



# GeoGail: A Model-Based Imitation Learning Framework for Human Trajectory Synthesizing

YUCHEN WU, Carnegie Mellon University, Pittsburgh, PA, USA

HUANDONG WANG, CHANGZHENG GAO, DEPENG JIN, and YONG LI, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China

Synthesized human trajectories are crucial for a large number of applications. Existing solutions are mainly based on the generative adversarial network (GAN), which is limited due to the lack of modeling the human decision-making process. In this article, we propose a novel imitation learning-based method to synthesize human trajectories. This model utilizes a novel semantics-based interaction mechanism between the decision-making strategy and visitations to diverse geographical locations to model them in the semantic domain in a uniform manner. To augment the modeling ability to the real-world human decision-making policy, we propose a feature extraction model to extract the internal latent factors of variation of different individuals and then propose a novel self-attention-based policy net to capture the long-term correlation of mobility and decision-making patterns. Then, to better reward users' mobility behavior, we propose a novel multi-scale reward net combined with mutual information to model the instant reward, long-term reward, and individual characteristics in a cohesive manner. Extensive experimental results on two real-world trajectory datasets show that our proposed model can synthesize the most high-quality trajectory data compared with six state-of-the-art baselines in terms of a number of key usability metrics and can well support practical applications based on trajectory data, demonstrating its effectiveness. Furthermore, our proposed method can learn explainable knowledge automatically from data, including explainable statistical features of trajectories and statistical relation between decision-making policy and features.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Applied computing**; • **Computing methodologies** → **Modeling and simulation**;

Additional Key Words and Phrases: Mobility Trajectory, Imitation Learning, Generative Models

Associate Editor: Hui Xiong

Authors' Contact Information: Yuchen Wu, Carnegie Mellon University, Pittsburgh, PA, USA; e-mail: yuchenwu@andrew.cmu.edu; Huandong Wang (corresponding author), Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China; e-mail: wanghuandong@tsinghua.edu.cn; Changzheng Gao, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China; e-mail: gcz20@mails.tsinghua.edu.cn; Depeng Jin, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China; e-mail: jindp@tsinghua.edu.cn; Yong Li, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China; e-mail: liyong07@tsinghua.edu.cn.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2024 Copyright held by the owner/author(s).

ACM 1556-472X/2024/12-ART20

<https://doi.org/10.1145/3699961>

**ACM Reference format:**

Yuchen Wu, Huandong Wang, Changzheng Gao, Depeng Jin, and Yong Li. 2024. GeoGail: A Model-Based Imitation Learning Framework for Human Trajectory Synthesizing. *ACM Trans. Knowl. Discov. Data.* 19, 1, Article 20 (December 2024), 23 pages.  
<https://doi.org/10.1145/3699961>

## 1 Introduction

Synthetic human trajectories are instrumental for a wide-range of applications, including network service optimization, transportation scheduling, and so on. For example, in cellular networks, based on synthesized trajectories, we can simulate the detailed process of movement and communication of network users to have a reliable performance analysis of the network [13, 24]. Similarly, we can simulate the traffic congestion before and after the implementation of certain policies (e.g., road expansion) based on synthetic human trajectories, which are crucial for urban planning [23, 39].

However, synthesizing human trajectories is also challenging due to the complicated spatio-temporal correlation and large stochasticity of the mobility trajectories. The rise of the deep learning paradigm provides promising solutions to synthesize high-quality human trajectories [10, 18, 22, 26, 36, 44], where the most successful and prominent methods are based on the **generative adversarial network (GAN)**. Specifically, GAN utilizes a generator to synthesize new data and a discriminator to distinguish generated data and real data. Then, it is trained through a mutual game between the generator network and the discriminator network. Utilizing the strong modeling ability of GAN, existing approaches combine it with **Convolutional Neural Network (CNN)** [10, 26, 36, 45, 50] and **Recurrent Neural Network (RNN)** [18, 22, 40] to synthesize human trajectories. However, human trajectories are generated by the complicated human decision-making process [8, 31], while GAN is designed to directly learn the data distribution from demonstrations without modeling the hidden decision-making process, leading to its weakness including the frequent mode collapse, lack of interpretability, and under-performance on unseen data [2, 5].

In this article, we propose a novel framework to synthesize human trajectories based on **imitation learning (IL)**. Different from GAN, our proposed method seeks to model the human decision-making process based on the state-of-the-art **generative adversarial imitation learning (GAIL)** method. After elaborately designing a feature extraction module to obtain important decision-making features, we propose a novel self-attention-based policy net, which can capture the long-term correlation of the decision-making process, and a multi-scale reward net that models both instant reward and long-term reward of human mobility. Furthermore, mutual information is used to reward synthesized trajectories to guarantee that individual characteristics play a vital role in guiding the decision-making process. Further, we propose a semantics-based action space and design a semantics-aware state transition model, which decouples the decision-making process from the geographical trajectories with large spatial differences. Specifically, the decision of visiting diverse locations is uniformly modeled in the semantic domain, which helps to extract the principle human decision-making process in terms of mobility behavior. In summary, our article makes the following contributions:

- We propose a novel framework based on IL, which is the first work to utilize the IL technique to synthesize human trajectories. Based on the obtained human mobility strategy, our proposed model can synthesize high-quality trajectories by simulating the human decision-making process.
- We propose a novel semantics-based interaction mechanism to decouple the neural network-based decision-making strategy from diverse geographical locations and alternatively model them in the semantic domain in a uniform manner, which helps extract the principle human

decision-making process of mobility behavior. Further, we design neural networks with novel structures to model the decision-making strategy, including a self-attention-based policy net to capture the long-term correlation of mobility and decision-making patterns, and a multi-scale reward net combined with mutual information to model the instant reward, long-term reward, and individual characteristics in a cohesive manner.

- We conduct extensive experiments on two real-world mobility datasets. Results show that compared with state-of-the-art algorithms, our model can synthesize the most high-quality trajectories in terms of a number of key usability metrics. Specifically, the performance metrics measured by **Jensen–Shannon divergence (JSD)** is improved on average. In addition, our model can well support practical applications based on trajectory data, i.e., the mobility prediction accuracy can be significantly improved based on trajectory datasets augmented by our proposed model. Furthermore, our proposed method can learn explainable knowledge automatically from data, including explainable statistical features of trajectories and statistical relations between decision-making policy and features.

## 2 Mathematical Model and System Overview

### 2.1 Mathematical Model

We denote the set of all users as  $\mathcal{U}$ . The trajectories of users are composed of a series of spatio-temporal points. To pave the way for modeling the spatio-temporal trajectories of datasets with different resolutions in a consistent manner, we divide time and locations into fix-sized time slots and geographical regions. We denote  $\mathcal{L}$  and  $\mathcal{T}$  as the set of all regions and time slots, respectively. Then, the trajectory of each user  $u \in \mathcal{U}$  can be represented by  $T_u = [(l_1, t_1), (l_2, t_2), \dots, (l_N, t_N)]$  with an identical time gap, i.e.,  $t_2 - t_1 = t_3 - t_2 = \dots = t_N - t_{N-1} = \Delta t$ , where  $N$  is the trajectory length. In addition, for the real-world human trajectory, i.e., the expert trajectory, we denote it as  $T^e$ . For the generated trajectory by models, we denote it as  $T^g$ . Without loss of generality, we only consider trajectories with the duration of whole days. That is,  $N = k \cdot N_d, \forall k \in \mathbb{N}$ , where  $N_d$  is the number of time slots within 1 day. Based on the above notations, the trajectory synthesizing problem investigated in this article can be formally defined as follows:

*Definition 1 (Trajectory Synthesizing Problem).* Given the real-world expert trajectories  $\{T_u^e\}_{u \in \mathcal{U}}$ , synthesizing new trajectories different from the original trajectories. In addition, the synthesized trajectories should preserve the original trajectory dataset's usability. Specifically, the preserved usability means that the synthesized trajectories should statistically resemble real-world trajectories, and can effectively support practical applications based on trajectory data, e.g., mobility prediction.

### 2.2 System Overview

Figure 1 shows the framework of our proposed system for trajectory synthesizing. Basically, this system utilizes the **inverse reinforcement learning (IRL)** techniques to learn the decision-making strategy of real-world users and then uses the learned decision-making strategy for generating trajectories. Specifically, this system takes expert user trajectories as input. Through a feature extraction module, important decision-making features are obtained based on user trajectories, of which the corresponding process is called the data level of our system. Then, the semantics level of our proposed system is utilized as the bridge between the neural network-based decision-making strategy and the data level. Specifically, the decision-making strategy in our system is defined on the action space derived from the classical **exploration and preferential return (EPR)** model [34]. Thus, all actions are able to describe the semantics of user's movement behavior, which are referred to as the semantic-based actions. Furthermore, the transition probability model between different actions discovered in the EPR model [34] is further used in our system, which is referred to as the semantics-aware state transition model. Specifically, the decision-making strategy interacts with

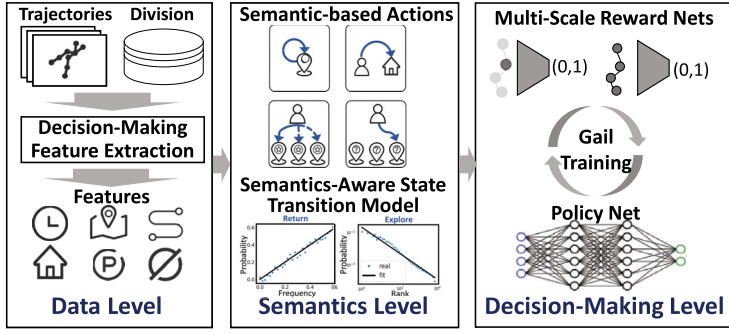


Fig. 1. The framework for our system.

trajectory data through the semantics-aware state transition model combined with the extracted decision-making features. That is, when the decision-making strategy chooses a specific action, the concrete visited location is determined by the probability distribution coming from the semantics-aware state transition model. In this way, the data level and decision-making level of our system are able to be decoupled. For example, when two users with different home locations want to go home, they can directly select the action corresponding to going home regardless of where their homes are. Therefore, our proposed model is able to capture the principle decision-making strategy of users without the interference of the diverse user profile, location profile, and so on. Finally, the decision-making level of our system is composed of two modules including the policy function and the reward function, which are all modeled by neural networks, and the powerful GAIL method is used to learn the parameters of them through a mutual game between the policy function and the reward function.

### 3 Markov Decision Process Modeling

In this section, we introduce how we model the human movement decision-making processes of different users as **Markov Decision Process (MDPs)**.

Specifically, we consider each user as an “agent” with a unique reward function. Formally, each MDP can be represented by a 4-tuple  $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$ . Specifically,  $\mathcal{S}$  represents the state space, and  $\mathcal{A}$  represents the action space.  $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$  represents the state transition probability, That is,  $P(s_{i+1}|s_i, a_i)$  is the probability distribution of the next state  $s_{i+1}$  conditioned on the current state  $s_i$  and action  $a_i$ . Finally,  $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is the reward function with  $R(s, a)$  representing the immediate reward received by executing the action  $a \in \mathcal{A}$  under the current state  $s \in \mathcal{S}$ . Below, we introduce how we model the above components of the MDP.

*State.* The state of each agent is characterized by his/her spatio-temporal locality  $s = (l, t, c)$ , where  $l$  represents the current location of the agent,  $t$  represents the timestamp, and  $c$  represents other decision-making features constructed based on users’ historical trajectories.

*Action.* We consider four actions that agents can take in each time slot including stay, home, explore, and return, which are described in detail as follows:

- *Stay*: When the agent anchors at a location to conduct an activity, it is defined as a stay. This action indicates the agent will stay at the current location in the next time slot.
- *Home*: The home plays an irreplaceable role in the mobility behavior of users. Thus, we define a separate action to describe the home-related mobility behavior. Specifically, this action represents the agent will go home in the next time slot.
- *Return*: This action represents that the agent chooses to move to a location that has been previously visited but not his/her home.
- *Explore*: The action of explore indicates the agent will move to a new location which has not been previously visited.



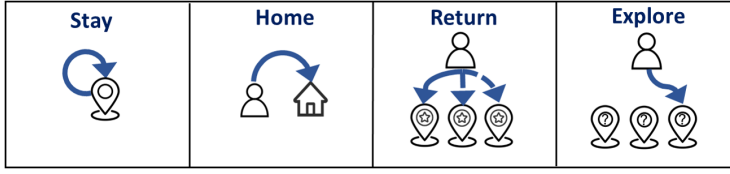


Fig. 2. Illustration of the four actions.

When it comes to model the action space of the decision-making process of human mobility, XGail [27] and TrajGail [52] focus on fine-grained information—instant orientation of the driver. DeepMove [9] and MoveSim [10] consider the coarse-grained location-based variation. To build a more general model that can represent the decision-making process of human mobility, we present those four actions (Figure 2). Explore and return actions have been proven to be effective in representing human mobility in past and recent papers [15, 28]. Some researchers present EPR models for returning action to increase prediction accuracy [4, 30]. As for the home and stay action, researchers have proven that adding the home action can better characterize the human mobility [17, 35]. Thus, these four actions are suitable for representing general human mobility actions.

**State Transition Probability.** This component describes the probability distribution of the next state conditioned on the current state and action. According to our definition of the state space  $\mathcal{S}$ , the next state  $s_{t+1}$  is composed of three parts, i.e.,  $l_i$ ,  $t_i$ , and  $c_i$ . Specifically, the transition of  $t_{i+1}$  is deterministic, and we have  $t_{i+1} = t_i + \Delta t$ . At the same time,  $c_{i+1}$  is determined by the historical trajectories, which is updated by the conditional variable  $s_i$ . Thus,  $c_{i+1}$  is also deterministic. As for  $l_{i+1}$ , when the action  $a_i$  is *stay* or *home*, it is still deterministic with  $l_{i+1}$  to be  $l_i$  or the home place of the agent, respectively. However, when  $a_i$  is *return* or *explore*, there are a large number of possible locations to choose from, and thus  $l_{i+1}$  is stochastic.

We utilize the classical EPR model to select from the new locations to explore or the previous locations to return. Specifically, if the agent decides to return to previously visited locations, he would choose a specific location  $l$  with the following probability distribution:

$$P(l_{i+1} = l | s_i, a_i = \text{Return}) = k_l / k_{all}, \quad (1)$$

where  $k_l$  represents the historical visitation times in place  $l$  of the agent, and  $k_{all}$  represents the total visitation times in all places of the agent.

On the other hand, if the individual decides to explore a new location, we choose a rank-based exploration model. Based on the distances to the current location  $l_i$ , we have a rank  $k(l, l_i)$  for each alternative destination  $l$ . For example, the location  $l_1$  closest to the current location  $l_i$  has  $k(l_1, l_i) = 1$ , and the second closest location  $l_2$  has  $k(l_2, l_i) = 2$ . Then, we have

$$P(l_{i+1} = l | s_i, a_i = \text{Explore}) \propto k(l, l_i)^{-\alpha}, \quad (2)$$

where  $\alpha$  is the sensitivity for distance. Smaller  $\alpha$  means people are less sensitive to trip distance, resulting in longer trip distance.

**Reward.**  $R(s_i, a_i)$  measures the reward that agent receives by executing the action  $a_i$  under the state  $s_i$ . In the IRL framework adopted by our proposed system, the reward function is also known and need to be learned from the expert user trajectories. Specifically, neural networks are adopted to model the reward function, which will be introduced in Section 4.

## 4 GAIL

Based on the above MDP models, the decision-making strategy of the user is described by a policy function  $\pi : \mathcal{S} \rightarrow \mathcal{A}$ . Specifically,  $\pi(a|s)$  represents the probability of choosing action  $a$  under the

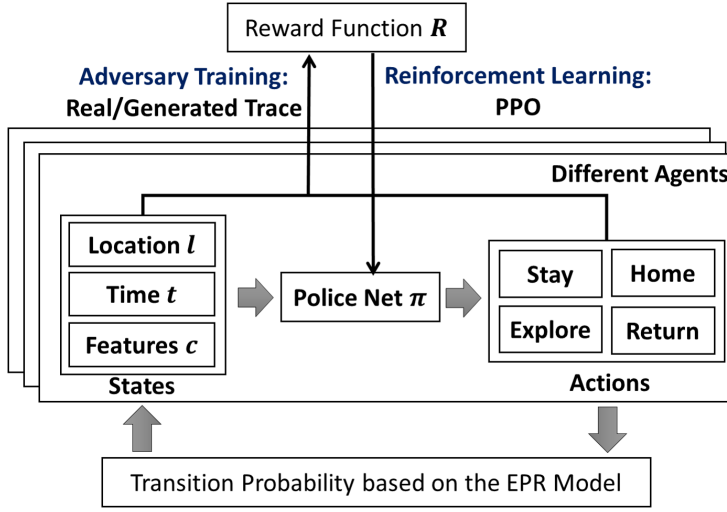


Fig. 3. The framework for GAIL.

current state  $s$ . All users choose the policy functions that maximize their total obtained rewards. Since the policy function and the reward function are all unknown in our problem, our goal is to learn the reward function and policy function from the expert user trajectories.

In order to achieve this goal, we utilize the power GAIL method. Specifically, in GAIL, the policy function is modeled by a generative neural network, which is utilized to sample user trajectories and is optimized by maximizing the total obtained reward of the sampled trajectories given the reward function. On the other hand, the reward function is modeled by the discriminator neural network, which takes both expert trajectories and sampled trajectories as input and output to what degree the sampled trajectories imitates the expert trajectories as the reward value. Finally, the reward function and policy function are joint learned in an adversarial learning manner. Figure 3 summarizes this process.

In the following part of this section, we will introduce the process of utilizing GAIL method to solve our problem in detail. First, we consider the significant variability of trajectories among different individuals and seek to extract the internal latent factors of variation, which are referred to as the decision-making features. Then, based on the extracted decision-making features, we introduce a self-attention generative network to model the policy function in Section 4.2. Next, we design a multi-scale reward function composed of an instant discriminator network to reward the instant state-action pairs and a long-term discriminator network to reward the long-term mobility patterns in Section 4.3. Finally, we present the jointly learning process of the policy net and the reward nets in Section 4.4. The whole structure is shown in Appendix A.1.

#### 4.1 Decision-Making Feature Extraction

Different users have different policy functions and reward functions. For example, the home place of one user might be the workplace of another user. Thus, the rewards that they obtain when visiting this place are completely different, which also leads to their different policy about this place. Thus, the spatio-temporal locality of users alone is not enough to determine their obtained reward and future actions. In order to solve this problem, we seek to extract other important decision-making features expect for their spatial-temporal locality, which can be summarized as follows:

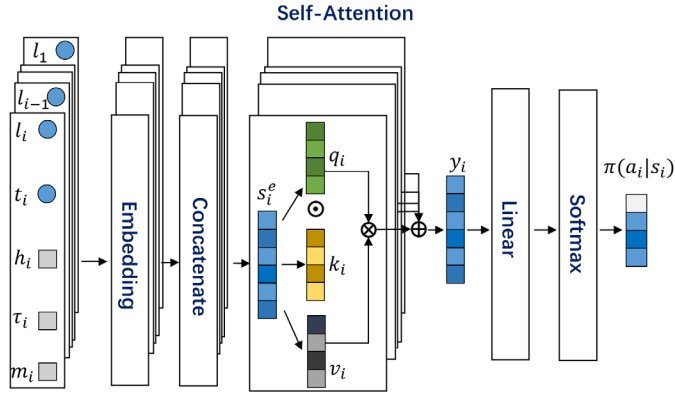


Fig. 4. Illustration of the self-attention-based generator network architecture.

- *Home Position  $h_i$* : As defined in Section 3, we take returning home as an important action in the action space of the MDP. Thus, the home position is also critical information for both policy functions and reward function. Specifically, the home position is labeled by the most frequently visited region during the nights (between 10 p.m. of the first day and 5 a.m. of the second day).
- *Staying Time  $\tau_i$* : Users' staying time at the current location is another important decision-making feature. It has been found that users' staying time at different locations follows a power-law distribution with a cutoff of 17 hours [34]. In our method, we extract the staying time of users at the current location as a decision-making feature and investigate whether we can learn its influence to users' movement behavior through the neural network-based policy function and reward function.
- *#Visited Location  $m_i$* : It is also found the probability of exploration is correlated to the number of distinctive locations visited by the user [34]. Thus, we denote this feature as  $m_i$  and use it as another decision-making feature.
- *Sub-Trajectory of the Past Day  $T_i^c$* : The above features mainly characterize the short-term state of user mobility. The only long-term state feature is the home position  $h_i$ . However, only  $h_i$  is still not sufficient to characterize the long-term states of user mobility. In order to solve this problem, we collect the sub-trajectory of the user within the past day as another feature, which can be represented by  $T_i^c = \{(l_j, t_j) | i - N_d \leq j < i\}$ .

## 4.2 Policy Function

As shown in Figure 4, we design a policy network based on the self-attention mechanism [38] to capture complicated correlations and regularities in the sequential mobility trajectories. The network generates the decision-making policy  $\pi(a|s)$  given the state  $s$ . Based on the learned policy, the network then randomly selects an action (flexible activities that determine the next location).

First, we have to design an encoding structure that converts the original state  $s$  to a specific format that serves as the input for the generator. The main reason for the encoding process is that the state  $s$  usually contains various types of attributes  $l_i, t_i, c_i$ . Here,  $c_i$  only calculates three different decision-making features—home position  $h_i$ , staying time  $\tau_i$ , visited locations  $m_i$ , as the sub-trajectory of the past day  $T_i^c$  has already been processed in the self-attention mechanism. We encode  $l_i, t_i, \tau_i, m_i$  with one-hot vectors and then convert them into embedding vectors based on the learnable parameters of embedding table  $W_l, W_t, W_\tau$ , and  $W_m$ . For the home position  $h_i$ , we use

the encoding method same as the  $l_i$  followed by a ReLU activation layer, which can be represented as follows:

$$h_i^e = \Phi(h_i) = \text{ReLU}(W_l h_i), \quad (3)$$

where  $h_i^e$  is the embedded vectors for home position  $h_i$ . Then, the raw input  $s$  composed of the current location  $l_i$ , current time  $t_i$ , staying time  $\tau_i$ , and the number of visited location  $m_i$  are first converted from the one-hot encoded vectors to their corresponding embedding vectors. Then, they are concatenated together with the embedding of the home position  $h_i^e$  and then jointly mapped into a low-dimensional embedding vector  $s^e$ . Note that the embedding weight matrices are shared among all trajectory points.

$$s_i^e = \tanh([W_l l_i; W_t t_i; W_\tau \tau_i; W_m m_i; h_i^e]). \quad (4)$$

In the generative modeling layer, we use a self-attention-based structure that captures the time-dependent transitions inside the trajectories. Self-attention mechanism [38] is proven to be efficient in sequence modeling and has a strong ability to model the long-term correlation and complicated spatio-temporal dependency of mobility trajectories [9, 21, 43]. We use the scaled dot-product attention with the input consisting of a query vector, a key vector, and a value vector, represented by  $q_i$ ,  $k_i$ , and  $v_i$ , respectively. They are all extracted from the dense vector  $s^e$  by using three independent non-linear networks as follows:

$$[q_i; k_i; v_i] = \text{ReLU}([W_Q s_i^e; W_K s_i^e; W_V s_i^e]). \quad (5)$$

Further, we denote the matrix composed of query vectors, key vectors, and value vectors as  $Q_i$ ,  $K_i$ , and  $V_i$ , respectively. For example, we have  $K_i = [k_1, k_2, \dots, k_i]$ . Then, the attention mechanism can be represented by the following equation:

$$y_i = \left[ \text{softmax} \left( \frac{q_i^T K_i}{\sqrt{d}} \right) V_i^T \right]^T, \quad (6)$$

where  $d$  is the dimension of the key vector  $k_i$ , and  $y_i$  is the obtained feature vector based on the self-attention layer. Finally, we use a linear layer with the softmax activation function to process the feature  $y_i$  to obtain the probability distribution of each action as the policy.

### 4.3 Reward Function

In the GAIL, the reward function is used to score trajectories by matching the generated state-action distribution with the expert's distribution. However, the mobility features of human are complicated and diverse, simply modeled reward function cannot capture the complex information and will face the compounding errors and covariate shift [32, 33]. The whole complex spatial-temporal mobility information can be divided into two different genres—low-level details (instant actions) and global characters (individual varieties and motivation of movement). To better extract these features, we process them separately by designing two reward functions presented in Figure 5. An instant reward function  $R^I$  is designed for directly rewarding the action based on the current status, and the other long-term reward function  $R^L$  is used to distinguish the personality information from the trajectory and measure the trajectory-level similarity with the expert trajectories. Finally, we can obtain the multi-scale reward function  $R^M$  as follows:

$$R^M = R^I + \lambda R^L, \quad (7)$$

where  $\lambda > 0$  is the parameter to balance the influence the instant reward and long-term reward. Then, we introduce how we compute  $R^I$  and  $R^L$  in the following part of this section in detail.

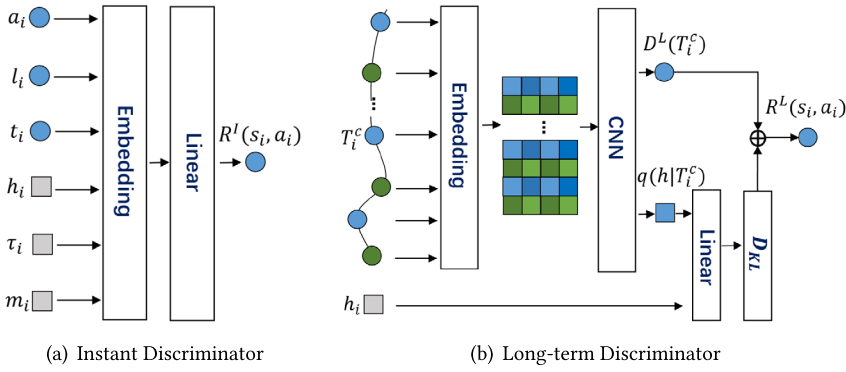


Fig. 5. Illustration of the instant discriminator and long-term discriminator.

**4.3.1 Instant Reward  $R^I$ .** The instant reward is modeled by an instant discriminator  $D^I$ , of which the goal is to measure the reality of the generated actions as human decisions. With the sampled action  $a$  from the generator network and the state  $s$  (instant message without the sub-trajectory of the past day  $T_i^c$ ) as input, the instant discriminator network outputs the reward signal which indicates to what degree the generated state–action imitates the real-world expert trajectories. Specifically, we first apply the same encoding mechanism to map the state–action pair into a dense vector. Then, the embedding vector will be processed by linear layers with ReLU activation functions later to extract the low-level features. Finally, the output will be fed into a linear layer with a sigmoid activation to be transformed into the discriminator output  $D^I(l_i, t_i, \tau_i, m_i, a_i)$ , and the instant reward can be expressed as follows:

$$R^I(s_i, a_i) = \log D^I(l_i, t_i, \tau_i, m_i, a_i). \quad (8)$$

**4.3.2 Long-Term Reward  $R^L$ .** While the instance discriminator network  $D^I$  generates learning signals for the generator by distinguishing the real and generated mobility state–action pairs, it fails to capture the global information and high-level individual characteristics, which is critical for high-quality mobility trajectory simulation. For the global information, we evaluate the reality of the generated trajectory and use a similar adversarial mechanism to make the generated distribution fit the real distribution in the trajectory level. For the individual characteristics, by considering the diversity of humans, we introduce the home position  $h_i$  to disentangle the individual features. As we have fed home position as one of the decision-making features, the generated trajectory  $T^c$  comes from our self-attention generator network should contain the information of the individual feature  $h$ . Inspired from InfoGail [19], we would like to increase the mutual information  $I(T^c; h)$  between the generated trajectory of  $T^c$  and home position  $h$ . Based on the definition,  $I(T_i^c; h)$  can be calculated as follows:

$$I(T^c; h) = H(h) - H(h|T^c) = H(h) + \mathbb{E}_{T^c} \mathbb{E}_h \log p(h|T^c). \quad (9)$$

Without access to the posterior  $p(h|T^c)$ , we cannot maximize the  $I(T^c; h)$  directly. To calculate the lower bound of  $I(T^c; h)$ , we introduce  $q(h|T^c)$  to approximate the true posterior  $p(h|T^c)$ .

$$\log p(h|T^c) = \log q(h|T^c) + \log \frac{p(h|T^c)}{q(h|T^c)}. \quad (10)$$

Further, we have  $\mathbb{E}_{h|T^c} \log \frac{p(h|T^c)}{q(h|T^c)}$  is the KL divergence conditioned on  $T^c$ , which is always larger than 0. Then, we calculate the left part via the reparametrization trick, which can be expressed as follows:

$$L_I(T^c, h) = \int p(h) \log q(h|T^c) dh + H(h) \leq I(T^c; h), \quad (11)$$

where  $p(h)$  is the home distribution of the agent. As  $H(h)$  is a constant, the maximizing  $I(T_i^c; h)$  can be converted into maximizing the left term, which is also equivalent to minimizing the KL divergence  $D_{KL}(p(h)||q(h|T_i^c))$ .

Thus, the output of our long-term discriminator network is designed to have a trajectory-level discriminator output and a home position estimation. Specifically, we show the structure of our discriminator in Figure 5. First, we convert the sub-trajectory information  $T_i^c$  into a 2D feature matrix  $X^e \in \mathbb{R}^{N_d \times N_{emb}}$  by using an embedding module, where  $N_{emb}$  is the length of the embedding vectors. Then, several convolution layers are selected to handle the generated feature matrix. Finally, we flat the convolutional features into a trajectory-level discriminator result for reality  $D^L(T_i^c)$  and a location choice for home position  $q(h|T_i^c)$  by using two additional linear layers.  $q(h|T_i^c)$  is further used to reward the individual characteristics. Specifically, we calculate by the negative KL divergence between real home  $p(h)$  and estimated home position  $q(h|T_i^c)$  as the reward, i.e.,  $-D_{KL}(p(h)||q(h|T_i^c))$ .

Thus, the whole reward function for the long-term discriminator network would be

$$R^L(s_i, a_i) = \begin{cases} \log D^L(T_i^c) - \beta D_{KL}(p(h)||q(h|T_i^c)), & \text{if } i = k \cdot N_d, k \in \mathbb{N}, \\ 0, & \text{otherwise,} \end{cases} \quad (12)$$

where  $\beta > 0$  is the parameter to adjust the influence of the reward in terms of the individual characteristics. Note that in Equation (12), a trajectory with the length of  $k$  days obtains non-zero rewards for exact  $k$  times at the end of each day.

#### 4.4 Model Training

As our GeoGail model is based on two different discriminator networks to reward the whole generation, the standard training algorithm cannot work well for the min-max optimization. To effectively train our model, we present two different kinds of training mechanisms, i.e., pre-training mechanism and GAIL training. For the pre-training mechanism, we pre-train the instant discriminator and generator to accelerate the training procedure and improve the performance of the whole framework. Then, in the GAIL training, we train our model based on the reinforcement learning update rule to maximize the total obtained reward.

**4.4.1 Pre-Training Mechanism.** In order to improve the performance of the whole framework and accelerate the training procedure, we propose to pre-train the model. We use the negative log-likelihood loss for the generator network to gain the fine details inside the expert trajectories by selecting the right actions, which can be expressed as follows:

$$\min_{\pi} - \mathbb{E}_{\pi_E} [\log(\pi(a|s))], \quad (13)$$

where  $\pi_E$  represents the expert policy. In addition,  $\mathbb{E}_{\pi}$  represents the expectation with respect to the trajectories sampled based on the policy  $\pi$ . Specifically, for an arbitrary function  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ , we have  $\mathbb{E}_{\pi} [f(s, a)] = \mathbb{E} [\sum_{i=1}^N f(s_i, a_i)]$ , where  $a_i \sim \pi(\cdot|s_i)$  and  $s_{i+1} \sim P(\cdot|s_i, a_i)$ . Thus,  $\mathbb{E}_{\pi_E}$  represents the expectation with respect to the expert trajectories.

As for the discriminators, they all need the sampled trajectories from the generator. Thus, we first sample trajectories based on the current policy  $\pi$  and then pre-train them based on the standard



GAN loss. Specifically, the loss of the pre-training for the instant discriminator can be expressed as follows:

$$\max_{D^I} \mathbb{E}_{\pi} [\log D^I(s, a)] + \mathbb{E}_{\pi_E} [\log(1 - D^I(s, a))]. \quad (14)$$

Differently, for the long-term discriminator, since each trajectory only obtains non-zero rewards at the end of each day, the corresponding discriminator  $D^L$  is also trained based on the sub-trajectory of each day instead of the state-action pair in Equation (14). Thus, we further define  $\mathbb{E}_{\pi}^L [f(s, a)] = \mathbb{E}[\sum_{k=1}^{N/N_d} f(s_{k \cdot N_d}, a_{k \cdot N_d})]$ . Then, the loss of the pre-training for the long-term discriminator can be expressed as follows:

$$\max_{D^L} \mathbb{E}_{\pi}^L [\log D^L(T^c)] + \mathbb{E}_{\pi_E}^L [\log(1 - D^L(T^c))]. \quad (15)$$

**4.4.2 GAIL Training.** The pre-training process helps accelerate the training procedure by obtaining a policy function by directly matching the state-action distribution  $(s, a)$  of the expert trajectories. However, the obtained policy function cannot guarantee the optimality of the overall reward and also suffers from *exposure bias*. Thus, we then train our model based on GAIL.

Specifically, our reward function is calculated by combining the instant discriminator network  $D^I$  and long-term discriminator network  $D^L$ , while the generator attempts to create more real trajectories by maximizing the total reward. The correctness of the discriminators is governed by optimizing the adversarial energy as in Equations (14) and (15). Differently from the pre-training process, the generator (i.e., policy) is optimized based on the reinforcement learning update rule instead of the negative log-likelihood loss, which can help optimize the total reward and is also not affected by *exposure bias* [14]. Specifically, the objective function can be expressed as follows:

$$\begin{aligned} \min_{\pi} \max_{D^I, D^L} \mathbb{E}_{\pi} [R^M(s, a)] + \mathbb{E}_{\pi_E} [\log(1 - D^I(s, a))] \\ + \lambda \mathbb{E}_{\pi_E}^L [\log(1 - D^L(s, a))] - H(\pi), \end{aligned} \quad (16)$$

where  $H(\pi)$  is causal entropy of the policy  $\pi$ , which can be represented by  $H(\pi) = \mathbb{E}_{\pi} [\log \pi(a|s)]$ . This term helps to find the more “uncertain” policy and maximize the total reward at the same time.

Overall, the loss of Equation (16) still contains the GAN loss of the discriminators shown in Equations (14) and (15). Differently, it updates the policy  $\pi$  via PPO update rule based on the reward  $R^M$  shown in Equation (7).

## 5 Experiments

### 5.1 Experimental Setting

**Datasets.** We use the GeoLife dataset provided by [54] and Mobile dataset to evaluate the performance of our framework. The first one is the widely used mobility dataset, which is collected by 182 volunteers with the GPS-phone for 5 years. The second one is collected from the mobile cellular network, which records the mobility trajectories of one hundred thousand users via base station co-localization. With different collection mechanisms, these two datasets record human mobility with different temporal and spatial resolutions, which guarantees the universality of the evaluation results. The experiment details are shown in Appendixes A.2, A.3, and A.4.

**Baseline Algorithms.** We compare our models with the following state-of-the-art mobility trajectory synthesizing methods.

- *IO-HMM* [48]: This model utilizes a modified Hidden Markov Process, which can incorporate external information of the sequence data as the context. Then, it generates trajectories based on the hidden states correlated with home and work.

- *TimeGeo* [17]: This model generates the trajectories with the probabilistic model and parameters with explicit physical meanings, including the weekly home-based tour number, dwell rate, burst rate, and so on.
- *DeepMove* [9]: This model utilizes the attention mechanism in the standard RNN model to capture the periodical patterns in mobility trajectories.
- *GAN* [12]: This model utilizes the GAN model to directly fit the distribution of the real-world human trajectory data by using a generator network and a discriminator network.
- *SeqGAN* [49]: This model uses the policy gradient method to train the generative model of the sequence data, of which the reward signal is obtained based on a GAN discriminator.
- *MoveSim* [10]: This model is recently proposed to synthesize human trajectories based on GAN, which introduces prior knowledge and physical regularities to the SeqGAN model.

Specifically, IO-HMM and TimeGeo are representative probabilistic trajectory synthesizing models, while DeepMove, GAN, SeqGAN, and MoveSim are representative deep learning models, where MoveSim is a state-of-the-art GAN-based trajectory synthesizing model proposed in recent years. As for other state-of-the-art methods, SocialGAN [1], XGail [27], and TrajGail [52] mainly focus on microscopic human trajectories, which models the human mobility in terms of speeds, directions, road link transitions, and so on. Thus, they require trajectory datasets with a small sampling interval, e.g., from tens of microseconds [1] to tens of seconds [27, 52], while we mainly focus on macroscopic human trajectories with large sampling interval (e.g., tens of minutes in the Mobile and GeoLife dataset [54]). Thus, their performance cannot be compared in our experimental scenarios. Overall, by comparing these selected algorithms covering the most representative probabilistic models and deep learning models, the performance of our proposed algorithm can be credibly evaluated.

*Parameters Settings.* In our GeoGail model, we first embed the input before entering the network. The location embedding size is set as 64, while the time embedding, the staying time embedding, and the #Visited location is set as 8. As for the home feature, it uses the same embedding net as the location embedding. Inside the generator, we use the self-attention structure, the inside hidden vector (Q, K, and V) is set as 64. Finally, a linear followed with softmax activation is chosen as the output. In the Instant discriminator network  $D^I$ , the structure of  $D^I$  is three linear layers with ReLU activation and one linear layer with sigmoid activation. In the long-term discriminator network  $D^L$ , the numbers of filters are set as {100, 200, 200, 200, 200, 100, 100, 100, 100, 100, 160, 160} and filter sizes are set as {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20}. All the parameters are initialized with a uniform distribution  $[-0.05, 0.05]$ . The learning rate is set as 0.0003 for the pre-train process and 0.00003 for the training. Generator, instant discriminator network, and long-term discriminator network have the same learning rate. The hyper-parameter  $\lambda$  between Instant reward  $R^I$  and Long-term reward  $R^L$  is set as 0.1 throughout our experiment

## 5.2 Quantitative Results

Inspired by the common practice in previous works [10, 17, 26], we choose six different metrics to evaluate the quality of generated data in terms of its usability compared with real-world expert trajectories, which are defined as follows:

- *Distance*: The cumulative travel distance of different users in the fixed time interval  $\Delta t$ .
- *Radius*: Radius of gyration ( $r_g$ ) [11]. It represents the spatial range of users' daily movement.
- *Duration*: The stay duration of users at different visited locations.
- *DailyLoc*: The number of visited locations per day for each user.
- *Rank*: The visiting frequency of top-100 locations for each user.
- *Exploration*: The probability to visit a new location as a function of the number of visited locations.

Table 1. Performance of Different Algorithms in Terms of Key Usability Metrics on the Two Datasets

Dataset	Mobile Dataset						GeoLife Dataset					
Metrics (JSD)	Distance	Radius	Duration	DailyLoc	I-Rank	Exploration	Distance	Radius	Duration	DailyLoc	I-Rank	Exploration
IO-HMM	0.0473	0.4116	0.2390	0.1130	0.1682	0.2145	0.1389	0.3953	0.3024	0.4792	0.3734	0.3514
TimeGeo	0.1036	0.4492	0.3006	0.5922	0.2041	0.2207	0.1674	0.3522	0.3536	0.5652	0.3855	0.3350
DeepMove	0.1152	0.4512	0.3743	0.3375	0.3929	0.4042	0.2149	0.4316	0.2654	0.3025	0.3526	0.5329
GAN	0.6283	0.6829	0.2123	0.6214	0.2132	0.7023	0.2432	0.7039	0.4357	0.6841	0.4294	0.7992
SeqGAN	0.5072	0.6603	0.1131	0.6931	0.2378	0.8361	0.1704	0.6682	0.1436	0.6931	0.3289	0.8371
MoveSim	0.0145	0.0806	0.0041	0.0844	0.0816	0.2121	0.0686	0.2912	0.0330	0.6758	0.0106	0.0708
Ours	<b>0.0113</b>	<b>0.0116</b>	<b>0.0035</b>	<b>0.0360</b>	<b>0.0054</b>	<b>0.0573</b>	<b>0.0035</b>	<b>0.0055</b>	<b>0.0270</b>	<b>0.0410</b>	<b>0.0095</b>	<b>0.0302</b>

All metrics are evaluated based on JSD with lower values indicating better performance. In addition, the best result and the second-best result of each metric are marked with bold and underline, respectively.

All the above metrics are characterized by probability distributions with each trajectory or each mobility record as one sample. In order to quantify these metrics, we utilize the JSD, which is a distance defined between probability distributions. Specifically, for arbitrary probability distribution  $p$  and  $q$ , the JSD between them is represented by the following equation:

$$JSD(p; q) = h((p + q)/2) - (h(p) + h(q))/2,$$

where  $h(\cdot)$  is the Shannon's entropy, and a smaller JSD indicates a larger similarity between  $p$  and  $q$ .

As shown in Table 1, our approach achieves the lowest JSD in all the models, with MoveSim to be the second best performing approach. IO-HMM achieves the third-lowest JSD, suggesting that the feature of the home position is useful in trajectory synthesizing. As a probability-based model with explicit physical meaning, TimeGeo performs better than the neural network models including SeqGAN and GAN. Using the reinforcement learning method allows SeqGAN to follow closely behind. GAN produces worse results than TimeGeo, suggesting that roughly adopting the GANs cannot capture the complex spatial-temporal features inside the mobility trajectories.

### 5.3 Interpretable IL

Based on the semantic level, our proposed GeoGail significantly improves the interpretability compared with existing neural network models by extracting the knowledge contained in the mobility trajectories. To interpret what knowledge the model has learned, we have conducted several experiments in terms of three statistical features and three statistical relationships.

When facing human mobility, statistic models usually define a number of parameters with explicit physical meanings and then generate mobility trajectories based on these parameters. Inspired by them, we first evaluate whether these parameters with explicit physical meanings are well-learned based on our proposed model. Specifically, TimeGeo defines the following physical parameters:

- $N_w$ : The weekly home-based tour number, which represents the travel likelihood when a person is at home.
- $\beta_1$ : Burst rate, which measures how much more active (or likely to travel) the person is at another place compared with home. Specifically,  $N_w\beta_1$  represents the travel likelihood when a person is out of the home.
- $\beta_2$ : Dwell rate, which measures the activity for the person to do consecutive activities out of home. Specifically,  $N_w\beta_2$  is the likelihood of performing consecutive out-of-home activities.

In TimeGeo, the probability of the next action is modeled based on a simplified assumption, i.e., a multinomial distribution with the parameters related to  $N_w$ ,  $\beta_1$ , and  $\beta_2$  to preserve corresponding characteristics. However, human trajectories are much more complicated, and in our proposed model, it is modeled by a non-linear neural network with much stronger modeling ability.

Table 2. Comparison of Interpretable Statistical Features

Dataset	Mobile Dataset			GeoLife Dataset		
Parms	$N_w$	$\beta_1$	$\beta_2$	$N_w$	$\beta_1$	$\beta_2$
real	16.703	3.540	2.639	5.191	7.338	6.343
MoveSim	3.207	20.525	20.092	3.461	11.098	9.485
Ours	15.417	3.265	2.337	5.549	6.052	5.344

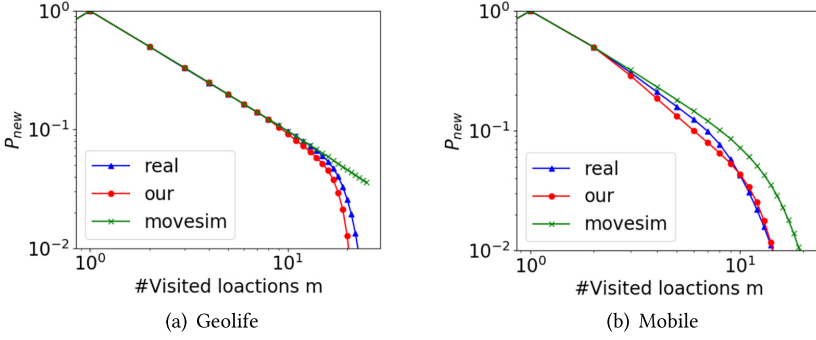


Fig. 6. Probability to visit a new location.

To interpret the effectiveness of our model, we would like to observe that our GeoGail can capture those physical parameters from expert trajectories in an unsupervised way, of which the results are shown in Table 2.

As we can observe in Table 2, MoveSim cannot capture these parameters. Specifically, MoveSim shows less sensitivity to the variety of datasets. That is,  $N_w$  of trajectories generated by MoveSim is almost unchanged in the two datasets, while a clear difference exists in the two real-world datasets. The reason is that MoveSim does not model the inherent decision-making process of human mobility, leading to its weakness in terms of modeling these physical parameters. Our proposed model shows a pleasing result compared with MoveSim. Specifically, our generated results show more sensitivity to those parameters, which fit the real data perfectly. Note that in our model, these parameters are not modeled in a supervised way, indicating our proposed model can learn these knowledge automatically.

As mentioned in Section 4.1, then we investigate whether we can learn the statistical relationships of decision-making features and users' movement behavior. Specifically, we show the statistical relationships between the probability to visit a new location  $P_{new}$  and the number of visited locations  $m$  in Figure 6, and the probability density function of the staying time  $P(\Delta t)$  in Figure 7. From Figure 6, we can observe that both MoveSim and our model can fit well at the beginning, but when it comes to the tail distribution, MoveSim performs much worse. Similarly, in Figure 7, we can observe that our proposed model outperforms MoveSim. More results are shown in Appendix A.5.

#### 5.4 Mobility Prediction with Augmented Datasets

In order to evaluate whether trajectories synthesized by our proposed model can effectively support practical applications, we select a representative application based on trajectory data, i.e., mobility prediction, to testify the utility of our proposed model. Specifically, in our experiment, we utilize a standard **Long Short-Term Memory (LSTM)** network to predict users' further movements based on their historical trajectories. We consider the scenario where only a limited number of

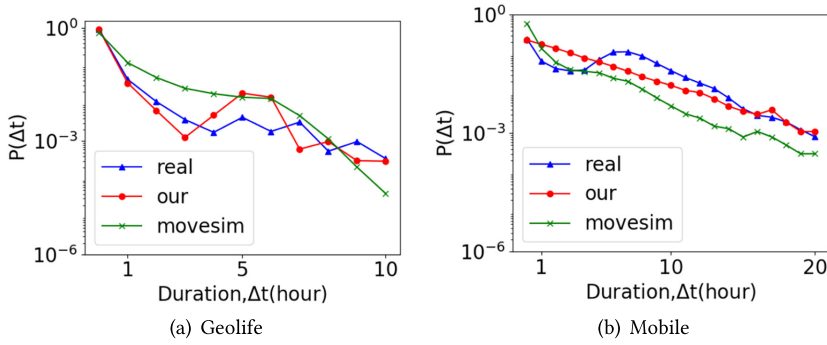
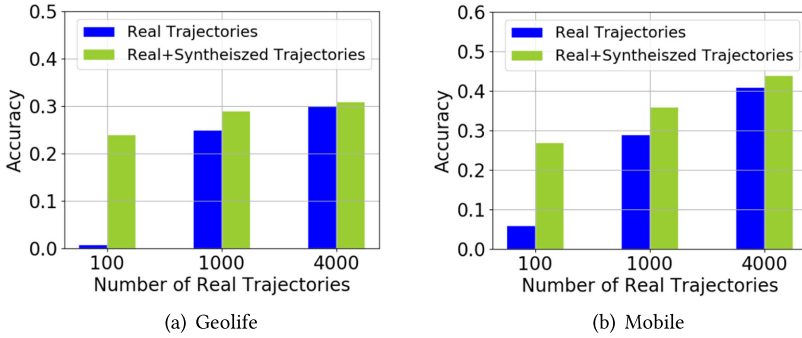
Fig. 7. Probability density function of the staying time  $\Delta t$ .

Fig. 8. Mobility prediction based on synthetic trajectories.

real-world human trajectories are available due to the collection cost or privacy concerns. Then, the synthesized trajectories are used to augment the trajectory data as the training samples to train the mobility prediction model. Finally, we evaluate the performance of the mobility prediction model on real-world trajectories to show whether the synthesized trajectory can effectively support the mobility prediction task. Specifically, we compare the prediction accuracy of the model trained on the augmented dataset compared with that only trained on the real-world trajectories, and we also vary the number of available real-world trajectories to have a more comprehensive evaluation, of which the results are shown in Figure 8. As we can observe, the mobility prediction model trained on the augmented dataset significantly outperforms that only trained on real-world trajectories, especially when the number of real trajectories is small (e.g., 100 trajectories), which proves the usability of the synthesized trajectories.

### 5.5 Ablation Study

We conduct a series of ablation experiments to demonstrate that our proposed components in GeoGail are indeed crucial for learning an effective policy. In our ablation experiments, we selectively remove some of the proposed components. *Our* includes all the components; *Ours-SA* replaces the self-attention network with a pure linear layers; *Ours-Pretrain* removes the pre-training mechanism. Table 3 shows their average JSD. We can observe that our proposed method performs the best with the help of the extracted decision-making features; the policy without the long-term discriminator network performs slightly worse. In addition, removing the pre-training mechanism will significantly increase the JSD, indicating the importance of the pre-training mechanism.

Table 3. The Ablation Study of Our Proposed Framework on Two Mobility Datasets

Dataset	Mobile Dataset						GeoLife Dataset					
Metrics (JSD)	Distance	Radius	Duration	DailyLoc	I-Rank	Exploration	Distance	Radius	Duration	DailyLoc	I-Rank	Exploration
Ours	<b>0.0113</b>	<b>0.0116</b>	<b>0.0035</b>	<b>0.0360</b>	<b>0.0054</b>	<b>0.0573</b>	<b>0.0035</b>	<b>0.0055</b>	<b>0.0270</b>	<b>0.0410</b>	<b>0.0095</b>	<b>0.0302</b>
Ours-SA	0.0178	0.0248	0.0235	0.0364	0.0070	0.0619	0.0040	0.1024	0.0560	0.1982	0.0111	0.0570
Ours-Pretrain	0.0185	0.1179	0.1191	0.6931	0.5158	0.2688	0.0310	0.3001	0.1063	0.5001	0.0142	0.2203

Bold denotes the best (lowest) results.

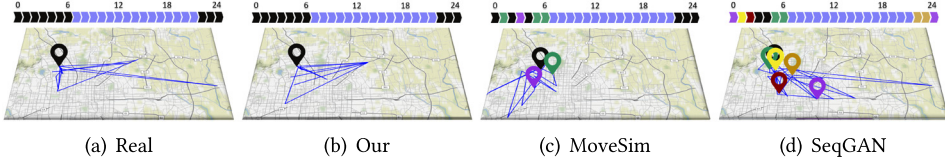


Fig. 9. Individual trajectory of one day, above with the time bar, where the time slices with different colors represent staying at different locations.

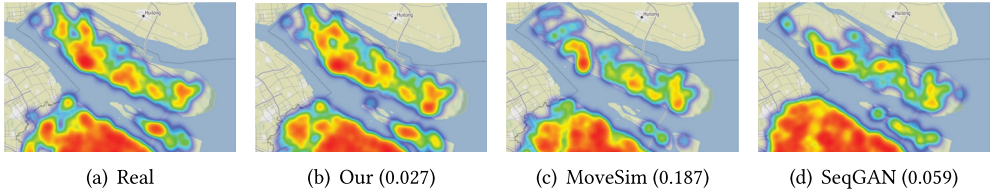


Fig. 10. Heat map of population density at 2 a.m., with the JSD compared with the real distribution in the brackets.

## 5.6 Qualitative Results

First, we show the visualization of a single trajectory in Figure 9. We can observe that SeqGAN fails to figure out the spatio-temporal information from the training data. Specifically, the generated trajectory is disorderly and complicated. MoveSim produces better results with higher regularity. However, we can still observe that the trajectory synthesized by MoveSim will go to a series of different locations at night, rather than staying at one place (i.e., his home) as the expert trajectory and the trajectory synthesized by our proposed model. Overall, compared with baselines, our proposed model can well capture the mobility behavior of users in terms of their sleep schedule.

Then, We show the heat map of population density at 2 a.m. of the Mobile dataset and trajectories synthesized by different models in Figure 10, where the JSD is calculated by comparing with the real-world population density. We choose the time of 2 a.m. since at this time users are supposed to stay at home for sleeping. As shown in Figure 10, we have the smallest JSD among all models. The GAN-based approaches, including MoveSim and SeqGAN, do not well capture the sleep schedule of users, leading to their underperformance.

## 6 Related Work

*Human Trajectory Synthesizing.* Due to the vital role of human mobility trajectory synthesizing in the network performance simulation [13], urban planning [23], and so on, the trajectory synthesizing model has been widely studied. Early methods model the human mobility process based on statistic models or probabilistic models [6, 16, 17, 29, 48, 53]. However, these models rely on simplified



assumptions of human mobility behaviors, leading to their weakness in modeling the complicated relation between multi-dimensional spatio-temporal trajectories. The rise of the deep learning paradigm provides promising solutions to this problem. In the early attempts, existing approaches utilize the standard RNN to model and synthesize human trajectories in an autoregressive manner [20, 37]. However, these models suffer from *exposure bias* when generating long-term human trajectories. Simultaneously, GAN has been exploited with significant success in natural image generation, generating natural language, and so on. Existing studies have used GAN [47] to synthesize human trajectories. And researchers have further exploration in CNN [10, 26, 36] and RNN [18, 22]. However, these GAN-based methods are designed to directly learn the data distribution from demonstrations without modeling the hidden decision-making process. Different from them, our proposed method aims to synthesize trajectories by modeling the hidden decision-making process of human mobility based on IL techniques, which can capture the hidden semantics and inherent stochasticity of human movement. Furthermore, our proposed method can automatically learn explainable knowledge from data, e.g., statistical relation between semantic actions and decision-making features.

*IL.* IL techniques are designed for learning the policy function of the agent, which maps the current states to the actions to be implemented, from the given demonstrations. Early methods solve this problem by directly copying the behavior of the demonstrations based on standard supervised learning methods [3], which perform poorly on unseen situations and thus suffer from variations in the tasks. IRL is then proposed to solve this problem by recovering the agent's reward function based on demonstrations and trains the policy function by maximizing the total reward at the same time. A number of IRL methods are proposed based on the linear reward function of manually extracted features [7, 55, 56]. Recently, GAIL is proposed to utilize non-linear neural network-based reward function and policy function, which has shown great success in solving the IL problem. A large number of IL-based applications have also been proposed, including dynamic treatment regimes [41], traffic signal control [46], and explanationable analysis for the humans' behavior [25, 27, 42, 51, 52], and so on. Different from them, our article mainly focuses on the human trajectory synthesizing problem. Furthermore, different from the traditional action space based on the speed and direction in existing methods, we propose a semantic level in our system to model the movement behavior to diverse locations in the semantic domain, which can capture the more essential decision-making purpose of users and process diverse locations in a uniform manner.

## 7 Conclusion

In this article, we present GeoGail, an IL-based method to synthesize human trajectories with explicitly modeling the decision-making process combined with the semantics of the mobility behavior. In addition, we propose a self-attention-based policy net with elaborately selected decision-making features and a multi-scale reward net with the instant reward, trajectory-level reward, and mutual information-based reward in terms of individual characteristics. Extensive experimental results show that our proposed method significantly outperforms state-of-the-art methods. Furthermore, important explainable knowledge is automatically learned from the data, including the weekly home-based tour number, dwell rate, burst rate, probability density function of the stay duration, and statistical relation between the exploration probability and the number of visited locations, demonstrating the benefits of modeling the underlying decision-making process of human mobility. As future work, we will consider more complicated action places and state transition models by incorporating fine-grained semantics-based actions and individual characteristics, e.g., personalized sensitivity for distance.

## References

- [1] Li Fei-Fei, Silvio Savarese, Alexandre Alahi, Agrim Gupta, and Justin Johnson. 2018. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [2] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (GANs). In *Proceedings of the 34th International Conference on Machine Learning*.
- [3] Michael Bain and Claude Sammut. 1995. A framework for behavioural cloning. In *Machine Intelligence 15*. 103–129.
- [4] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R. James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J. Ramasco, Filippo Simini, and Marcello Tomasini. 2018. Human mobility: Models and applications. *Physics Reports* 734 (2018), 1–74.
- [5] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. 2019. Seeing what a GAN cannot generate. In *Proceedings of the International Conference on Computer Vision*.
- [6] Vincent Bindschadler and Reza Shokri. 2016. Synthesizing plausible privacy-preserving location traces. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy*.
- [7] Abdeslam Boularias, Jens Kober, and Jan Peters. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 182–189.
- [8] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [9] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*.
- [10] Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. 2020. Learning to simulate human mobility. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- [11] M. C. González, C. A. Hidalgo, and A. L. Barabási. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196 (2008), 779.
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the Advances in Neural Information Processing Systems*.
- [13] Andrea Hess, Karin Anna Hummel, Wilfried N. Gansterer, and Günter Haring. 2016. Data-driven human mobility modeling: A survey and engineering guidance for mobile networking. *ACM Computing Surveys* (2016).
- [14] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Proceedings of the Advances in Neural Information Processing Systems*.
- [15] Tianran Hu, Yinglong Xia, and Jiebo Luo. 2019. To return or to explore: Modelling human mobility and dynamics in cyberspace. In *The World Wide Web Conference*, 705–716.
- [16] Sibren Isaacman, Richard Becker, Ramon Caceres, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. 2012. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, 239–252.
- [17] Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. González. 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences* 113, 37 (2016).
- [18] Vaibhav Kulkarni, Natasha Tagasovska, Thibault Vatter, and Benoit Garbinato. 2018. Generative models for simulating mobility trajectories. arXiv:1811.12801. Retrieved from <https://arxiv.org/abs/1811.12801>
- [19] Jiaming Song, Li Yunzhu, and S. Ermon. 2017. InfoGAIL: Interpretable imitation learning from visual demonstrations. In *Proceedings of the 31st Conference on Neural Information Processing Systems*.
- [20] Ziheng Lin, Mogeng Yin, Sidney Feygin, Madeleine Sheehan, Jean-Francois Paiement, and Alexei Pozdnoukhov. 2017. Deep generative models of urban mobility.
- [21] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.
- [22] X. Liu, Hanzhou Chen, and Clio Andris. 2018. trajGANs: Using generative adversarial networks for geo-privacy protection of trajectory data (Vision paper). In *Location Privacy and Security Workshop*.
- [23] Yanchi Liu, Chuanren Liu, Nicholas Jing Yuan, Lian Duan, Yanjie Fu, Hui Xiong, Songhua Xu, and Junjie Wu. 2014. Exploiting heterogeneous human mobility patterns for intelligent bus routing. In *Proceedings of the 2014 IEEE International Conference on Data Mining*.
- [24] Tze Meng Low, Yuejie Chi, James Hoe, Swarun Kumar, Akarsh Prabhakara, Laixi Shi, Upasana Sridhar, Nikolai Tukanov, Chengyue Wang, and Yuchen Wu. 2022. Zoom out: Abstractions for efficient radar algorithms on COTS architectures. In *Proceedings of the 2022 IEEE International Symposium on Phased Array Systems & Technology (PAST '22)*, 1–6. DOI: <https://doi.org/10.1109/PAST49659.2022.9975097>

- [25] Yanhua Li, Xun Zhou, Zhenming Liu, Jie Bao, Yu Zheng, Jun Luo, Menghai Pan, and Weixiao Huang. 2020. Is reinforcement learning the choice of human learners? A case study of taxi drivers. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*.
- [26] Kun Ouyang, Reza Shokri, David S. Rosenblum, and Wenzhuo Yang. 2018. A non-parametric generative model for human trajectories. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 3812–3817.
- [27] Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, and Jun Luo. 2020. xGAIL: Explainable generative adversarial imitation learning for explainable human decision analysis. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- [28] Luca Pappalardo, Salvatore Rinzivillo, and Filippo Simini. 2016. Human mobility modelling: Exploration and preferential return meet the gravity model. *Procedia Computer Science* 83 (2016), 934–939.
- [29] Luca Pappalardo and Filippo Simini. 2017. Data-driven generation of spatio-temporal routines in human mobility. *Data Mining and Knowledge Discovery* 32, 1 (2017).
- [30] Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-László Barabási. 2015. Returners and explorers dichotomy in human mobility. *Nature Communications* 6, 1 (2015), 1–8.
- [31] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2016. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proceedings of Computer Vision (ECCV '16)*, 549–565.
- [32] S. Ross and D. Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*.
- [33] G. J. Gordon S. Ross and D. Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 6.
- [34] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. 2010. Modelling the scaling properties of human mobility. *Nature Physics* 6, 10 (2010), 818–823.
- [35] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albertlaszlo Barabasi. 2010. Limits of predictability in human mobility. *Science* 327, 5968 (2010), 1018–1021.
- [36] Ha Yoon Song, Moo Sang Baek, and Minsuk Sung. 2019. Generating human mobility route based on generative adversarial network. In *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*.
- [37] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. 2016. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*.
- [39] Huandong Wang, Changzheng Gao, Yuchen Wu, Depeng Jin, Lina Yao, and Yong Li. 2023. PateGail: A privacy-preserving mobility trajectory generator with imitation learning. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence and 35th Conference on Innovative Applications of Artificial Intelligence and 13th Symposium on Educational Advances in Artificial Intelligence (AAAI '23/IAAI '23/EAAI '23)*. AAAI Press, Article 1631, 9 pages. DOI: <https://doi.org/10.1609/aaai.v37i12.26700>
- [40] Huandong Wang, Qizhong Zhang, Yuchen Wu, Depeng Jin, Xing Wang, Lin Zhu, and Li Yu. 2024. Synthesizing human trajectories based on variational point processes. *IEEE Transactions on Knowledge and Data Engineering* 36, 4 (2024), 1785–1799. DOI: <https://doi.org/10.1109/TKDE.2023.3312209>
- [41] Lu Wang, Wenchao Yu, Xiaofeng He, Wei Cheng, and Hongyuan Zha. 2020. Adversarial cooperative imitation learning for dynamic treatment regimes. In *Proceedings of the Web Conference*.
- [42] Guojun Wu, Yanhua Li, Shikai Luo, Ge Song, Qichao Wang, Jing He, Jieping Ye, Xiaohu Qie, and Hongtu Zhu. 2020. A joint inverse reinforcement learning and deep learning model for drivers' behavioral prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.
- [43] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. 2017. Modeling trajectories with recurrent neural networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*.
- [44] Yuchen Wu, Huandong Wang, Zhiwei Xu, Depeng Jin, and Yong Li. [n.d.]. GeoGail: A model-based imitation learning framework for human trajectory synthesizing. ([n.d.]).
- [45] Yuchen Wu, Runtong Zhang, and Keiji Yanai. 2020. Attention guided unsupervised image-to-image translation with progressively growing strategy. In *Pattern Recognition*. Michael Cree, Fay Huang, Junsong Yuan, and Wei Qi Yan (Eds.). Springer Singapore, Singapore, 85–99.
- [46] Yuanhao Xiong, Guanjie Zheng, Kai Xu, and Zhenhui Li. 2019. Learning traffic signal control from demonstrations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.
- [47] Dan Yin and Qing Yang. 2018. GANs based density distribution privacy-preservation on mobility data. In *Security and Communication Networks*.

- [48] Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-François Paiement, and Alexei Pozdnoukhov. 2017. A generative model of urban activities from cellular data. *IEEE Transactions on Intelligent Transportation Systems* 19, 6 (2017).
- [49] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.
- [50] Runtong Zhang, Yuchen Wu, and Keiji Yanai. 2020. Pre-trained and shared encoder in cycle-consistent adversarial networks to improve image quality. In *Pattern Recognition*. Shivakumara Palaiahnakote, Gabriella Sanniti di Baja, Liang Wang, and Wei Qi Yan (Eds.), Springer International Publishing, Cham, 312–325.
- [51] Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. 2019. Unveiling taxi drivers' strategies via cGAIL: Conditional generative adversarial imitation learning. In *Proceedings of the 2019 IEEE International Conference on Data Mining*.
- [52] Xin Zhang, Yanhua Li, Xun Zhou, Ziming Zhang, and Jun Luo. 2020. TrajGAIL: Trajectory generative adversarial imitation learning for long-term decision analysis. In *Proceedings of the 2020 IEEE International Conference on Data Mining*.
- [53] Ping Zhao, Hongbo Jiang, Jie Li, Fanzi Zeng, and Guanglin Zhang. 2019. Synthesizing privacy preserving traces: Enhancing plausibility with social networks. *IEEE/ACM Transactions on Networking*, 99 (2019), 1–14.
- [54] Yu Zheng, Xing Xie, Wei-Ying Ma. 2010. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin* 33, 2 (2010), 32–39.
- [55] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. 2010. Modeling interaction via the principle of maximum causal entropy. In *Proceedings of the 27th International Conference on Machine Learning*.
- [56] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.

## Appendix

### A Appendix for Reproducibility

#### A.1 Pseudo-Code of GeoGail

The whole training procedure of our GeoGail can be summarized in the following Algorithm A1. As we can observe, this algorithm starts by randomly initializing weight of  $\pi$ ,  $D^I$ , and  $D^L$ . Then, it iteratively sample the home position for the agent and generate the corresponding synthetic trajectory. By sampling expert trajectories with the same size, this algorithm first update  $D^I$  and  $D^L$  based on Equation (16) and then update  $\pi$  via PPO update rule based on the reward  $R^M$  shown in Equation (7).

---

#### Algorithm A1: GeoGAIL

---

**Input:** Expert trajectories  $T_E \sim \pi_E$ .

- 1: Initialize  $\pi$ ,  $D^I$ ,  $D^L$  with random weights;
  - 2: Pre-train  $\pi$ ,  $D^I$ ,  $D^L$  based on Equations (13)–(15), respectively;
  - 3: **for**  $i=0,1,2,\dots$  **do**
  - 4:   Sample a batch of home positions  $h_i \sim p(h)$  for the agents;
  - 5:   Sample a batch of trajectories  $T_i \sim \pi(h_i)$ ;
  - 6:   Sample expert trajectories  $T_E \sim \pi_E$  with same batch size;
  - 7:   Update  $D^I$  and  $D^L$  based on Equation (16);
  - 8:   Update  $\pi$  via PPO update rule with the reward  $R^M$  in Equation (7);
  - 9: **end for**
- 

#### A.2 Detail for Processing Our Data

##### A.2.1 Data Selection.

— *Mobile dataset* was collected in Shanghai by using a major mobile network operator. It records the anonymous user ID, accessed base station, and timestamp of each accessing.

Table A1. Statistic Information of Two Mobility Datasets

Datasets	Mobile Dataset	GeoLife Dataset
Duration	1 April to 7 April 2016	April 2007 to August 2012
#Users	100,000	182
#Records/user	261	453
#Locations	9,000 base stations	GPS coordinates

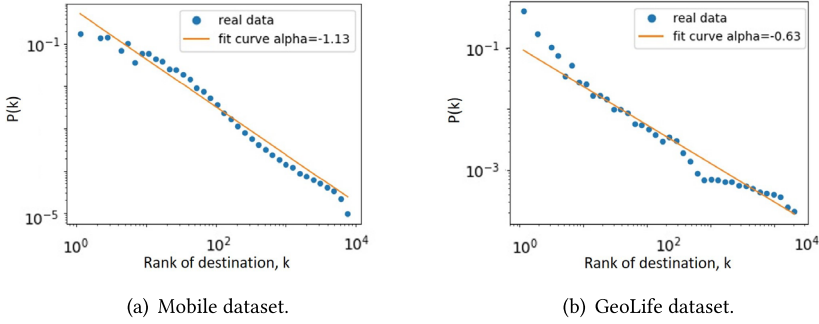


Fig. A1. Empirical probability.

— *GeoLife dataset* was collected in (Microsoft Research Asia) Geolife project by 178 users in a period of over 4 years (from April 2007 to October 2011). A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude and altitude. This dataset contains 17,621 trajectories with a total distance of 1,251,654 kilometers and a total duration of 48,203 hours. we project GPS coordinates into the grids by containing up to three digits after the decimal point.

**A.2.2 Home Position Selection.** Home position is crucial in our GeoGail model. To extract the home position in each dataset, we label the most frequently visited stay-region during the nights (between 7 p.m. of the first day and 8 a.m. of the second day) as the home stay-region.

**A.2.3 Alpha Selection.** When facing the explore action, the exploring location is connected with  $P(l_{i+1} = l | s_i, a_i = \text{Explore}) \propto k(l, l_i)^{-\alpha}$ . For an individual to select an exploration destination, we measure  $P(l_{i+1} = l | s_i, a_i = \text{Explore})$  aggregating all users' destinations. The empirical probability of choosing the rank  $k$  location as a trip destination in each data can be calculated in Figure A1.

### A.3 Network Architecture of GeoGail

In our GeoGail model, we first embed the input before entering the network. The location embedding size is set as 64, while the time embedding, the staying time embedding, and the #Visited location is set as 8. As for the home feature, it uses the same embedding net as the location embedding. Inside the generator, we use the self-attention structure, the inside hidden vector (Q, K, and V) is set as 64. Finally, a linear followed with softmax activation chose as the output. In the instant discriminator network  $D^I$ , the structure of  $D^I$  is three linear layers with ReLU activation and one linear layer with sigmoid activation. In the long-term discriminator network  $D^L$ , the numbers of filters are set as {100, 200, 200, 200, 200, 100, 100, 100, 100, 100, 160, 160} and filter sizes are set as {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20}. All the parameters are initialized with the uniform distribution

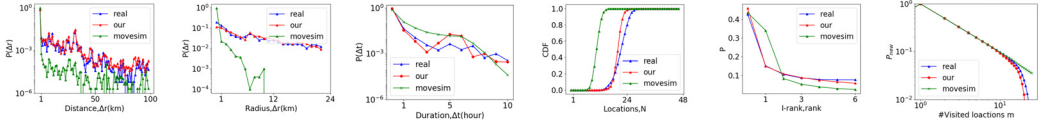


Fig. A2. The comparison between mobility patterns in generated trajectory data (our framework and MoveSim) and real trajectory from GeoLife dataset.

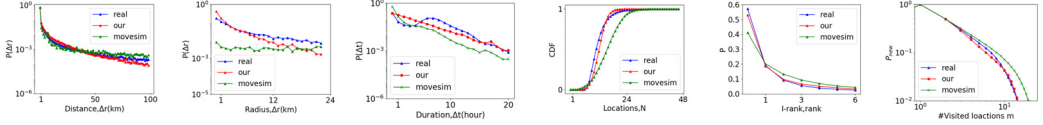


Fig. A3. The comparison between mobility patterns in generated trajectory data (our framework and MoveSim) and real trajectory from Mobile dataset, where  $P$  denotes the probability distribution.

$[-0.05, 0.05]$ . The learning rate is set as 0.0003 for the pre-train process and 0.00003 for the training. Generator, instant discriminator network, and long-term discriminator network have the same learning rate. The hyper-parameter  $\lambda$  between Instant reward  $R^I$  and Long-term reward  $R^L$  is set as 0.1 throughout our experiment

#### A.4 Parameters of Baselines

- *IO-HMM*: We label the most frequently visited stay-region during the nights (between 7 p.m. of the first day and 8 a.m. of the second day) as the home place. For activity labels, we obey the rules in the original article. We use fully unsupervised IO-HMM as a generative model during the sequence generations
- *TimeGeo*: The home place was selected in the same way as the IO-HMM. The weekly home-based tour number set as 16.7 for Mobile dataset and 5.2 for GeoLife dataset. Burst rate is 3.5 for Mobile and 7.3 for GeoLife dataset. Dwell rate is 2.6 for Mobile and 6.3 for GeoLife dataset.
- *DeepMove*: First, based on the population density, we choose a start point to predict its next trajectory point. Then combining input and output into a new trajectory and predict the next point. Repeat this process until complete the whole reproducing.
- *GAN*: We normalize the original GPS coordinates of location into  $[0,1] \times [0,1]$  two sequences and feed them into the network. LSTM is chosen inside the generator, where the hidden size is set as 64. We build the discriminator the same as our long-term discriminator network. The learning rate is set as 0.001 for both the generator and discriminator.
- *SeqGAN*: We use LSTM in the generator where the embedding size for one-hot location is 32 and the hidden size is 64. We build the discriminator the same as our long-term discriminator network. The learning rate is set as 0.001 for both generator and discriminator.
- *MoveSim*: The location embedding size is set as 64, while the time embedding is set as 16. We build the discriminator the same as our long-term discriminator network. The learning rate is set as 0.001 for both the generator and discriminator. The default weight of mobility regularity-aware loss is set as 1.0.

#### A.5 Extension of Table 1: Mobility Pattern Distribution Comparison

As the extension of Table 1, we compare the distribution of our framework, the best baseline (MoveSim), and real data directly on these metrics in Figures A2 and A3. Presented in the mobility pattern probability distribution of GeoLife dataset in Figure A2, we can find that the distribution of



our framework (red line) has a better result than the real data (green line) distribution, which is much closer to the real data. Besides, we also compare in Figure A3. By comparing with the best baseline MoveSim (green line), the mobility regularity distribution of our framework (red line) has a better result than the real data of the Mobile dataset (blue line) on all the metrics.

Received 6 June 2022; revised 6 June 2022; accepted 15 December 2023