greater than μ, locations with high possibility are discrete from each other, and they are more likely to be the high possibility locations that are only suitable with the current trajectory. In other words, it generates more personal predictions. With the sampled point leaving μ further, the tendency is getting more obvious. We believe this can be explained from a view of balancing the general and individual characteristics of human mobility. When modeling human mobility, most of the time, trajectories follow a general mobility pattern. While every individual trajectory also contains a unique mobility feature. The trajectory generation model is looking for a way to balance the general pattern and the personal feature. It maps two opposite human mobility characteristics into opposite directions of the latent space. The balance is achieved when we randomly sampled in the latent space.

Conclusions

In this research, we proposed a generative human mobility simulation method named TrajGDM. The method models the trajectory generation as an uncertainty decreasing process. We proposed a trajectory generator neural network, which aims to predict the existing uncertainty in a trajectory, and the trajectory generation process is defined as a process that the uncertainty of a trajectory is removed by the generator step by step. To achieve that, we defined a trajectory diffusion process to model the uncertainty adding process in a trajectory, so that our trajectory generator could be trained by learning from the relationship between the original trajectory and the uncertainty added trajectory after the diffusion process. Based on the diffusion process and generation process, we introduced our training method, which was designed to train our trajectory generator to learn from the trajectory dataset recorded in a discrete representation method. At last, we introduced the sampling process for our trajectory generation model. The model may generate a synthesized trajectory dataset that is promised to be diverse and realistic through the sampling process.

By comparing the performance of our method with five strong baselines in two datasets, our model achieves great improvement in simulating the individual mobility and the diversity of generated trajectories. As one of few real generative models, the performance of our model surpassed the commonly used discriminative models, offering additional potential for the trajectory generation task. Moreover, by visualizing the trajectory generation process, we found out that the uncertainty reducing process is similar to the route planning process of a human being, which shows the model has learned the generation correctly.

We conducted zero-shot experiments on two basic trajectory tasks, trajectory prediction, and reconstruction. The zero-shot performance of our model was close to the benchmark method in trajectory prediction and surpassed all baseline methods in trajectory reconstruction. The ability to solve multiple trajectory tasks is contributed to the universal mobility pattern captured by the model through learning the generation process of a trajectory. It also demonstrates that building a trajectory foundation model with a trajectory generation model is feasible. To the best of our knowledge, there is no such work that achieves zero-shot inference in any trajectory task.

At last, to understand how does our model learn human mobility. We interpreted the relationship between the latent space and the real-world space by controlling the sampler sampling at different points in the latent space. Surprisingly, it was found out that the latent space was not mapped symmetrically. The model was found to seek a balance between the crowd's mobility pattern and individual mobility pattern.

To sum up, in this work, we proposed a novel human mobility modeling method, which is able to generate high-quality trajectories. The proposed method also showed remarkable performance in capturing the universal mobility pattern and solving multiple trajectory tasks based on that, which is believed to be an important signature of a trajectory foundation model.

However, there still are a few deficiencies. The influence of time was not taken into consideration in the model. We have to admit that time has a significant influence on human mobility. At the current stage, we put more attention on modeling the moving pattern of a trajectory. So, we only modeled and generated human mobility in one hour, which is a relative short term. We believe that the influence of time in such a short term is minimized. In our further work, we will further explore time's influence on human mobility by modeling the trajectory with a wider time span.

Data and codes availability statement

The trajectory data and codes used in the experiment is available at: https://figshare.com/s/ad9dc3e775e891ab655d.

References

Bao, Y., Huang, Z., Li, L., Wang, Y., & Liu, Y. (2021). A BiLSTM-CNN model for predicting users' next locations based on geotagged social media. *International Journal of Geographical Information Science*, *35*(4), 639-660. https://doi.org/10.1080/13658816.2020.1808896

Chen, X., Xu, J., Zhou, R., Chen, W., Fang, J., & Liu, C. (2021). TrajVAE: A Variational AutoEncoder model for trajectory generation. *Neurocomputing*, *428*, 332-339. https://doi.org/https://doi.org/10.1016/j.neucom.2020.03.120

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, *34*, 8780-8794.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018). *DeepMove: Predicting Human mobility with Attentional Recurrent Networks* Proceedings of the 2018 World Wide Web Conference, Lyon, France. https://doi.org/10.1145/3178876.3186058

Feng, J., Yang, Z., Xu, F., Yu, H., Wang, M., & Li, Y. (2020). *Learning to Simulate Human mobility* Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA. https://doi.org/10.1145/3394486.3412862

Gong, S., Li, M., Feng, J., Wu, Z., & Kong, L. (2022). Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Commun. ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gu, T., Chen, G., Li, J., Lin, C., Rao, Y., Zhou, J., & Lu, J. (2022). Stochastic trajectory prediction via motion indeterminacy diffusion. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Hoteit, S., Secci, S., Sobolevsky, S., Ratti, C., & Pujolle, G. (2014). Estimating human trajectories and hotspots through mobile phone data. *Computer Networks*, *64*, 296-307. https://doi.org/https://doi.org/10.1016/j.comnet.2014.02.011

Huang, D., Song, X., Fan, Z., Jiang, R., Shibasaki, R., Zhang, Y., Wang, H., & Kato, Y. (2019, 28-30 March 2019). A Variational Autoencoder Based Generative Model of Urban Human mobility. 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR),

Isaacman, S., Becker, R., Cáceres, R., Martonosi, M., Rowland, J., Varshavsky, A., & Willinger, W. (2012). *Human mobility modeling at metropolitan scales* Proceedings of the 10th international conference on Mobile systems, tasks, and services, Low Wood Bay, Lake District, UK. https://doi.org/10.1145/2307636.2307659

Ji, Y., Gao, S., Huynh, T., Scheele, C., Triveri, J., Kruse, J., Bennett, C., & Wen, Y. (2023). Rethinking the regularity in mobility patterns of personal vehicle drivers: A multi-city comparison using a feature engineering approach. *Transactions in GIS*, *27*(3), 663-685. https://doi.org/https://doi.org/10.1111/tgis.13043

Jiang, S., Yang, Y., Gupta, S., Veneziano, D., Athavale, S., & González, M. C. (2016). The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National*

Academy of Sciences, *113*(37), E5370-E5378. https://doi.org/doi:10.1073/pnas.1524261113

Jiang, W., Wayne, Wang, J., & Jiang, J. (2023). Continuous Trajectory Generation Based on Two-Stage GAN. *arXiv pre-print server*. https://doi.org/None

arxiv:2301.07103

Jiao, R., Liu, X., Zheng, B., Liang, D., & Zhu, Q. (2022, 23-27 Oct. 2022). TAE: A Semi-supervised Controllable Behavior-aware Trajectory Generator and Predictor. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),

Karnewar, A., & Wang, O. (2020). Msg-gan: Multi-scale gradients for generative adversarial networks. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kodali, N., Abernethy, J., Hays, J., & Kira, Z. (2017). On Convergence and Stability of GANs. *arXiv pre-print server*. https://doi.org/None

arxiv:1705.07215

Li, L., Pagnucco, M., & Song, Y. (2022). Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Li, M., Gao, S., Lu, F., Liu, K., Zhang, H., & Tu, W. (2021). Prediction of human activity intensity using the interactions in physical and social spaces through graph convolutional networks. *International Journal of Geographical Information Science*, *35*(12), 2489-2516. https://doi.org/10.1080/13658816.2021.1912347

Li, M., Gao, S., Lu, F., & Zhang, H. (2019). Reconstruction of human movement trajectories from large-scale low-frequency mobile phone data. *Computers, Environment and Urban Systems*, *77*, 101346. https://doi.org/https://doi.org/10.1016/j.compenvurbsys.2019.101346

Li, M., Lu, F., Zhang, H., & Chen, J. (2020). Predicting future locations of moving objects with deep fuzzy-LSTM networks. *Transportmetrica A: Transport Science*, *16*(1), 119-136. https://doi.org/10.1080/23249935.2018.1552334

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., & Hashimoto, T. B. (2022). Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, *35*, 4328-4343.

Liu, Z., Ma, T., Du, Y., Pei, T., Yi, J., & Peng, H. (2018). Mapping hourly dynamics of urban population using trajectories reconstructed from mobile phone records [https://doi.org/10.1111/tgis.12323]. *Transactions in GIS*, *22*(2), 494-513. https://doi.org/https://doi.org/10.1111/tgis.12323

Liu, Z., Shi, W., Yu, Y., Chen, P., & Yu Chen, B. (2022). A LSTM-based approach for modeling the movement uncertainty of indoor trajectories with mobile sensing data. *International Journal of Applied Earth Observation and Geoinformation*, *108*, 102758. https://doi.org/https://doi.org/10.1016/j.jag.2022.102758

Mai, G., Janowicz, K., Hu, Y., Gao, S., Yan, B., Zhu, R., Cai, L., & Lao, N. (2022). A review of location encoding for GeoAI: methods and tasks. *International Journal of Geographical Information Science*, *36*(4), 639-673. https://doi.org/10.1080/13658816.2021.2004602

Mai, G., Janowicz, K., Yan, B., Zhu, R., Cai, L., & Lao, N. (2020). Multi-scale representation learning for spatial feature distributions using grid cells. *arXiv preprint arXiv:2003.00824*.

Mai, G., Xuan, Y., Zuo, W., Janowicz, K., & Lao, N. (2022). Sphere2Vec: Multi-Scale Representation Learning over a Spherical Surface for Geospatial Predictions. *arXiv pre-print server*. https://doi.org/None

arxiv:2201.10489

Mao, W., Xu, C., Zhu, Q., Chen, S., & Wang, Y. (2023). Leapfrog Diffusion Model for Stochastic Trajectory Prediction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Ouyang, K., Shokri, R., Rosenblum, D. S., & Yang, W. (2018). A Non-Parametric Generative Model for Human Trajectories. International Joint Conference on Artificial Intelligence,

Qi, M., Qin, J., Wu, Y., & Yang, Y. (2020). Imitative non-autoregressive modeling for trajectory forecasting and imputation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Rao, J., Gao, S., Kang, Y., & Huang, Q. (2020). LSTM-TrajGAN: A Deep Learning Approach to Trajectory Privacy Protection. *arXiv pre-print server*. https://doi.org/None

arxiv:2006.10521

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Salzmann, T., Ivanovic, B., Chakravarty, P., & Pavone, M. (2020, 2020//). Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. Computer Vision – ECCV 2020, Cham.

Simini, F., Barlacchi, G., Luca, M., & Pappalardo, L. (2021). A Deep Gravity model for mobility flows generation. *Nature Communications*, *12*(1), 6576. https://doi.org/10.1038/s41467-021-26752-4

Song, C., Qu, Z., Blumm, N., & Barabási, A.-L. (2010). Limits of Predictability in Human mobility. *Science*, *327*(5968), 1018-1021. https://doi.org/doi:10.1126/science.1177170

Song, J., Meng, C., & Ermon, S. (2020). Denoising Diffusion Implicit Models. *arXiv pre-print server*. https://doi.org/None

arxiv:2010.02502

Song, X., Kanasugi, H., & Shibasaki, R. (2016). *Deeptransport: prediction and simulation of human mobility and transportation mode at a citywide level* Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, New York, USA.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

Tashiro, Y., Song, J., Song, Y., & Ermon, S. (2021). CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, *34*, 24804-24816.

Theis, L., A\"aron, & Bethge, M. (2016). A note on the evaluation of generative models. *arXiv pre-print server*. https://doi.org/None

arxiv:1511.01844

Wang, P., Wang, H., Zhang, H., Lu, F., & Wu, S. (2019). A Hybrid Markov and LSTM Model for Indoor Location Prediction. *IEEE Access*, *7*, 185928-185940. https://doi.org/10.1109/ACCESS.2019.2961559

Wang, P., Zhang, T., Zheng, Y., & Hu, T. (2022). A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation. *International Journal of Geographical Information Science*, *36*(6), 1231-1257. https://doi.org/10.1080/13658816.2022.2032081

Yin, M., Sheehan, M., Feygin, S., Paiement, J. F., & Pozdnoukhov, A. (2018). A Generative Model of Urban Activities from Cellular Data. *IEEE Transactions on Intelligent Transportation Systems*, *19*(6), 1682-1696. https://doi.org/10.1109/TITS.2017.2695438

Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1). https://doi.org/10.1609/aaai.v31i1.10804

Yu, Y., Yao, Y., Liu, Z., An, Z., Chen, B., Chen, L., & Chen, R. (2023). A Bi-LSTM approach for modeling movement uncertainty of crowdsourced human trajectories under complex urban environments. *International Journal of Applied Earth Observation and Geoinformation*, *122*, 103412. https://doi.org/https://doi.org/10.1016/j.jag.2023.103412

Zhao, Z., Shaw, S.-L., Xu, Y., Lu, F., Chen, J., & Yin, L. (2016). Understanding the bias of call detail records in human mobility research. *International Journal of Geographical Information Science*, *30*(9), 1738-1762. https://doi.org/10.1080/13658816.2015.1137298

Zheng, Y. (2015). Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.*, *6*(3), Article 29. https://doi.org/10.1145/2743025