

RESEARCH ARTICLE

Simulating human mobility with a trajectory generation framework based on diffusion model

Chen Chu^{a,b}, Hengcai Zhang^{a,b}, Peixiao Wang^{a,b} and Feng Lu^{a,b}

^aState Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, China; ^bUniversity of Chinese Academy of Sciences, Beijing, China

ABSTRACT

Most mobility modeling methods are designed to solve specific tasks, leading to questions regarding their deficiency in generalizability. Inspired by the bloom of foundation models, we proposed a Trajectory Generation framework based on the 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, in which a deep learning network with a novel training method is proposed to learn from the process. We compared the proposed trajectory generation method with six baselines on two public trajectory datasets. The results showed that the similarity between the generated and real trajectory movements measured by the Jensen-Shannon Divergence improved significantly on both datasets. Moreover, we applied zero-shot inferences on two basic trajectory tasks: trajectory prediction and trajectory reconstruction. The accuracy improved by a maximum of 25.6% on two tasks. The universal mobility pattern that is suitable for solving multiple trajectory tasks is verified, inferring the strong generalizability of our model. Finally, the study provides insights into artificial intelligence's understanding of human mobility by exploring the way the model maps the trajectory in the latent space into reality.

ARTICLE HISTORY

Received 25 July 2023
Accepted 25 January 2024

KEYWORDS

Human mobility; trajectory generation; diffusion model; geo-foundation model

1. Introduction

Human mobility simulation plays a central role in a wide range of applications, 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. Currently, the most common approach for solving different human mobility tasks is to build multiple corresponding models with each model learn a part of the movement pattern. For example, a trajectory prediction model aims to capture the mobility pattern between the observed trajectory and the predicted point. However, it cannot reconstruct the missing points in a trajectory because of the limitation of its learning

objective. The generalization ability of a model is restricted by the specific mobility pattern the model has learned. Therefore, learning the universal mobility pattern in a trajectory dataset is a necessary step towards achieving Artificial General Intelligence for human mobility. In this study, we defined the universal mobility pattern in a trajectory dataset as the movement pattern followed by all trajectory points of any trajectory in the dataset (Noulas *et al.* 2012, Schlapfer *et al.* 2021). Models succeed in learning the universal pattern should be able to solve multiple relevant trajectory tasks.

Simulating the generation process of trajectories is one way to learn the universal mobility pattern from the dataset. Mobility simulation, also known as trajectory generation, aims to generate a synthesized mobility dataset based on the mobility pattern shown in the real one (Jiao *et al.* 2022). Classic mobility simulation methods simulate human mobility using mechanistic methods (Isaacman *et al.* 2012, Jiang *et al.* 2016, Simini *et al.* 2021). These studies model human mobility based on researchers' prior knowledge of human mobility patterns and assume that every movement of individuals has an explicit purpose. The strong assumptions limit the ability of the models to generate detailed trajectories. Moreover, most mechanistic models employ external location features, such as land use, points of interest, etc. (Yin *et al.* 2018). The introduction of external features also constrains the transfer ability of mechanistic models.

With deep generative models achieving great success in many generation tasks, models such as the Variational Autoencoder (VAE) (Kingma and Welling 2013), Generative Adversarial Network (GAN) (Goodfellow *et al.* 2020), and Denoising Diffusion Probabilistic Model (DDPM) (Ho *et al.* 2020) have been used in human mobility simulation. The generative model raises the concept of latent space (Doersch 2016). In a generative model, the latent space refers to a feature space that composes all possible samples. A generative model learns to map a latent distribution to an actual data distribution. After learning this generation process, as the latent distribution is manually defined and continuous, the model can sample numerous vector representations and generate numerous different trajectories. Therefore, the existence of a latent space promises generation diversity, which is of vital importance for human mobility simulation because the movement activity of each human being is unique and inherently stochastic (Song *et al.* 2010).

However, most of current generative models have limitations in terms of trajectory generation. The VAE failed to generate qualified trajectories because the noise-adding operation was fuzzy for a trajectory with an explicit meaning (Chen *et al.* 2021). As the most well-known structure for deep generative models, GAN is notoriously difficult to train (Karnewar and Wang 2020, Kodali *et al.* 2017). Moreover, instead of sampling from the latent space to obtain the latent representation and then generating a trajectory according to the meaning of the vector, most GAN-based trajectory generation models abandon the use of latent space (Feng *et al.* 2020, Jiang *et al.* 2023, Yu *et al.* 2017). Their learning objective is no longer to learn how to map the latent distribution to the real distribution. Therefore, they lose the ability to generate diverse trajectories. In order to distinguish them, we only name models that aim at learning to bridge two distributions as 'real' generative models.

To learn the universal mobility pattern in a trajectory dataset, generate high-quality trajectories with great diversity and understand how artificial intelligence understands

the human mobility, we proposed a Trajectory Generation Framework based on the Diffusion Model (TrajGDM), which aims to learn the universal human mobility pattern by simulating the generation process of trajectories. Inspired by the natural uncertainty of human mobility (Liu *et al.* 2022, Yu *et al.* 2023), we modeled the generation process of a trajectory as a process in which the uncertainty in the trajectory was gradually removed. The basic concept of the proposed model is illustrated in Figure 1.

As shown in Figure 1, the trajectory generation process is modeled as an uncertainty-reducing process. X_t denotes the trajectory representation at step t of the trajectory generation process p , and diffusion process q . We employed a deep learning network with parameter h to estimate the uncertainty in X_t based on X_{t-1} . A detailed trajectory is generated after T steps of the uncertainty reduction. To generate a realistic trajectory, the model is trained to capture the universal mobility pattern in the dataset through learning to estimate the uncertainty in a trajectory, which trains the model to generate a trajectory from a latent representation X_T . The universal mobility pattern is learned by the entire trajectory generation process, which includes network with parameter h and other fixed parameters in p and q . After the model learned the universal pattern through the process, we conducted zero-shot trajectory prediction and reconstruction to evaluate the generalization ability of the model. Our contributions can be summarized as follows:

We proposed a novel human mobility simulation method named TrajGDM, which models trajectory generation as an uncertainty-reducing process. A trajectory encoder, decoder and generator network was proposed to learn the universal mobility pattern from the generation process of trajectories. Moreover, we define the trajectory diffusion process and trajectory generation process with a novel training method to train the model and generate trajectories in discrete representation.

We compared the trajectory generation performance of our method with those of six strong baselines using two datasets. Our model achieves significant improvement in simulating individual mobility and other metrics while promising the diversity of generated trajectories. Furthermore, by visualizing the trajectory generation process and interpreting the latent space of the model, a new perspective on artificial intelligence's understanding of human mobility was 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 by learning the trajectory generation process. This demonstrates the strong generalizability of our model and its potential of serving as a foundation model in human mobility modeling.

Figure 1. The intuition of TrajGDM.

The remainder of this paper is organized as follows. [Section 2](#) introduces related studies in modeling human mobility. In [Section 3](#), we define the trajectory, trajectory dataset, and the target of trajectory generation, prediction and reconstruction. In [Section 4](#), the TrajGDM framework is proposed. In [Section 5](#), we provide detailed information regarding the experiment datasets and model implementation. We introduce six baselines used for comparison. In [Section 6](#), we analyze the experimental results of trajectory generation, prediction, and reconstruction, and then interpret the latent space of the trajectory. Finally, we discuss and conclude the paper in [Sections 7, 8](#).

2. Related work

Simulating human mobility has always been the most challenging task in trajectory data mining. Although most human mobility models have been built to solve specific problems, such as human movement prediction (Bao *et al.* 2021) and trajectory reconstruction (Li *et al.* 2019, Liu *et al.* 2018, Qi *et al.* 2020), they have failed to capture the universal mobility pattern applied to the entire trajectory. Only a few studies have attempted to solve this problem based on a general structure (Musleh 2022). In this study, we proposed a new human mobility simulation model based on a generative model. The model could simultaneously generate, predict, and reconstruct trajectories.

Most existing methods learn human mobility patterns by using sequential models. Recurrent neural networks and attention mechanisms are often used to model the temporal relationships within a trajectory (Bao *et al.* 2021, Feng *et al.* 2020, Li *et al.* 2022a). Regarding the spatial relationships between locations, a classic embedding function is employed in most models (Kang *et al.* 2017, Feng *et al.* 2018). Because it is well-adapted to all task scenarios. Furthermore, spatial information must be provided to the models as prior knowledge to achieve better performance. Graph structures are often used (Salzmann *et al.* 2020). Location encoding is another way to represent spatial relationships (Mai *et al.* 2022). Representing a location using its neighbor fuzzy locations (Li *et al.* 2020) or a Markov transition matrix (Wang *et al.* 2019) can be regarded as a type of fixed-location encoding method. In contrast, methods such as Space2Vec (Mai *et al.* 2020) and Sphere2Vec (Mai *et al.* 2023) focus on preserving the distance relationship between locations while making the vectors trainable based on different downstream tasks. In this study, we introduce a location encoding method that suits discrete location representation methods.

Current generation methods are primarily 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 remaining trajectory. A discriminator was designed to return rewards (Feng *et al.* 2020). TS-TrajGen is 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 parameters of the A* algorithm between the OD. It can then output a continuous trajectory using a path search algorithm (Jiang *et al.* 2023). However, these GAN-based trajectory-generating methods abandoned the use of latent space, which is important for a generative model to generate a diverse sample. Only a few trajectory-generation models have been designed to generate trajectories from latent vectors. TrajGAN converts trajectory points into pixels in a picture of an entire region and employs a standard CNN-based GAN structure to generate a trajectory (Ouyang *et al.* 2018). The SVAE combines the VAE and a sequence-to-sequence model and generates a trajectory by sampling from the designed latent space (Huang *et al.* 2019). The latest research Act2Loc combines machine learning and mechanistic models (Kang *et al.* 2023).

Recently, a novel generative model, the diffusion model, has bloomed in many realms, including image generation (Dhariwal and Nichol 2021, Ho *et al.* 2020), text generation (Gong *et al.* 2022, Li *et al.* 2022b), time-series modeling (Tashiro *et al.* 2021) and trajectory prediction (Gu *et al.* 2022, Mao *et al.* 2023), etc. The diffusion model aims to model the generation process of a sample step-by-step, starting from a latent representation sampled from the latent space. Currently, the vast majority of diffusion-based trajectory generation models aim to generate trajectories recorded as numerical coordinates (Gu *et al.* 2022, Mao *et al.* 2023). Because the coordinates are numerical, these methods can directly employ the original structure and training method of the classic diffusion model. However, the adaptability and applicability of these methods are limited. Many human mobility datasets are not recorded in the format of numerical coordinates, such as Call Detail Record data (CDR), in which locations are represented by location indexes. In contrast to other diffusion model-based trajectory generation methods, we proposed a novel training method that allows the model to generate trajectories recoded in location indexes to make the model more extensible and adaptive to more geographic datasets. Moreover, most of these studies focused only on generating trajectories at small scales, such as streets or courts (Gu *et al.* 2022, Mao *et al.* 2023), and rarely extended to generate trajectories for an entire city.

In this study, we proposed a trajectory generation model with a manually constructed latent space, the model can generate diverse and realistic trajectories. Moreover, we explore how the model learns to map the latent space to the real-world space. This is important to understand how neural networks learn the physical world.

3. Preliminary

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

Definition 1. Trajectory

Trajectory X is formed by a serial of trajectory points, which are recoded in location indexes loc : n denotes the trajectory length.

$$X = \{loc_1, loc_2, \dots, loc_n\} \quad (1)$$

Definition 2. Trajectory dataset

The number of trajectories that record the human mobility activity in a region forms the human mobility dataset D : Num denotes the number of trajectories recorded in the dataset.

$$D = \{X^1, X^2, \dots, X^{Num}\} \quad (2)$$

Definition 3. Trajectory latent space

The latent space of a trajectory dataset is a high-dimension feature space Z , where each trajectory in dataset D has a corresponding representation. In general, we must define a mapping function M to map all trajectories in D to their corresponding locations in Z : The function can be defined as

$$Z_D = \{M(D)\} \quad (3)$$

where Z_D denotes the latent representation of dataset D :

Definition 4. Trajectory generation

The objective of trajectory generation is to generate a realistic dataset $\hat{D} = \{\hat{X}^1, \hat{X}^2, \dots, \hat{X}^{Num}\}$, by sampling the trajectories from latent space Z_D : This can be formulated as follows:

$$\hat{D} = F_h(Z_D) \quad (4)$$

where the trajectory generation model F learns the mobility pattern in the trajectory dataset D by its trainable parameter h : It then generates a synthesized dataset \hat{D} , which is expected to similar to the real dataset D from all aspects. The basic measurement formula is as follows:

$$\min_h JSD(D, \hat{D}) \quad (5)$$

JSD is the Jensen-Shannon Divergence (JSD), which is generally used to measure the difference between two distributions. M can be a different metric used to measure the characteristics of trajectories in a dataset. This similarity can be evaluated from many aspects, such as the similarity of individual movements or geographical distributions of all trajectory points. We introduce the evaluation method more specifically in the evaluation metrics section.

Definition 5. Trajectory prediction

Trajectory prediction aims to predict future movements of moving objects. This is defined as follows:

$$\{\overline{loc}_{n-l}, \dots, \overline{loc}_n\} = Pre(\{loc_1, \dots, loc_{n-l}\}) \quad (6)$$

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

Definition 6. Trajectory reconstruction

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

$$\overline{u}_1, \dots, \overline{u}_{i-1}, \overline{u}_i, \dots, \overline{u}_M, \text{loc}_{i-1}, u_i, \dots, u_M, \text{loc}_{i+1}, \dots, \text{loc}_M \quad (7)$$

where u_i denotes the missing trajectory point i and \overline{u}_i denotes the corresponding trajectory points reconstructed by model *Rec*; l denotes the length of the missing trajectory points to be reconstructed.

4. Trajectory generation framework

We modeled the generation of a trajectory as a process in which the uncertainty in the trajectory was gradually removed. A trajectory diffusion process was constructed to simulate the uncertainty-adding process to train a model with the ability to predict and remove uncertainty in a trajectory. The general architecture of the TrajGDM framework is shown in Figure 2.

In Figure 2, there are two important parts of the framework: the diffusion process and the generation process. The generation process assumes that the generation of a trajectory is an uncertainty-removing process. More specifically, in the initial stages of human mobility, only a few essential factors are formed by a person, such as its travel purpose, destination and origin. These factors were represented by a latent vector. At this stage, the entire trajectory is full of uncertainty, which means that the detailed route that satisfies all key factors is unknown. Based on these key factors, the person gradually decides on each movement. We designed a deep learning network, called a trajectory generator, to estimate the uncertainty in trajectory X_t , the network is

Figure 2. Structure of the TrajGDM framework.

denoted as G : The trajectory generator predicts and eliminates the uncertainty in the current trajectory. Through T iterations in the trajectory generation process, the uncertainty in the trajectory is removed, and a detailed trajectory corresponding to the key factors in the representation is generated. To train trajectory generator G , we constructed 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 added in a trajectory. The latent representation of all the trajectories in the dataset follows a normal distribution after T steps. During the diffusion process, we made our trajectory generator learn to estimate the added noise and recover the original real trajectory X_0 :

In the remainder of this section, we first introduce how we encode a trajectory into a hidden representation using the location-encoding function P and trajectory encoder E : We then introduce specific definitions of the trajectory diffusion and trajectory generation processes. In addition, we introduce our training process, which combines diffusion and generation to train the trajectory generator. Subsequently, a sampling process is proposed to generate trajectory data by sampling the latent vectors from the latent space. Finally, the structure of the trajectory decoder D and trajectory generator network G are presented.

4.1. Trajectory encoder

In this study, the model is designed to be applicable to large-scale trajectory datasets, most of which record a trajectory as a sequence of location indexes, such as indexes of CDR grids, road segments, blocks, towns, or Thiessen polygons. Therefore, the aim of the research is to generate trajectories recorded in location indexes, which are discretely represented. To encode discrete trajectory data representation into a continuous feature space X_0 , we proposed a trajectory encoder E . The formula of the trajectory encoding function E is as follows:

$$X_0 = E(\text{LSTM}(\text{Loc}_1, \text{Loc}_2, \dots, \text{Loc}_n)) \quad (8)$$

$$E(\text{LSTM}(\text{Loc}_1, \text{Loc}_2, \dots, \text{Loc}_n)) = \text{LSTM}(\text{Loc}_1, \text{Loc}_2, \dots, \text{Loc}_n) \quad (9)$$

where we encode the trajectory with an *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 the corresponding trajectory. We also proposed a trainable location-encoding function P to provide the model with an awareness of the spatial relationship between locations. The location-encoding function to map point *loc* is formulated as follows:

$$P(\text{loc}) = \text{Concat}(E(\text{loc}), \text{AdjLoc}_1, \dots, \text{AdjLoc}_c) \quad (10)$$

$$E(\text{loc}) = \text{Encoder}_{\text{onehot}}(\text{loc}) W_E \quad (11)$$

where W_P is the parameter of the location encoding function, *Concat* is the concatenate function, c is the hyperparameter of the location encoding function. *AdjLoc_i* is the adjacent query function, which returns *locⁱ* the i th adjacent location with the location *loc*: E is a general embedding function that embeds the locations into a feature space. *Encoder_{onehot}* is a one-hot sorting function that converts each location index *loc*

into a one-hot vector. $W_E \in \mathbb{R}^{l \times h}$ is the trainable weights of the function E , where l and h respectively represent the number of locations and the dimension of the embedded vector. Considering that the movement of a trajectory is continuous in real-world space, it is important to provide an adjacent relationship to the model so that the model learns from people's prior knowledge of the spatial relationship rather than starting to learn the relationship between locations from nothing. Therefore, instead of simply representing a location using its embedding vector, the proposed location-embedding function combines adjacent locations of the location loc : The function employs a hyperparameter c to emphasize the actual location loc and employs a trainable weight matrix W_P to learn complex spatial information from the adjacent relationship. After encoding each location in a trajectory into a feature space, the trajectory encoder E is employed to construct representation X_0 of the entire trajectory X : A trajectory encoder is vital for generating a continuous trajectory because it integrates the representation of points into the representation of a trajectory.

4.2. Trajectory diffusion process

We construct the trajectory diffusion process to gradually add uncertainty to a trajectory. To ensure that the trajectory can be reconstructed after adding uncertainty, we built a T steps Markov chain. At each step, 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 onto a latent representation X_T , that follows the latent distribution $q(\mathbf{X}_{1:T}|\mathbf{X}_0)$. The stepwise diffusion process from X_0 to X_T is formulated as follows:

$$q(\mathbf{X}_{1:T}|\mathbf{X}_0) = \prod_{t=1}^T q(\mathbf{X}_t|\mathbf{X}_{t-1}) \quad (12)$$

$$q(\mathbf{X}_t|\mathbf{X}_{t-1}) = \text{Normal}(\mathbf{X}_t; \mu_t, \sigma_t^2) \quad (13)$$

where $q(\mathbf{X}_t|\mathbf{X}_{t-1})$ is one step in diffusion process, $\text{Normal}(\mathbf{X}|\mu, \sigma^2)$ is a sample function, which samples a vector \mathbf{X} from a normal distribution with mean μ and variance σ^2 , scheduler $b_t \in [0, 1]$ 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 b_1 to b_T . To simplify the calculation, according to the notable property of the diffusion process, we can sample the trajectory \mathbf{X}_t at an arbitrary timestamp t in a closed form as follows:

$$q(\mathbf{X}_t|\mathbf{X}_0) = \text{Normal}(\mathbf{X}_t; \bar{\mu}_t, \bar{\sigma}_t^2) \quad (14)$$

where $a_t = 1 - b_t$ and $\bar{\mu}_t = \sum_{s=0}^{t-1} a_s \mu_s$. According to this formula, we can directly get the noised trajectory representation \mathbf{X}_t by sampling from an \mathbf{X}_0 -based Gaussian distribution. The formula for sampling is as follows:

$$\mathbf{X}_t = \bar{\mu}_t + \bar{\sigma}_t \epsilon \quad (15)$$

where ϵ is a random Gaussian noise, and it is regarded as the uncertainty added on the trajectory \mathbf{X}_0 . Considering $a_t < 1$, when the diffusion step t is large enough, the $\bar{\mu}_t$ will be sufficiently close to 0, which leads to the $q(\mathbf{X}_t|\mathbf{X}_0)$ converging to a standard

Gaussian distribution. This property ensures that all trajectory representations in the training dataset D are projected onto the same latent distribution. The good property of the normal distribution also ensures that we can easily sample from it in a controllable manner. This attribute is important for many downstream tasks in human mobility modeling.

4.3. Trajectory generation process

After the uncertainty was added, all trajectories were mapped into a latent distribution following a normal distribution. The trajectory generation process was 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 that adds a small bit of uncertainty to the trajectory at each step, the trajectory generator only needs to learn to estimate the uncertainty added at each step. The generation process was also built as a T steps process; therefore, the model only needed to estimate a small portion of the uncertainty in each step. And after T times of iterations, the model distribution $p_{h|l}(x_{0:T}|z)$ is formed. The generation process is formulated as follows:

$$p_{h|l}(x_{0:T}|z) = p_{h|l}(x_0|z) \prod_{t=1}^T p_{h|l}(x_t|x_{t-1}, z) \quad (16)$$

$$p_{h|l}(x_{t-1}|x_t, z) \sim \text{Normal}(x_{t-1}; l_{h|l}(x_t, t|z), \sigma_{h|l}^2(x_t, t|z)) \quad (17)$$

where $p_{h|l}(x_{0:T}|z)$ is the latent distribution of all trajectories in the training dataset, $N(l, \sigma^2)$ refers to a normal distribution with mean l and variance σ^2 ; $p_{h|l}(x_{t-1}|x_t, z)$ is the estimation of x_{t-1} given x_t by model G with the parameter h . The variance term of the Gaussian transition is manually set as $\sigma_{h|l}^2(x_t, t|z) = b_t(l_{h|l}(x_t, t|z), \frac{1-\alpha_{t-1}}{1-\alpha_t})$, which has been proved to have good performance in practice (Ho *et al.* 2020).

4.4. Training process

We defined the trajectory diffusion process to add uncertainty to a trajectory, and the trajectory generation process to remove it. For the next step, we need to know how to combine the two processes and define a training objective so that a trajectory generator can learn from the added noise in the diffusion process and use the learned pattern to generate a trajectory through the 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 $p_{h|l}(x_{0:T}|z)$ and the data distribution $q_{l|z}(x_{0:T}|z)$ reflected by real trajectory data. Therefore, our training objective was to maximize the variational lower bound as follows:

$$\max_h E_{q_{l|z}(x_{0:T}|z)} \log p_{h|l}(x_{0:T}|z) = \max_h E_{q_{l|z}(x_{0:T}|z)} \log p_{h|l}(x_{0:T}|z) - \log q_{l|z}(x_{0:T}|z) \quad (18)$$

E refers to the mathematical expectation of a variable under a given condition. Because we modeled the trajectory generation process as Gaussians

with trainable mean functions and fixed variances, the objective above can be simplified as follows:

$$\min_h E_{x_0, e} \frac{b_t^2}{2b_t a_t \bar{t} - a_t \bar{t}} \|e - G_h(p_{\bar{t}} \bar{X}_0, 1 - \bar{a}_t e, t)\|_2^2 \quad (19)$$

where $G_h(p_{\bar{t}} \bar{X}_0, 1 - \bar{a}_t e, t)$ is the uncertainty predicted by trajectory generator G , at generation step t to approximate e . The objective can be simply interpreted as minimizing the difference between the uncertainty e added to the observed trajectory X_0 in each step of the trajectory diffusion process with the uncertainty G_h estimated by the trajectory generator G :

However, unlike the color of the pixels in a picture, which is naturally distributed in a continuous feature space, the representation of geographic locations is generally 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 differences between the locations rather than their absolute positions in feature space. The original training loss of the diffusion model derives gradient from minimizing the difference between the estimation of the added noise e_h in each step with the truly add one e , which has been proved to be efficient in modeling the continuous feature space (Rombach *et al.* 2022). The discrete state space requires a more distinct training loss to direct the optimization of the relationship between locations in the feature space. Therefore, in addition to using location encoding to map the discrete location representation into a continuous feature space, we modified the training process such that the model could focus on learning the relationship of locations through the trajectory. Another difference with the classical latent diffusion model is that the trajectory encoder E , decoder D and generator G_h are trained simultaneously. Different trajectory datasets have totally different mobility patterns and geographic scales, so the relationship between locations is totally different, which makes it meaningless to pre-train a location encoder or decoder across different datasets. On the other hand, the current training method makes the model easy to converge, which makes the pretrained encoder and decoder become unnecessary. Specifically, we directly calculate the estimation of X_0 according to the $q_{\bar{t}}(X_t|X_0)$ formula for the trajectory diffusion process; the function can be formulated as follows:

$$\bar{X}_0 = \frac{1}{\bar{r}_t} \sum_{i=1}^{\bar{r}_t} G_h(X_t, \bar{t}_i) \quad (20)$$

We then decode the estimated \bar{X}_0 using a trajectory decoder D , so that the hidden representation is decoded to the probabilistic of each discrete location. We then used the gradient from a SoftMax Cross-Entropy loss to train the network; the function can be formulated as follows:

$$\min_h E_{X, e} N(\bar{0}, 1)_{\bar{t}} - X \log D(\bar{X}_0) \quad (21)$$

Therefore, our training algorithm can be summarized as [Algorithm 1](#).

Algorithm 1 Training algorithm

```

1: repeat:
2:    $X_t \sim q_{ij} X_{t-1} \in P_{ij} X_{t-1}$ 
3:    $t \leftarrow \text{Uniform}(12, \dots, T)_{ij}$ 
4:    $e \sim N(0, I)_{ij}$ 
5:   Take gradient from  $q_{ij} X_{t-1} \in P_{ij} X_{t-1}$ 
    $r_{hjj} = X - \log D \left( \frac{p_{hij} X_{t-1}}{q_{ij} X_{t-1}} \right) = \frac{1}{r_t} - 1 - G_{hij}(X_{t-1}, t)_{jj}$ 
6: until converged

```

4.5. Sampling process

As the requirement of the optimization function $\min_{\theta} E_{x_0, e, \dots} [I_{hij}(X_t, t)_{ij}]$ in the trajectory generation process has to predict under the following form:

$$I_{hij}(X_t, t)_{ij} = \frac{1}{a_t} \frac{p_{hij}(X_t)}{q_{ij}(X_t)} = \frac{b_t}{1 - a_t} \frac{p_{hij}(X_t)}{q_{ij}(X_t)} \quad (22)$$

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

$$X_{t-1} = \frac{1}{a_t} X_t - \frac{b_t}{1 - a_t} \frac{p_{hij}(X_t)}{q_{ij}(X_t)} z \quad (23)$$

where z denotes a random variable sampled from a standard Gaussian distribution. According to the formula, once all the parameterized models were trained through the training process, we can estimate $G_{hij}(X, t)_{ij}$ as the estimation of the uncertainty in X_t and generate a trajectory through the generation algorithm in [Algorithm 2](#).

Algorithm 2 Generating algorithm

```

1:  $X_T \sim N(0, I)_{ij}$ 
2: for  $t \in T, \dots, 1$  do
3:    $z \sim N(0, I)_{ij}$  if  $t > 1$ , else  $z = 0$ 
4:    $X_{t-1} = \frac{1}{a_t} X_t - \frac{b_t}{1 - a_t} \frac{p_{hij}(X_t)}{q_{ij}(X_t)} z$ 
5: end for
6:  $\hat{X} = D_{ij} X_0$ 
7: return  $\hat{X}$ 

```

4.6. Trajectory generator

We defined the diffusion and generation processes and proposed a training process to point out an optimization objective to train the trajectory generator and a sampling process to use the trajectory generator to generate a new trajectory. The key structure of the human mobility modeling framework is the trajectory generator network, which is proposed to generate a trajectory by reducing its uncertainty. We designed a

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

Transformer-based trajectory generation network to capture the spatial-temporal relationships in a trajectory. The structure of the trajectory generator is shown in Figure 3.

Trajectory generation has strict requirements for serial patterns in the generated trajectory. The generated trajectory must be continuous and must follow the mobility pattern in a real dataset. Therefore, we proposed a trajectory generator based on the sequence-to-sequence structure. As shown in Figure 3, the trajectory generator takes X_t as the input, which denotes the latent representation of trajectory X at generation step t . A step-encoding function SE is used to encode the current step t into the latent feature X_t . The step encoding function is a variation of the positional encoding function in the Transformer (Vaswani *et al.* 2017) and is formulated as follows:

$$SE(X_t, t) = \begin{cases} \sin \frac{t}{10000^{\frac{2i}{d}}}, & \text{if } i \% 2 = 0 \\ \cos \frac{t}{10000^{\frac{2i}{d}}}, & \text{if } i \% 2 \neq 0 \end{cases} \quad (24)$$

where d is the dimension of X_t and i is the dimension. Considering that the amount of added uncertainty in different steps of the diffusion process varies, step encoding injects step information into X_t such that the model can learn to estimate the strength of the uncertainty in the current step. Subsequently, an LSTM network and a transformer encoder are employed to model the serial relationship in the trajectory. Through experiments, we

found that the LSTM network is very important for generating a continuous trajectory and that 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. The decoder then decodes the uncertainty sequence step-by-step and takes the output of the last moment as the input. The decoding process begins by embedding a start token, which ensures that the generated series will perform well in the serial relationship.

4.7. Trajectory decoder

The uncertainty predicted by the trajectory generator is used to generate X_{t-1} by sampling from $p_{\theta}(X_{t-1}|X_{t-2})$ which is represented as a minus symbol in Figure 3. Through T steps of iterations following the sampling process, X_0 is predicted. Trajectory decoder D is used to decode the hidden representation of a trajectory into the probability distribution of all candidate locations. We employed a transformer encoder and decoder structure. The input of the decoder was X_t , the decoder decoded a trajectory from the memory of the encoder, and another series of the embedding of the output token. We designed the trajectory decoder to be as simple as possible so that the model mostly relies on the trajectory generator to model human mobility by estimating its uncertainty.

5. Experiment

5.1. Datasets

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

T-Drive: This dataset contains real taxi GPS trajectories in Beijing, China. It contains the trajectory of 10,357 taxis from Feb. 2 to Feb. 8, 2008. The average sample frequency was 2.95 minutes. Considering the problem of missing data, we resampled the location of every taxi every 5 min such that all trajectories in the dataset had a fixed time interval. We extracted the positioning points on the six-ring road, which accounted for 98.2% of all the points in the dataset. Then, the region in the six-ring road was divided into 27×27 grids by a square with a 2000 meters edge length, which was determined by the mobility frequency and average moving distance in the dataset. A total of 169,984 trajectories were recorded.

Geo-life: This dataset was collected from 182 individuals. GPS trajectories record their mobility activities over five years. We also resampled all trajectories into a 5 min time interval and extracted points on the six-ring road. Considering that mobility activity was relatively weak, the division was set to 500 m, so there were 110×110 grids in total. At last, there are 79,360 trajectories left.

All trajectories in each dataset were resampled to a 5 min time interval. The model was trained using human mobility trajectories for 1 hour, which included 12 trajectory points for every trajectory. The principle of selecting a proper grid scale for a dataset is to ensure that two temporally adjacent trajectory points are divided into spatially adjacent grids. For the T-Drive dataset, the average driving speed for a taxi is 30

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

kilometers per hour, which is 2500 meters per 5 minutes. The current 2000-meters scale makes 78.4% of adjacent trajectory points in T-Drive located in spatially adjacent grids. Similarly, for the Geo-life dataset, the average moving speed for pedestrians is 400 meters per 5 minutes, considering there are other traveling modes besides walking. The current 500-meters grid scale ensures that 73.6% of adjacent trajectory points in T-Drive located in adjacent or secondary adjacent grids.

The dimension of location embedding is set to 512. The number of layers for trajectory encoder E was set to 2. The number of hidden units in all layers in the model was set to 512. The number of diffusion steps in the model was 1,000. The model was implemented using PyTorch and trained on an Nvidia Titan V GPU.

5.2. Evaluation metrics

We employed five different evaluation metrics to quantitatively evaluate the quality of the 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 JSD is defined as follows:

$$JSD(D, \hat{D}) = \frac{1}{2} KL(D, \frac{D + \hat{D}}{2}) + \frac{1}{2} KL(\hat{D}, \frac{D + \hat{D}}{2}) \quad (25)$$

where D and \hat{D} are two different distributions, and KL measures the Kullback–Leibler divergence between two distributions. It can be formulated as follows:

$$KL(D, \hat{D}) = \sum_{x \in \mathcal{X}} D(x) \log \frac{D(x)}{\hat{D}(x)} \quad (26)$$

where x denotes a sample point in the discrete sample space X : The smaller the JSD, the more similar the two datasets. We measured the JSD between the generated and real datasets of the following five metrics.

5.2.1. Moving distance (moving)

The mobility of a moving object is an important characteristic in evaluating the generation quality of an individual trajectory. This metric measures the distance between two adjacent moments in a trajectory. It reflects the individual mobility patterns of all the trajectories in a dataset.

5.2.2. Geographical distribution (distribution)

In addition to simulating the mobility of every single person, it is important to generate a dataset that follows the same geographical distribution as the real one. This metric evaluates the geographical distribution of the trajectory points in all generated trajectories. A geographical distribution similar to a real distribution promises the basic utility of the generated dataset for downstream tasks.

5.2.3. 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, evaluation metrics were developed to measure the usability of the generated trajectory in OD flow pattern mining. O-Dis measures the location distribution of the start locations in all trajectories in a dataset. D-Dis measures the distribution of the end locations of all trajectories.

5.2.4. Diversity

The diversity of the generated trajectories is also very important from the perspective of the utility of a forged dataset. The diversity of the trajectory dataset was measured by counting the ratio of trajectories appearing more than twice in the dataset. If two trajectories had the same sequence of location indexes, we defined them as the same. The lower the ratio, the more diverse the dataset. The quality of the generated dataset can be determined based on whether the value of the metric is close to that of the real dataset.

5.3. Baselines

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

FC-LSTM (Sutskever *et al.* 2014): The FC-LSTM is a well-known discriminative model for dealing with sequence-to-sequence tasks. It starts generating a trajectory by sampling a point from the density distribution of all trajectory points, and the remaining trajectory is generated using a step-by-step process. The dimension of the 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): This is a state-of-the-art method used for human mobility simulations. This discriminative model is based on a GAN structure. This method samples the starting point of a trajectory from the population density distribution. The remaining trajectory was then output from a specifically modified trajectory prediction model called SeqNet. The model was trained using a discriminator with a reinforcement-learning technique. The dimension of the location embedding and the number of hidden units in all networks were set to 512, which are the same as ours.

SeqGAN (Yu *et al.* 2017): SeqGAN is a benchmark model for sequence generation. It combines GAN with reinforcement learning to provide the generator with a policy gradient. Two GRU networks were employed as its generator and discriminator. Unlike the original GAN, which starts generating a representation randomly sampled from the latent space, the model starts generating a trajectory with the trainable embedding of a start token. Therefore, the model can only 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. A variational autoencoder is a typical generative model. The model employs two LSTM networks as its encoder and decoder for the VAE. It generated a trajectory by decoding a sampled vector from a Gaussian latent space.

Generative SeqGAN (Yu *et al.* 2017): This model is a generative version modified from the seqGAN. As mentioned previously, seqGAN is regarded as a discriminative model owing to its deficiency in the latent space. We modified the seqGAN by changing its start token using a latent representation sample from a normal distribution. Thus, the model generates a trajectory by mapping the latent distribution to reality.

TrajSynVAE (Wang *et al.* 2024): TrajSynVAE is a novel trajectory generation model that combines the classical temporal point process with a novel neural variational inference framework, leading to a strong ability to model human trajectories with a continuous temporal distribution. The model begins generating a trajectory by sampling from the latent space; therefore, it is a generative model.

6. Results

6.1. Trajectory generation

We compared our trajectory generation model with six baseline methods on two datasets. To evaluate the quality of the generated trajectory, we calculated the distributions of the five metrics in each generated dataset and compared them with their distribution in a real dataset. We also calculated the performance of the metrics for a test dataset that was randomly sampled from a real trajectory dataset. Considering the variation and complexity of human mobility, differences can be observed between the two subsets sampled from the same dataset. A test dataset was used to evaluate this part of the difference. We can evaluate the quality of the trajectory data by comparing with the training dataset. A comparison of the results is presented in Table 1.

The models in Table 1 can be divided into discriminative and generative models according to whether they generate data by sampling from a latent space. Models

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

Metrics (JSD)	T-Drive				
	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
TrajSynVAE	0.3922	0.1234	0.1467	0.1426	0.0
TrajGDM	0.05490	0.1171	0.1358	0.1326	0.01367

Metrics (JSD)	Geo-life				
	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
TrajSynVAE	0.5824	0.3006	0.3212	0.3098	0.0
TrajGDM	0.1142	0.1226	0.1231	0.1386	0.02226

such as FC-LSTM, seqGAN, and MoveSim are regarded as discriminative. Although MoveSim and seqGAN are based on a GAN structure and output data using a so-called generator, the generator does not learn to map a trajectory from a latent distribution. They employed discriminative trajectory-prediction networks to generate trajectories by repeating the prediction process. The trajectory prediction network was trained to model the conditional distribution given the observed part of the trajectory, which can be denoted as $P_{i|loc_1, loc_2, \dots, loc_{i-1}}$ where the emerging probability of a point loc_i is determined by its previous $i - 1$ trajectory points. This distribution is theoretically different from the true learning objective for a trajectory generation model, which is $P_{i|loc_1, loc_2, \dots, loc_i}$ denotes the emerging probability of an entire trajectory. However, when the dataset is sufficiently large and the division of the predicted trajectory points loc_i and the observed trajectory points $loc_1, loc_2, \dots, loc_{i-1}$ is randomly sufficient, these two distributions overlap to some extent. Therefore, a discriminative model can output a set of trajectories that are likely to fit the distribution of a real trajectory dataset. This was the result of the overlap of the two different distributions. Moreover, it benefits from directly maximizing the probabilistic that each trajectory point emerges by $P_{i|loc_1, loc_2, \dots, loc_{i-1}}$. The discriminative models can perform well in fitting the general geographical distribution. This explains why most discriminative models show better performance in terms of the Distribution, O-Dis, and D-Dis metrics. To start the predicting iteration, models such as 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 good performance on the O-Dis metric and further improves 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 limited the diversity of the outputs. Theoretically, once a discriminative model completes training, its output probability under every circumstance is

fixed. Considering that the starting method is also fixed, the model is very likely to output highly repetitive trajectories. This is reflected in the poor performance of all discriminative models in terms of the diversity metric.

In contrast to discriminative models, generative models, such as TrajVAE, Generative SeqGAN, TrajSynVAE, and TrajGDM, model trajectories with a continuous latent space. Generation begins by sampling a representation vector from the latent space, and different sample points in the latent space represent different trajectories. Generative models generate trajectories by mapping a representation vector from a latent space to a trajectory through a generation process. Theoretically, considering that the latent space is continuous, a generative model can generate numerous variations of a trajectory. This explains why all the generative models showed very low repetition rates in the diversity metric. In particular, our method exhibited outstanding performance on the diversity metric, which was closest to the repetition rate of the validation trajectory dataset. For TrajVAE, TrajSynVAE, and Generative SeqGAN, the repetition rate was zero because of their poor performance in simulating the individual trajectory. Their generated trajectories are not continuous in space; therefore, generated trajectories have a lower chance of being similar to each other.

While other generative models show poor 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%. The ratio comes to 50.3% in T-Drive. This implies that the generated trajectories are more similar to trajectories in the real world. Moreover, benefitting from directly modeling the trajectory distribution $P_{h|l_0, loc_1, loc_2, \dots, loc_{l-1}}$, our method also shows a great improvement in the geographical distribution metric. As for the OD distribution, except for MoveSim, which generates trajectory origins by sampling from the real distribution, our method still shows better performance than all other parametric methods. The difference in generation performances of two datasets is mainly caused by the data variation in two datasets. This could be reflected by similarity between the training dataset and the Test Dataset (Line 1 in Table 1). The similarity is higher in the Geo-life dataset than the T-Drive dataset, which means that the data variation between different trajectories in the Geo-life dataset is higher. The variation may be caused by several reasons, including the scale of the grids, variation of the trajectory points distribution density, etc. Therefore, a model's best simulation performances of different datasets are correlated with the data variation, and this variation could be reflected by the similarity between the training dataset and test dataset.

To understand how a trajectory was generated, we visualized the trajectories at different steps of the generation process. We also selected several real trajectories that were similar to the generated trajectories for comparison. The results are shown in Figure 5. The background color represents the average generating probability of all the points in the trajectory.

As it shown in Figure 5, the generated trajectories are located in both downtown area and suburban area, and various mobility intensities are shown in different trajectories. An interesting phenomenon that occurred in all trajectories was that the origin and destination were determined in the first place. Even though in the trajectory decoder, the generation of a trajectory's destination is partially based on its previous points because of its sequence-to-sequence structure, and the location points before

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

the destination still vary, destinations in most of the generated trajectories remain unchanged after the first few steps. This phenomenon demonstrates that the important factors are determined in the first place in human movement, which means that important factors, such as the purpose of travel, are included in the latent representation of a trajectory. The details of the trajectory are specified later based on these key factors using our trajectory generator. This uncertainty-reducing generation process is similar to the route-planning process for humans. 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.

Although the number of diffusion steps was set to 1000, the generated trajectories did not change significantly after the first 300 steps. This is because our optimization target was set to directly estimate X_0 from X_t and the trajectory points were nominated according to the output probability distribution of the model. The probability distribution was fixed after hundreds of steps.

6.2. Trajectory general intelligence

To generate a realistic trajectory, a trajectory generator is required to capture the universal human mobility patterns that suit the entire trajectory dataset. Human mobility

patterns play an important role in almost 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 a number of different models, with each of them only capturing part of the pattern mobility and being able to handle a related problem. Zero-shot trajectory prediction and reconstruction experiments were performed using the proposed model. The experiment tested whether our model captured universal mobility patterns by learning the generation process of a trajectory. The target is similar to the basic task of the Large Language Model (LLM), which learns knowledge (corresponding to the universal mobility pattern) from a language dataset (corresponding to the human mobility dataset) of human beings. The success of the LLM has shown that it is possible to achieve this. Zero-shot learning is widely used to evaluate the LLM performance on different unseen tasks to illustrate the knowledge learned by the LLM (Bommarito *et al.* 2023, Kojima *et al.* 2022, Liu *et al.* 2023, Wei *et al.* 2021). The zero-shot inference result is expected to verify the possibility of achieving trajectory general intelligence through modeling the universal human mobility pattern in a trajectory dataset. To the best of our knowledge, few trajectory modeling methods can conduct zero-shot inference on multiple tasks.

6.2.1. Trajectory prediction

Trajectory prediction is one of the most important tasks in trajectory data mining (Zheng 2015). A model is required to capture the mobility patterns between the observed and predicted trajectories. We employed a finished trained trajectory-generation network. To achieve this, instead of putting the latent representation of the entire trajectory to the trajectory generator, the input to the generation network is changed to a 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 is formulated as follows:

$$X_T \sim \text{sampler}(N(0, I)) \quad (27)$$

$$X_t \sim G(\text{concat}(E(P_1|loc_1), P_1|loc_2, \dots, P_{t-1}|loc_{n-1}), X_{t-1}) \quad (28)$$

$$Y \sim D(X_0) \quad (29)$$

where $X_t \in \mathbb{R}^d$, I 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 . For trajectory prediction, the sampler sample at 0. This is further explained in the latent space interpretation section. Considering that the generation model had not been previously trained for the prediction task, it was believed to be a zero-shot prediction. The trajectory generator was trained to generate a trajectory sequence according to its spatial-temporal relationship. Therefore, by providing the generator with exact information on the observed part of the trajectory, the model is expected to generate the remaining part accordingly. The accuracy of the prediction was calculated by the ratio of trajectories in which the predicted location indexes were the same as the real indexes. When predicting more than one future location, only trajectories with all predicted indexes are correct and regarded as successful predictions.

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%

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 that of DeepMove (Feng *et al.* 2018), which is a benchmark deep-learning human mobility prediction method. For a fair comparison, we abandoned the use of other features such as user ID. We then compared the prediction accuracies for different prediction lengths. A comparison of the results is presented in Table 2.

As shown in Table 2, the zero-shot prediction accuracy of our model surpassed the accuracy of DeepMove in the T-Drive dataset by up to 23.4%; however, the accuracy was lower than that of DeepMove in the Geo-life dataset. This can be explained by the difference in learning objective between the two models, DeepMove is trained to maximize the likelihood of $P(y_{1:n}|loc_{1:n-l}, y_{1:l})$ while the objective of our generative model is to maximize $P(y_{1:n}|loc_{1:n-l}, loc_{n-l:n})$. The difference can be regarded as DeepMove tending to output, where the moving object is 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 $y_{1:l}|loc_{1:n-l}, loc_{n-l:n}$ most similar to a real one. As the predicted length l increased, the observed $loc_{1:n-l}$ played a less important role and the difference between the two distributions decreased. This explains why the two prediction accuracies become 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 shows the uncertainty reducing process in prediction. TrajGDM gradually reduced the high probability space. Taking the trajectory in the first column as an example, at the beginning, the cab was predicted to move downtown in the center of the area according to the general mobility pattern, followed by most of the trajectory. In the next step, the model adjusts its prediction results based on the historical trajectory and predicts that it will return to its former route. However, the output appears unreasonable because the trajectory moves too far in a step. With the uncertainty gradually removed, the model generated later predictions based mostly on the current observations, and was predicted to move along its way. It also shows the spatial awareness that only nearby locations were predicted to be the high-probability region. A similar pattern can be observed in all the generation processes: the output high-probability region is transferred from a global high-trajectory density region to one that is primarily based on individual movement characteristics.

6.2.2. Trajectory reconstruction

Trajectory reconstruction or imputation is another difficult problem in human mobility modeling. In reality, limited by the cost of data transmission and storage space, most human mobility datasets are collected in a discontinuity form, and the problem of



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

missing data is common (Wang *et al.* 2022, Zhao et al. 2016). Trajectory reconstruction is a key pre-processing 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 was formulated as follows:

$$X_T \sim \text{sampler}(N(0, I), \mu) \quad (30)$$

$$X_t \sim G(\text{concat}(\text{E}(\mu_{t-1}), \text{P}(\mu_{t-1}), \text{E}(\mu_t), \text{P}(\mu_t)), \sigma) \quad (31)$$

$$Y \sim D(X_0) \quad (32)$$

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%

where $X_t \in \mathbb{R}^{l \times d}$, l is the length of trajectory points to be reconstructed. i is the index of the first missing point in the trajectory. We randomly masked parts of the points in a trajectory. The reconstruction follows the formula above. The evaluation method is the same as the prediction, and the accuracy of the reconstruction is also calculated by the ratio of the reconstructed location indexes that are the same as the real one. The reconstruction accuracy was compared with those of several baseline methods. The MDP-TR combines multiple machine learning methods and exhibits excellent performance in reconstructing CDR trajectory data (Li *et al.* 2019). Spatial interpolation, which reconstructs a trajectory by joining each pair of existing points, is one of the most common baseline methods for trajectory reconstruction (Hoteit *et al.* 2014). A comparison of the results is presented in Table 3. MDP-TR cannot handle continuous missing points; therefore, the corresponding positions in the table are masked.

As shown in Table 3, our method achieved the best reconstruction performance in the Geo-life dataset, surpassing both the mechanism and learning methods. The improvement in accuracy was as high as 25.6%. Another advantage of our model is that it is more flexible, can reconstruct continuous missing points, and can reconstruct a trajectory with multiple discrete missing points simultaneously.

The data variation of the two datasets and the different learning strategies of the models are considered to be the main reasons for the difference in models' performance in these two tasks. The similarity between the test and training dataset shows that the data variation of Geo-life is much higher than that of T-Drive, which explains why most models perform worse in Geo-life in prediction and reconstruction. While T-Drive records trajectories of 10,357 taxis, Geo-life records the movement of 182 individuals. DeepMove's Historical Attention module allows the model to focus on the historical pattern of each specific individual, making it more likely to perform well on datasets with few individuals.

The zero-shot prediction and reconstruction abilities prove that the human mobility patterns captured by our model are universal. Learning human mobility from trajectory generation and applying it to other trajectory tasks is a possible way to achieve artificial general intelligence in a trajectory.

6.3. Latent space interpretation

Understanding the relationship between latent and real-world spaces can help interpret how the model learns human mobility. In this section, this relationship is explored by controlling the *sampler* during the sampling process for trajectory prediction. The basic settings were the same as those described in the previous section. It is still a

zero-shot prediction based on a trained model. We changed the latent representations of the points to be predicted using different sampling points in the latent space. It is worth noting that the embedding representations of locations can also be regarded as distributed in latent space (better known as feature space). While, in this research, we only utilize the definition of latent space in generative models, which refers to the feature space contains all the latent trajectory representations.

TrajGDM models human mobility with a Gaussian distribution, which means that in prediction, all appropriate positions loc_i that make the trajectory $\{loc_{1:i-1}, loc_i\}$ like a real one are assumed to follow a normal distribution in the latent space. Therefore, by sampling from different places in the latent space and analyzing their generation results, we can obtain a brief interpretation of how the model builds a connection between the latent and real-world spaces.

We sampled from 5 different places in the latent space $N(\mu, \sigma^2)$ where μ denotes the mean of the Gaussian distribution and σ is its standard deviation. Point μ owns the highest probability density in the entire latent space. While the probability 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, probability distribution, and distance from the predicted point to the last point of the observed trajectory. A comparison of the prediction accuracy and output distance is presented in Table 4.

In Table 4, the prediction accuracy is the highest at point μ and decreases with the probability density reduced in other sampled points. This implies that the high probability density region in the latent space is mapped to the trajectory points with the highest emergence frequency using the trajectory generator. More specifically, when we sampled at point μ , which owns the highest probability density in the latent space, the generated trajectory point also appears most frequently in the circumstance that $loc_{1:i-1}$ is observed. For the sampled points with a lower probability density, their corresponding generated trajectory points have a lower emerging probability. Therefore, the prediction accuracy is relatively low when sampled in these regions. Spatially, this correlation was also reflected in distance metrics. 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 $loc_{1:i-1}$: It is worth noting that even though the emerging frequency is relatively low at these low-probability-density places, they are still sampled in the range of the latent distribution, and the outputs are generated under the guidance of condition $loc_{1:i-1}$: Therefore, their emergence should still be reasonable under condition $loc_{1:i-1}$.

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

Sampled points	T-Drive				
	$\mu - 2\sigma$	$\mu - \sigma$	μ	$\mu + \sigma$	$\mu + 2\sigma$
Probability 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

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

to some extent. To comprehend the relationship between the latent space and real-world space with different sampled points, we visualized the probability distribution in geography. The results of several typical trajectories are shown in Figure 7.

In Figure 7, by comparing the variation of high probability region generated by sampling from different sample points, it is found that sampling at point l generates the most suitable probability distribution for the current trajectory. As mentioned previously, points with high probability density values in the latent space correspond to places with a high emergence frequency. The high probability region generated by sampling at l contains most of the common locations that the trajectory might move to under current observation. Moreover, when sampled points are smaller than l , the high probability region covers a wider area. Interestingly, it is found that the distribution is overlapped with the global high point density region of the whole dataset. While sampling at points greater than l , locations with high probability are discrete from each other, and they are more likely to be the high probability locations that are only suitable with the current trajectory. That is, it generates more personal predictions. As the sampled point leaves l further, this tendency becomes more obvious. This can be explained from the perspective that the model is seeking for a balance between the crowd and individual human mobility patterns. When modeling human mobility, most of the time, trajectories follow a general mobility pattern. Each individual trajectory also contains unique mobility features. The trajectory generation model looks for a way to balance general patterns and personal features. It maps two opposing human mobility characteristics in opposite directions to the latent space. Balance is achieved through random sampling in the latent space.

7. Discussion

We interpreted the relationship between the latent space and real-world space by controlling the sampler sampling at different points in the latent space. Surprisingly, it was found that the latent space was not mapped symmetrically. The model was found

to seek a balance between the crowd and individual mobility patterns. The finding has both theoretical and practical implications. From theoretical perspective, exploring the latent representation of trajectories could reveal how AI bridges latent space with the real-world physical space, this is important for the development of explainable AI. From practical perspective, the finding could help the model diversify and customize generation. By controlling the sampler to sample from the latent space, the generation could to be more general (follow crowd pattern) or customized (follow individual pattern). This quality would be useful when the prediction results are used for location recommendation, as the diversity of recommendations is an important evaluation metric for a recommendation system. The asymmetry mapping pattern could be explained from both the mathematical aspect of deep learning and the geographic aspect of the law of human mobility. More future works are expected to further explore this interesting pattern in the future.

Considering the trajectory generation process is a multiple-step Markov chain, it could take more time for TrajGDM to generate a trajectory than other methods. By generating trajectories in a large batch, the average generation consumption for each trajectory can be reduced. Additionally, methods like DDIM (Song *et al.* 2020) could be used to minimize the generation process into 50 steps, which would significantly reduce the generating time, as long as a little drop down of the generation quality could be tolerated.

To some extent, the scale of the grids may influence the generation quality. The selection of the scale of grids should be decided by many factors, such as the average moving distance between two trajectory points, the traveling mode of the moving object, the total area of the study region, etc. Moreover, instead of using grids to represent locations, other more customized location division methods, such as road segments, blocks, or Thiessen polygons, can also be employed and may significantly boost the model's performance in specific circumstances.

The influence of time is also an important factor in human mobility, which was not considered in the model. We have to admit that time has a significant influence on human mobility. While for mobility simulation, the significance of temporal effects depends on the characteristics of the dataset. For example, for the T-Drive dataset, which records taxi trajectories, as long as we cannot have the travel destination of each passenger who gets on the taxi, it is impossible and meaningless to predict or simulate the long-term movement of a taxi. By focusing on a relatively short term, the model aims at modeling the spatial relationship between locations and the short-term spatial-temporal mobility patterns of trajectories. We minimized the influence of time by only simulating the human mobility pattern in one hour, which is a relatively short period. For datasets like Geo-life, introducing temporal aspects into the model would benefit the model in simulation. In future work, we will further explore the influence of time on human mobility by modeling trajectories over a wider time span.

8. Conclusions

In this study, we proposed a generative human mobility simulation method called TrajGDM. This method models trajectory generation as an uncertainty-reducing process. We proposed a trajectory generator network, which aims to predict the existing

uncertainty in a trajectory. We defined a trajectory diffusion process to model the uncertainty adding process in a trajectory such 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 and generation processes, we introduced our training method, which was designed to train the trajectory generator to learn from a trajectory dataset recorded using a discrete representation method. Finally, we introduced the sampling process for the trajectory generation model. The model can generate a synthesized trajectory dataset that is diverse and realistic through the sampling process.

By comparing the performance of our method with six strong baselines in two public datasets, our model achieved a great improvement in simulating the individual mobility and diversity of the generated trajectories. Moreover, by visualizing the trajectory generation process, we found that the uncertainty-reducing process is similar to the route planning process of a human, indicating that the model learned the generation correctly. After a generation model is trained, zero-shot experiments were conducted on two basic trajectory tasks: trajectory prediction and reconstruction. The generalizability of the model verifies the universal mobility pattern captured by learning the trajectory generation process. This demonstrates that it is feasible to build a trajectory foundation model based on a trajectory generation model. Finally, we explored how our model learns human mobility. The model was found to seek a balance between the crowd and individual mobility patterns.

For our future work, on the one hand, we expect to further improve the generalizability of mobility models. Training the model to learn the mobility pattern from multiple mobility datasets could be the next step towards the realization of mobility AGI. On the other hand, we also expect to further explore AI's understanding of human mobility to support the development of explainable spatial AI. Latent space interpolation would be the earliest and most direct way. Another interesting direction is to learn the internal and external factors that drive human mobility separately. We hope that the development of mobility simulation could lead to the development of the theory of human mobility.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The research was supported by the National Key Research and Development Program of China (No. 2022YFB3904102) and China National Postdoctoral Support Program for Innovative Talents(No.BX20230360). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders. All the authors acknowledge the anonymous reviewers for providing valuable comments and suggestions.

Notes on contributors

Chen Chu is a Master candidate 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. He contributed to the conception of the research idea, algorithm implementation, visualization, and the writing of the manuscript. chuchen0411@igsnnr.ac.cn.

Hengcai Zhang is 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 3D computing, urban computing and spatial-temporal data mining. He contributed to the supervision of the experiments, manuscript revision, and the writing of the manuscript. zhanghc@lreis.ac.cn.

Peixiao Wang is a Postdoctoral Fellow from State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Nature Resources Research, Chinese Academy of Sciences. He received Ph.D. degree under from State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, and received the M.S. degree from The Academy of Digital China, Fuzhou University. His research topics include spatiotemporal data mining, and spatiotemporal prediction, especially focus on spatiotemporal prediction of transportation systems. He provided the computational resource and contributed to the manuscript revision. wpx@lreis.ac.cn.

Feng Lu is a Professor at 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. He contributed to the conception of the research idea, manuscript revision and supervision. luf@lreis.ac.cn.

ORCID

Hengcai Zhang <http://orcid.org/0000-0002-5004-9609>

Data and codes availability statement

The trajectory data, codes and trained models are available at: <https://doi.org/10.6084/m9.fig-share.23804769.v2>. The data and codes are also available at: <https://github.com/chuchen2017/TrajGDM>.

References

- Bao, Y., *et al.*, 2021. A BiLSTM-CNN model for predicting users' next locations based on geo-tagged social media. *International Journal of Geographical Information Science*, 35 (4), 639–660.
- Bommarito, J., *et al.*, 2023. GPT as knowledge worker: a zero-shot evaluation of (AI)CPA capabilities. arXiv preprint arXiv:2301.04408.
- Chen, X., *et al.*, 2021. TrajVAE: A Variational AutoEncoder model for trajectory generation. *Neurocomputing*, 428, 332–339.
- Dhariwal, P., and 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., *et al.*, 2018. DeepMove: Predicting human mobility with attentional recurrent networks. Proceedings of the 2018 World Wide Web Conference, 1459–1468.
- Feng, J., *et al.*, 2020. Learning to Simulate human mobility. Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 3426–3433.

- Gong, S., et al., 2022. Diffuseq: Sequence to sequence text generation with diffusion models. arXiv preprint arXiv:2210.08933.
- Goodfellow, I., et al., 2020. Generative adversarial networks. *Communications of the ACM*, 63 (11), 139–144.
- Gu, T., et al., 2022. Stochastic trajectory prediction via motion indeterminacy diffusion. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 17113–17122.
- Ho, J., Jain, A., and Abbeel, P., 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 6840–6851.
- Hoteit, S., et al., 2014. Estimating human trajectories and hotspots through mobile phone data. *Computer Networks*, 64, 296–307.
- Huang, D., et al., 2019. A variational autoencoder based generative model of urban human mobility. IEEE conference on multimedia information processing and retrieval (MIPR), 425–430.
- Isaacman, S., et al., 2012. Human mobility modeling at metropolitan scales. Proceedings of the 10th international conference on Mobile systems, applications, and services, 239–252.
- Ji, Y., et al., 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.
- Jiang, S., et al., 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences of the United States of America*, 113 (37), E5370–E5378.
- Jiang, W., et al., 2023. Continuous trajectory generation based on two-stage GAN. arXiv pre-print arXiv:2301.07103.
- Jiao, R., et al., 2022. TAE: A semi-supervised controllable behavior-aware trajectory generator and predictor. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 12534–12541.
- Kang, L., et al., 2017. Road2Vec: Measuring traffic interactions in urban road system from massive travel routes. *ISPRS International Journal of Geo-Information*, 6, 321.
- Kang, L., et al., 2023. Act2Loc: a synthetic trajectory generation method by combining machine learning and mechanistic models. *International Journal of Geographical Information Science*, 1–25.
- Karnewar, A., and Wang, O., 2020. MSG-GAN: Multi-scale gradients for generative adversarial networks. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 7799–7808.
- Kingma, D.P., and Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv: 1312.6114.
- Kodali, N., et al., 2017. On convergence and stability of GANs. arXiv pre-print arXiv: 1705.07215.
- Kojima, T., et al., 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35, 22199–22213.
- Li, L., Pagnucco, M., and Song, Y., 2022a. 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, 2231–2241.
- Li, M., et al., 2019. Reconstruction of human movement trajectories from large-scale low-frequency mobile phone data. *Computers, Environment and Urban Systems*, 77, 101346.
- Li, M., et al., 2020. Predicting future locations of moving objects with deep fuzzy-LSTM networks. *Transportmetrica A: Transport Science*, 16 (1), 119–136.
- Li, M., et al., 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.
- Li, X., et al., 2022b. Diffusion-LM improves controllable text generation. *Advances in Neural Information Processing Systems*, 35, 4328–4343.
- Liu, Z., et al., 2018. Mapping hourly dynamics of urban population using trajectories reconstructed from mobile phone records. *Transactions in GIS*, 22 (2), 494–513.

- Liu, Z., *et al.*, 2022. A LSTM-based approach for modelling the movement uncertainty of indoor trajectories with mobile sensing data. *International Journal of Applied Earth Observation and Geoinformation*, 108, 102758.
- Liu, Z., *et al.*, 2023. DelD-GPT: zero-shot medical text de-identification by GPT-4. arXiv pre-print arXiv:2303.11032.
- Mai, G., *et al.*, 2020. Multi-scale representation learning for spatial feature distributions using grid cells. arXiv preprint arXiv:2003.00824.
- Mai, G., *et al.*, 2022. A review of location encoding for GeoAI: methods and applications. *International Journal of Geographical Information Science*, 36 (4), 639–673.
- Mai, G., *et al.*, 2023. Sphere2Vec: A general-purpose location representation learning over a spherical surface for large-scale geospatial predictions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202, 439–462.
- Mao, W., *et al.*, 2023. Leapfrog diffusion model for stochastic trajectory prediction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5517–5526.
- Musleh, M., 2022. Towards a unified deep model for trajectory analysis. Proceedings of the 30th International Conference on Advances in Geographic Information Systems, 109.
- Noulas, A., *et al.*, 2012. A tale of many cities: universal patterns in human urban mobility. *PLOS One*, 7 (5), e37027.
- Ouyang, K., *et al.*, 2018. A non-parametric generative model for human trajectories. Proceedings of the 27th International Joint Conference on Artificial Intelligence, 3812–3817.
- Qi, M., *et al.*, 2020. Imitative non-autoregressive modeling for trajectory forecasting and imputation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12736–12745.
- Rao, J., *et al.*, 2020. LSTM-TrajGAN: A deep learning approach to trajectory privacy protection. arXiv pre-print arXiv:2006.10521.
- Rombach, R., *et al.*, 2022. High-resolution image synthesis with latent diffusion models. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 10684–10695.
- Salzmann, T., *et al.*, 2020. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *Computer Vision – ECCV*, 12363, 683–700.
- Schlapfer, M., *et al.*, 2021. The universal visitation law of human mobility. *Nature*, 593 (7860), 522–527.
- Simini, F., *et al.*, 2021. A deep gravity model for mobility flows generation. *Nature Communications*, 12 (1), 6576.
- Song, C., *et al.*, 2010. Limits of predictability in human mobility. *Science (New York, N.Y.)*, 327 (5968), 1018–1021.
- Song, J., Meng, C., and Ermon, S., 2020. Denoising diffusion implicit models. arXiv pre-print arXiv:2010.02502.
- Song, X., Kanasugi, H., and 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, 2618–2624.
- Sutskever, I., Vinyals, O., and Le, Q.V., 2014. Sequence to sequence learning with neural networks. Proceedings of the 27th International Conference on Neural Information Processing Systems, 3104–3112.
- Tashiro, Y., *et al.*, 2021. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34, 24804–24816.
- Theis, L., *et al.*, 2016. A note on the evaluation of generative models. arXiv pre-print arXiv:1511.01844.
- Vaswani, A., *et al.*, 2017. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000–6010.
- Wang, H., *et al.*, 2024. Synthesizing human trajectories based on variational point processes. *IEEE Transactions on Knowledge and Data Engineering*, 1–15.
- Wang, P., *et al.*, 2019. A hybrid markov and LSTM model for indoor location prediction. *IEEE Access*, 7, 185928–185940.

- Wang, P., et al., 2022. A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation. *International Journal of Geographical Information Science*, 36 (6), 1231–1257.
- Wei, J., et al., 2021. Finetuned language models are zero-shot learners. arXiv pre-print arXiv: 2109.01652.
- Yin, M., et al., 2018. A generative model of urban activities from cellular data. *IEEE Transactions on Intelligent Transportation Systems*, 19 (6), 1682–1696.
- Yu, L., et al., 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31 (1), 2852–2858.
- Yu, Y., et al., 2023. A Bi-LSTM approach for modelling movement uncertainty of crowdsourced human trajectories under complex urban environments. *International Journal of Applied Earth Observation and Geoinformation*, 122, 103412.
- Zhao, Z., et al., 2016. Understanding the bias of call detail records in human mobility research. *International Journal of Geographical Information Science*, 30 (9), 1738–1762.
- Zheng, Y., 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology*, 6 (3), 1–41.