Activity Trajectory Generation via Modeling Spatiotemporal Dynamics

Yuan Yuan
Department of Electronic Engineering
Tsinghua University
Beijing, China

Jingtao Ding*
Department of Electronic Engineering
Tsinghua University
Beijing, China

Huandong Wang
Department of Electronic Engineering
Tsinghua University
Beijing, China

Depeng Jin
Department of Electronic Engineering
Tsinghua University
Beijing, China

Yong Li
Department of Electronic Engineering
Tsinghua University
Beijing, China

ABSTRACT

Human daily activities, such as working, eating out, and traveling, play an essential role in contact tracing and modeling the diffusion patterns of the COVID-19 pandemic. However, individual-level activity data collected from real scenarios are highly limited due to privacy issues and commercial concerns. In this paper, we present a novel framework based on generative adversarial imitation learning, to generate artificial activity trajectories that retain both the fidelity and utility of the real-world data. To tackle the inherent randomness and sparsity of irregular-sampled activities, we innovatively capture the spatiotemporal dynamics underlying trajectories by leveraging neural differential equations. We incorporate the dynamics of continuous flow between consecutive activities and instantaneous updates at observed activity points in temporal evolution and spatial transformation. Extensive experiments on two real-world datasets show that our proposed framework achieves superior performance over state-of-the-art baselines in terms of improving the data fidelity and data utility in facilitating practical applications. Moreover, we apply the synthetic data to model the COVID-19 spreading, and it achieves better performance by reducing the simulation MAPE over the baseline by more than 50%. The source code is available online: https://github.com/tsinghua-fib-lab/Activity-Trajectory-Generation.

CCS CONCEPTS

• **Computing methodologies** → **Modeling and simulation**; **Simulation types and techniques**; *Continuous simulation*;

KEYWORDS

Activity Simulation, Spatiotemporal dynamics, GAIL

*Jingtao Ding is the corresponding author (dingjt15@tsinghua.org.cn).

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3542671

ACM Reference Format:
Yuan Yuan, Jingtao Ding*, Huandong Wang, Depeng Jin, and Yong Li. 2022. Activity Trajectory Generation via Modeling Spatiotemporal Dynamics. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3542671

1 INTRODUCTION

The continuous spread of COVID-19 with increasing infectivity (like Omicron Virus) has possessed an intractable challenge of pandemic prevention and control [46]. As COVID-19 is transmitted mainly by person-to-person contacts, effective contact tracing is quite essential and governments are taking measures to keep close contacts in quarantine to prevent the pandemic from spreading. Activity trajectories record individuals' participation in various location-based activities, such as traveling, working, and eating out, and thus play a fundamental role in the pandemic modeling and control. On the one hand, the complex spatiotemporal associations underlying daily activities can give rise to recessive infection that is difficult to deal with. On the other hand, compared with general mobility trajectories, the semantic information, i.e., activity type, is provided in the activity trajectory, which is more useful for tracing close contacts and precision epidemiology [25, 35].

Despite the great value of individual activity trajectories in pandemic control, such data are highly limited in applications due to privacy issues and commercial concerns [23]. Meanwhile, the direct replay of real-world data has limitations in supporting advanced modeling [9], e.g., counterfactual scenarios that what will happen if people reduce the use of public transportation. Therefore, realistic simulation of individual-level daily activities to generate massive high-quality activity trajectories becomes an essential and meaningful research problem, which covers the deficiency of real-world data in modeling the pandemic spread and facilitating rational policymaking.

To effectively model the diffusion patterns of the pandemic, it is necessary to simulate fine-grained activities (i.e., what activity to take at what place and time) of massive individuals in daily life. Though many works have studied [8, 9, 19] the problem of mobility prediction/simulation, they mainly focus on the regularities and variations of spatial movements without considering concrete activity types, which however are important for modeling the pandemic spread. Moreover, due to the inherent randomness and sparsity of

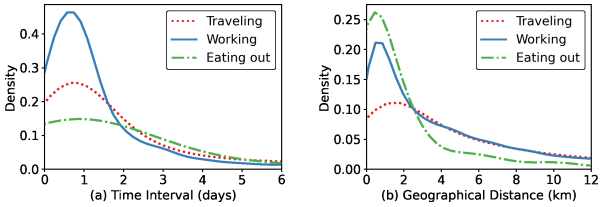


Figure 1: Spatiotemporal distributions of a real-world dataset. (a) interval distributions of repeat actions; (b) distance distributions between the activity location and home.

irregular-sampled activities and higher uncertainties induced by larger intervals, existing methods, such as RNN-based [8, 9] and Markov-based approach [2, 53], are insufficient for modeling real-world activity trajectories with the discrete-time or time-invariant assumption.

Figure 1 illustrates the results of preliminary analyses on a real-world activity dataset. As we can observe in Figure 1(a), different activities reveal diverse time interval distributions between the two observations. For example, going to work is relatively more regular than eating out. Besides, though researchers [45] have proved that the overall distribution of spatial movements is fat-tailed that can be described by limited parameters, Figure 1(b) delivers obviously distinct spatial patterns, where traveling accepts a longer geographical distance. The trajectory appears to be just a sequence of activity transitions, however, it is the underlying spatiotemporal dynamics that intrinsically drive these activity points and exhibit distinct temporal and spatial distributions. Therefore, to better match reality, it is necessary to inherently take the latent dynamics into consideration and capture them in a principle way.

In this paper, we present a framework based on Generative Adversarial Imitation Learning (GAIL) [15] to generate artificial activity trajectories. To effectively capture complex transition patterns with spatiotemporal dependency, we propose to model continuous-time spatiotemporal dynamics underlying the observed activities. By characterizing trajectories as point processes, we design the policy function with a neural spatiotemporal point process model, where the evolution of spatiotemporal dynamics is captured by the hidden states. Specifically, we jointly incorporate the dynamics of continuous flow between consecutive activities and instantaneous updates at observed activity points in temporal evolution and spatial transformation. To tackle the challenge that irregular-sampled activities are inherently random and sparse, we model the dynamic evolution in continuous time and space by leveraging neural differential equations [4]. Finally, based on the learned dynamics, the activity with time and location is generated sequentially. In summary, our main contributions are as follows:

- To the best of our knowledge, we are the first to capture underlying spatiotemporal dynamics via neural differential equations for modeling activity trajectories.
- We propose a novel activity simulation framework based on GAIL, which models the activity decision as a spatio-temporal point process and designs an activity decision model informed by the learned spatiotemporal dynamics in continuous domain.
- Extensive experiments on two real-world datasets show that our proposed framework outperforms state-of-the-art baselines in generating activity trajectories with retained fidelity. Besides, we validate the utility of the generated data in supporting practical applications, i.e., activity prediction. Furthermore, we demonstrate the effectiveness of the synthetic data in modeling and simulating the COVID-19 spreading.

2 PRELIMINARIES

2.1 Problem Definition

In this section, we formally define our research problem of learning to generate activity trajectories. It is equal to generating activity sequences with the timestamp and spatial locations. We first define the location-based activity and activity trajectory, and then formulate our research problem.

Definition 1. ***Location-based Activity.*** *An activity-based location is described by a four-element tuple $\tau_i = (t_i, k_i, lat_i, lon_i)$, where t_i is the sampled timestamp, k_i is the activity type, lat_i and lon_i are latitude-longitude coordinations.*

Definition 2. ***Activity Trajectory.*** *An activity trajectory is a sequence of location-based activities generated by an individual in daily life. It is represented by a sequence of chronologically ordered points $S = [\tau_1, ..., \tau_n]$, where τ_i represents a location-based activity.*

Problem 1. *Given a real-world activity trajectory dataset, our goal is learning to simulate individuals' decision-making process to generate artificial activity trajectories while retraining the fidelity and utility of the real-world data.*

2.2 Background

Spatial-temporal Point Process. STPP [6] models a sequence of stochastic events occurred in continuous space and time. In summary, it is characterized by a non-negative *intensity function* given the history of events up to time t, $\mathcal{H}_t = \{(t_i, x_i)|t_i \leq t\}$:

$$\lambda(t, \mathbf{x}|\mathcal{H}_t) = \lim_{\Delta t \to 0, \Delta \mathbf{x} \to 0} \frac{\mathbb{P}(t_i \in [t, t+\Delta t], x_i \in B(\mathbf{x}, \Delta \mathbf{x})|\mathcal{H}_t)}{|B(\mathbf{x}, \Delta \mathbf{x})|\Delta t}, \quad (1)$$

where $B(\mathbf{x}, \Delta \mathbf{x})$ denotes an infinitesimal spatial ball centered at location $\mathbf{x} \in \mathbb{R}^d$ with radius $\Delta \mathbf{x}$. The conditional intensity function $\lambda(t, \mathbf{x}|\mathcal{H}_t)$ represents the probability of the event occurring in the time interval $(t, t+\Delta t]$ and locating in the spatial ball $S = B(\mathbf{x}, \Delta \mathbf{x})$. $\lambda^*(t, \mathbf{x}) = \lambda(t, \mathbf{x}|\mathcal{H}_t)$ is used to represent the conditional intensity where $*$ denotes the dependence on the history $\mathcal{H}_t$. Given the history of N events $\mathcal{H}$ within a time interval $[0, T]$, the joint log-likelihood of observing $\mathcal{H}$ is given as follows:

$$\log p(\mathcal{H}) = \sum_{i=1}^{N} \log \lambda^*(t_i, \mathbf{x}_i) - \int_0^T \int_{\mathbb{R}^d} \lambda^*(\tau, \mathbf{x}) d\mathbf{x} d\tau. \quad (2)$$

The conditional intensity function $\lambda^*(t, \mathbf{x})$ plays a critical role in the realization of the STPP, which decides the occurring time and spatial location of the event.

Generative Adversarial Imitation Learning. GAIL [15] is derived from an alternative approach of imitation learning and formulates the problem of learning a policy function π_θ to generate expert-like action conditioned on the state. In the meantime, a discriminator $D\phi$ is trained to distinguish between policy-generated and real-world state-action pairs, and further rewards the policy

function for confusing itself. The objective of GAIL is to solve a minimax problem as follows:

$$\max_{\phi} \min_{\theta} -\lambda H(\pi_\theta) + \mathbb{E}_{\pi}[\log D_\phi(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D_\phi(s,a))], \tag{3}$$

where $\mathbb{E}_\pi$ represents the expected reward of the sequences under the policy π, and π_E represents the policy under the expert sequences. $H(\pi_\theta)$ is an entropy regularization term, which controls to find the policy π with maximum causal entropy.

Our proposed framework is based on the basic GAIL framework. In addition to the policy and reward functions that have to be optimized, we also innovatively learn the spatiotemporal dynamics to better characterize the decision process of the activity choice.

2.3 Sequential Decision Processes

The activity trajectory is generated by an individual's sequential decisions on what activity to perform at what time and location. The activity choice is affected by the individual's historical decisions. Therefore, we depict an individual's activity choice as a Markov Decision Process (MDP), which can be described by a 5-element tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{G}, \mathcal{R} >$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ represents the state transition, $\mathcal{G}$ denotes the spatiotemporal dynamics underlying state transitions, and $\mathcal{R}$ is the reward function. The basic elements of the MDP are defined as follows:

- **State**: It is defined as the history of activities $s_t = [a_1, ..., a_i]_{t_i<t}$, where $a_i = (t_i, k_i, lat_i, lon_i)$ includes the information of the timestamp, activity type, and latitude-longitude coordinations.
- **Action**: It is defined as the activity selected each step in the trajectory, which can be described as $a_i = (t_i, k_i, lat_i, lon_i)$.
- **State transition**: After the selected action, the state s_t will be update to $s_t = [a_1, ..., a_i, a_{i+1}]$.
- **Spatiotemporal dynamics**: They are defined as the evolution of the hidden states between consecutive activities and the update at activity times, which are continuous in time and space.
- **Reward function**: It evaluates the utility of taking an action under the state, whose input is the state-action pair. It is unknown and has to be learned.

3 METHOD

In this section, we present a novel framework, entitled **ActSTD**, which learns to generate **Act**ivity trajectories with capturing underlying **S**patio**T**emporal **D**ynamics. We first overview the ActSTD framework, and then elaborate on the details of key components.

3.1 Overview

As shown in Figure 2, the proposed framework has three key components: (i) learning spatiotemporal dynamics $\mathcal{G}_{st}$, which is modeled by a neural spatio-temporal point process, (ii) the policy function π_θ, which learns to generate action $\hat{a}$ similar to real-world cases based on the state $\hat{s}$ and spatiotemporal dynamics $\hat{g}$, (iii) the discriminator D_ϕ, which is trained to distinguish between policy-generated and real-world cases.

The left part of Figure 2 shows the learning of spatiotemporal dynamics underlying state transitions. It takes the state extracted from the observed trajectory as input and learns an embedding process

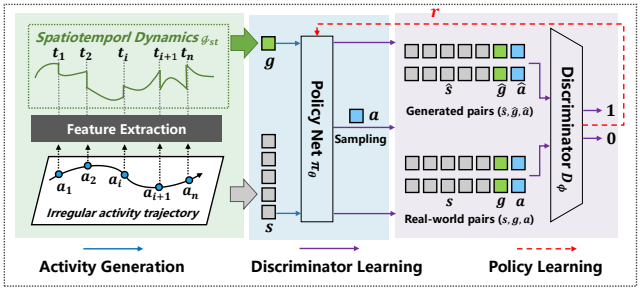


Figure 2: Illustration of the ActSTD framework. It has three key components: the spatiotemporal dynamics $\mathcal{G}_{st}$, the policy net π_θ, and the discriminator D_ϕ.

$g(t)$ in continuous time to describe the spatiotemporal dynamics. Specifically, we use neural differential equations [4] to describe the continuous characteristics both in time and space. Then, the policy network π_θ takes both the state s and the dynamics g as inputs to obtain an action distribution $\pi_\theta(a|s,g)$ and sample an action a accordingly. Subsequently, the discriminator D_ϕ distinguishes between the generated state-dynamics-action tuple $(\hat{s}, \hat{g}, \hat{a})$ and the observed tuple (s, g, a), and provides the reward for confusing its binary classification to in turn optimize the policy function.

3.2 Learning Spatiotemporal Dynamics

In the ActSTD framework, we consider an important factor underlying the decision process, namely, the spatiotemporal dynamics. Different from the previous modeling in GAIL, where the state is only updated upon the occurrence of a new action, we innovatively model the evolutionary dynamics between activity observations, which can adequately consider the continuous characteristics of states in time and space. The temporal dynamics and spatial dynamics are entangled by sharing the same hidden states $h(t)$. For easy understanding, we introduce the modeling of these two dynamic mechanisms, respectively.

3.2.1 ***Temporal Dynamics.*** The evolution of $h(t)$ is modeled by Neural Jump Stochastic Differential Equations [18], which is similar to a recurrent neural network with continuous-time hidden states between observed points. It should be pointed out that $h(t)$ serves as a summary of the activity history and act as the condition to predict future activity. On the one hand, $h(t)$ evolves continuously between observed points. On the other hand, the observed points, namely, the occurrence of location-based activities, can trigger instantaneous updates to $h(t)$, which incorporates the activity effects to the latent dynamics. The mechanism consisting of continuity and discontinuity is essential, because it not only captures the underlying patterns between activities, but also allows historical activity points to affect future development. In summary, the continuous flow and instantaneous update can be formulated as:

$$\frac{d\mathbf{h}(t)}{dt} = f_h(t, \mathbf{h}(t), \mathbf{x}(t)) \quad \text{(between activity times)} \tag{4}$$

$$\lim_{\Delta t \to 0^+} \mathbf{h}(t_i + \Delta t) = g_h(t_i, \mathbf{h}(t_i), \mathbf{x}(t_i)) \quad \text{(at activity times)} \tag{5}$$

where f_h is modeled by an MLP, g_h is modeled by a GRU to update the hidden states, and t_i denotes the activity time.

3.2.2 ***Spatial Dynamics***. Following neural STPP [3], we apply Continuous Normalizing Flow (CNF) [33] to model the conditional spatial density $p(\mathbf{x}|t)$. In the same way, the update of the normalizing flow is also conditioned on the hidden state $h(t)$, with the assumption that the activity history influences the spatial distribution. Similar to the pairwise continuity in temporal dynamics, the dynamics of spatial distribution also include continuous flow and instantaneous updates: (i) the spatial distribution follows a continuous-time normalizing flow between observed spatial locations, (ii) the spatial distribution can also be updated instantaneously by new activities, which is modeled by a discrete normalizing flow model. Such dynamics can be formulated as follows:

$$\frac{d\mathbf{x}(t)}{dt} = f_x(t, \mathbf{h}(t), \mathbf{x}(t)) \quad \text{(between activity times)} \quad (6)$$

$$\lim_{\Delta t \to 0^+} \mathbf{x}(t_i + \Delta t) = g_x(t_i, \mathbf{h}(t_i), \mathbf{x}(t_i)) \quad \text{(at activity times)} \quad (7)$$

where f_x is modeled by a continuous normalizing flow and g_x is realized by a standard linear flow.

3.3 Policy Design

The policy function is defined as a conditional intensity function $\lambda^*(t)$, which is used to sample the activity with the information of the timestamp, activity type, and spatial location. The conditional intensity function $\lambda^*(t)$ is defined as follows:

$$\lambda^*(t) = \mathbb{P}\{\text{event in } [t, t+dt) | \mathcal{H}_t\}\,, \quad (8)$$

where $*$ denotes the dependence on the history $\mathcal{H}$.

In our setup, the activity marks are considered, including the activity type k and spatial location (lon, lat). Thus, we model them using two conditional distributions $m(k|t)$ and $p(\mathbf{x}|t)$. The two distributions represent the probability that the activity type k and the spatial location x are selected conditioned on the occurrence of activity at time t. Following [3], we also perform decomposition of the conditional intensity function. Specifically, it is divided into three components as follows:

$$\lambda^*(t, k, \mathbf{x}) = \lambda^*(t) m(k|t) p(\mathbf{x}|t, k), \quad (9)$$

where $\lambda^*(t)$ denotes the intensity of the temporal process, $m(k|t)$ denotes the probability of activity type k given the time t, and $p(\mathbf{x}|t, k)$ is the conditional spatial density of $\mathbf{x}$ conditioned on the time t and activity type k. We model the first two terms by calculating intensity of each activity type $\lambda_k^*(t)$ as follows:

$$\lambda_k^*(t) = L_\lambda(h(t)),\ \lambda^*(t) = \sum_{k=1}^{K} \lambda_k^*(t),\ m(k|t) = \lambda_k^*(t) / \sum_{k=1}^{K} \lambda_k^*(t), \quad (10)$$

where L_λ is realized by an MLP followed with a softplus nonlinear layer to guarantee the positive value of intensities. Figure 3 demonstrates how we sample an activity with the policy function. After obtaining the intensity $\lambda^*(t)$ and the activity type distribution $m(k|t)$, we sample a time interval by $\Delta t \sim \text{Exp}(\lambda^*(t))$ and an activity type $k \sim \text{Categorical}(m(k|t))$. Then, the conditional spatial density evolves following the spatial dynamics.

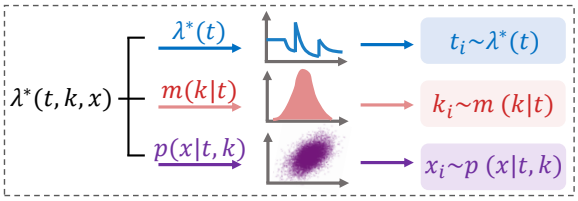


Figure 3: Illustration of the sampling process. Given the decomposed distributions, the time interval t_i, activity type k_i, and spatial location x_i are sampled.

3.4 Discriminator Design

GAIL uses a reward function to evaluate the actions by comparing the policy-generated actions with expert actions. It is modeled by a discriminator D_ϕ which provides rewards to guide the policy optimization. As shown in Figure 2, the input of the discriminator is a three-element tuple (s, g, a). We leverage an LSTM to encode the state history s and use an embedding layer to obtain the representation of the action a. Then we concatenate the state embedding, action embedding, and dynamics representation, and feed it into an MLP layer followed by a binary classifier. The optimization of ϕ is done with the following loss function:

$$\mathcal{L}_D = \mathbb{E}_{(s,g,a)\in\mathcal{T}_E} \log D_\phi(s, g, a) + \mathbb{E}_{(s,g,a)\in\mathcal{T}_G} \log(1 - D_\phi(s, g, a)), \quad (11)$$

where $\mathcal{T}_E$ and $\mathcal{T}_G$ are real-world activities and policy-generated activities, respectively. Thus, the reward can be calculated as follows:

$$R = \log D_\phi(s, g, a)\ . \quad (12)$$

3.5 Hybrid Training Technique

To enhance the learning of spatiotemporal dynamics, we propose a hybrid training technique to train the ActSTD framework. In the neural spatiotemporal point process, the likelihood is clearly defined and computationally tractable. Therefore, it can be trained via maximum likelihood as follows:

$$\log p(\mathcal{H}) = \sum_{i=1}^{N} \log \lambda^*(t_i) + \sum_{i=1}^{N} \log m(k_i|t_i) + \sum_{i=1}^{N} \log p(x_{t_i}|t_i, k_i) - \int_0^T \lambda^*(\tau) d\tau\ . \quad (13)$$

Additionally, we learn the policy and reward functions using an adversarial learning objective based on Eq. (3). Although MLE is statistically efficient, the combination of the adversarial learning is necessary because the optimality of MLE holds only without model misspecification for the generator [49]. Besides, [14] has shown that a purely adversarial learning paradigm tends to generate good samples but with low likelihoods. The ActSTD framework allows both maximum likelihood learning of the spatiotemporal dynamics and adversarial learning of the policy and reward functions. Algorithm 1 in Appendix shows the training procedure with hybrid techniques. Besides, we also consider other training options and investigate the performance experimentally in Section 4.3. The optional training methods include:

- **Option 1 (Pre-trained):** Pre-train the spatio-temporal dynamics model $\mathcal{G}_{st}$ with MLE, and then fix $\mathcal{G}_{st}$ when optimizing the policy net π_θ with PPO algorithm and the discriminator D_ϕ by a binary classification task.
- **Option 2 (Hybrid-trained):** The spatiotemporal dynamics model $\mathcal{G}_{st}$, the policy net π_θ, and the discriminator D_ϕ are all updated in the training process. Specifically, $\mathcal{G}_{st}$ is updated with real trajectories by MLE, π_θ and D_ϕ are trained with adversarial learning.
- **Option 3 (Jointly-trained):** The spatiotemporal dynamics model $\mathcal{G}_{st}$ is integrated into the policy net and is jointly optimized with the policy net by the PPO algorithm [43].

4 EXPERIMENT

In this section, we perform extensive experiments on two real-world activity datasets to answer the following research questions:

- **RQ1:** Compared with state-of-the-art baselines, can our framework deliver more realistic activity simulations?
- **RQ2:** How do different training methods that learn spatiotemporal dynamics contribute to the final performance?
- **RQ3:** What's the utility of the generated activity trajectories in supporting practical applications?
- **RQ4:** What's the performance of our framework to facilitate the modeling of epidemic spreading?

4.1 Experimental Settings

4.1.1 ***Datasets.*** We evaluate the performance of the ActSTD framework on two large-scale real-world activity datasets. As activity data are characterized by visits to different types of locations, we use the location category as the activity type. The first dataset was collected by a major mobile network operator in China, which includes 10000 users in Beijing for a duration of one month. This dataset contains the following categories: Company, Concerts, Culture and Art, Education, Entertainment, Food, Government, Life Service, Market, Medicine, School, Shop, Sports, Travel, and University. The second dataset [51] was collected from Foursquare, including 1000 users with a duration of one month. The anonymous user ID, timestamp, activity type, and latitude-longitude coordinations are recorded in both datasets. This dataset contains activity types as: Arts and Entertainment, Athletics and Sports, Clothing Store, College and University, Food, Food and Drink Shop, Medical Center, Movie Theater, Nightlife Spot, Office, Outdoors and Recreation, Professional and Other places, School, Shop and Service.

4.1.2 ***Baselines.*** We compare the ActSTD's performance with the following state-of-the-art baselines: SMM [24], TimeGeo [19], LSTM [17], SeqGAN [54], TrajGAIL [5], Movesim [9] and Neural STPP [3]. The details of baselines are introduced in Appendix B.

4.1.3 ***Performance criteria.*** The objective of this work is to generate activity trajectories that are similar to real-world activities. Here, we define the "similarity" between the generated activities and real-world activities from two aspects: **dataset level** and **individual level**. First, the dataset-level similarity denotes the distributional or statistical similarity between two datasets from an overall perspective. This macro-level similarity measures how closely the generated dataset matches the reality of statistical characteristics by comparing distributions of key patterns. Second, the individual-level similarity measures the similarity of an individual's generated trajectory to the observed cases. For example, given the same state, a good simulator should not only predict the next activity accurately, but also perform well in generating the future trajectories with multiple consecutive steps.

For the dataset-level evaluation, we choose essential aspects of activity patterns for statistical similarity as follows:

- ***Distance***: The moving distance between activities in individuals' trajectories, which is a metric from the spatial perspective.
- ***Radius***: Radius of gyration is the root mean square distance of all activity locations from the central one, which measures the spatial range.
- ***Interval***: Time intervals between adjacent activities in activity trajectories, which is a metric from the temporal perspective.
- ***DailyAct***: Daily performed activities [9], which is calculated as the number of activities per day for each individual. It reflects a certain degree of daily rhythms.
- ***ActType***: The distribution of activity type, which is calculated as the frequency of each activity at the macro-level.

Specifically, we use Jensen–Shannon divergence (JSD) [11], to measure the discrepancy of these distributions between the generated data and real-world data. The JSD metric is defined as follows:

$$\mathrm{JSD}(p||q) = H((p+q)/2) - \frac{1}{2}(H(p) + H(q)) \tag{14}$$

where p and q are two distributions for comparison, and H is the Shannon entropy. Lower JSD denotes a closer match to the statistical characteristics and thus indicates a better generation result.

In the individual-level evaluation, we following existing studies [52] and use the standard evaluation metrics for prediction performance, *i.e.*, accuracy. Specifically, we both predict the activity type and spatial location.

4.2 Overall Performance (RQ1)

We demonstrate the performance of ActSTD and state-of-the-art baselines, where both dataset-level similarity and individual-level similarity are evaluated in the following sections.

4.2.1 ***Dataset-level Evaluation.***

Table 1 reports the results of our framework and state-of-the-art baselines for the dataset-level evaluation.

Performance on Mobile Network Operator Dataset. As we can observe in Table 1, SMM performs the worst across all metrics, indicating that the time-invariant assumption cannot describe real-world activity patterns. TimeGeo outperforms SMM significantly, especially for the spatial-related metrics including *Distance* and *Radius*. It is reasonable as TimeGeo directly fits the distribution of the jump-size (*Distance*) by a fat-tailed distribution [45], which can well capture distance characteristics. However, it fails to model the choice of activity type, which suggests that the activity decision does not solely rely on the distance. The incorporation of deep neural networks improves the similarity of activity type significantly, and SeqGAN even achieves the best performance on the *ActType* metric. Despite this, these models cannot achieve acceptable results on other spatial-related and temporal-related metrics, which reflect the intractable challenge of capturing the spatiotemporal

Dataset	Mobile Network Operator					Foursquare				
Metrics (JSD)	Distance	Radius	Interval	DailyAct	ActType	Distance	Radius	Interval	DailyAct	ActType
SMM	0.348	0.583	0.184	0.649	0.047	0.245	0.244	0.117	0.405	0.066
TimeGeo	**0.056**	0.203	0.015	0.174	0.040	0.146	0.528	0.133	0.199	0.079
LSTM	0.121	0.556	0.044	0.319	0.004	0.124	0.420	0.078	0.266	0.021
SeqGAN	0.101	0.549	0.035	0.226	**0.003**	0.097	0.372	0.065	0.271	0.018
TrajGAIL	0.248	0.539	0.175	0.416	0.007	0.158	0.484	0.230	0.693	0.023
Movesim	0.289	0.556	0.160	0.232	0.009	0.154	0.480	0.260	0.693	0.011
Neural STPP	0.168	0.296	0.009	0.051	0.013	0.215	0.430	**0.034**	0.105	0.0066
Ours	0.071	**0.142**	**0.0077**	**0.0084**	0.004	**0.082**	**0.138**	0.054	**0.093**	**0.0063**

Table 1: Performance of the proposed ActSTD framework and baselines in terms of JSD. Lower JSD value indicates a better performance. Bold means the best results and underline means the second-best results.

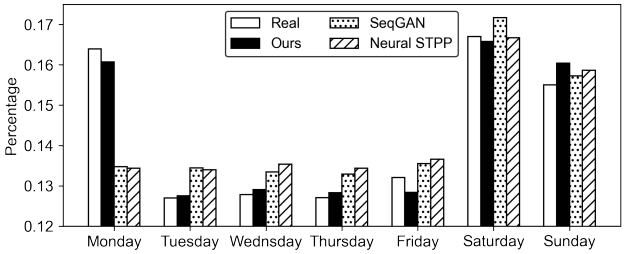


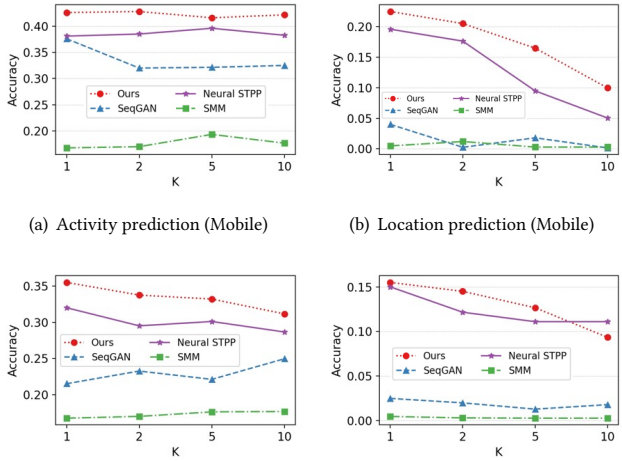
Figure 4: Weekday distributions of the real-world data, the generated data by our framework and two best baselines.

dynamics underlying activity trajectories. Neural STPP achieves the second-best performance on spatial-related and temporal-related metrics, which proves the importance and necessity of considering spatiotemporal dynamics. Overall, ActSTD achieves the best performance, which ranks 1st on three metrics and 2nd on two metrics. For the three metrics that rank 1st, ActSTD reduces the JSD from 14.4% to 83.5%. For the two metrics ranking 2nd, ActSTD also has comparable performance compared with the best baseline.

Performance on Foursquare Dataset. The evaluation results on the Foursquare dataset are different from the Mobile Network Operator dataset, which can be explained by the data sparsity and the small data scale. Due to the data sparsity that intervals between consecutive activities are obviously longer, the activity trajectory in the Foursquare dataset does not exhibit obvious regularities. As a result, models including SMM and TimeGeo, which need to estimate temporal and spatial regularity parameters from the training data, deliver worse performance. Particularly, the TimeGeo model loses its advantage in modeling spatial characteristics. Despite the changes in the dataset distribution, ActSTD still achieves the best performance with ranking 1st on four metrics and 2nd on one metric. For the four metrics that rank first, ActSTD reduces the JSD from 4.5% to 43.0%. For the *Interval* metric ranking second, ActSTSD still achieves comparable results with the best baseline.

Distribution Visualization. In addition, to further demonstrate the performance gain on the dataset-level evaluation, we visualize the weekday distribution of the real data, the data generated by ActSTD and by the other two best baselines, SeqGAN and Neural STPP. As we can observe in Figure 4, the data generated by our framework has the most similar distribution to the ground truth.

In summary, the performance gain on two real-world datasets and the visualization of the weekday distribution all show the better



(a) Activity prediction (Mobile) (b) Location prediction (Mobile)

(c) Activity rediction (Foursquare) (d) Location prediction (Foursquare)

Figure 5: Performance of the prediction task of our proposed framework, SMM, and two best baselines. We vary the number of future steps in prediction as 1, 2, 5, 10**.**

performance of the proposed ActSTD framework in generating activity trajectories that match the reality at the macro level.

4.2.2 ***Individual-level Evaluation.*** In the individual-level evaluation, we measure how similar each generated activity trajectory is to the real circumstances by performing activity predictions. To illustrate the performance of the ActSTD framework, we not only predict the next activity, but also predict the following k steps, where $k \in \{1, 2, 5, 10\}$. Specifically, we take the historical 10 activities as the known state and predict the next k activities.

As we can observe in Figure 5, ActSTD performs the best across prediction tasks. SMM performs the worst across all cases due to the unrealistic Markov assumption [10] that the future conditional distribution only relies on the current state and is independent of the history. Besides, as the number of future steps to predict increases, the performance of ActSTD and SeqGAN are not reduced a lot, which can be explained as the optimization with long-term rewards. In summary, the prediction performance of these methods at the individual-level evaluation is consistent with the performance at the dataset-level evaluation to a large degree, indicating the prediction bears certain similarities to the simulation.

Dataset	Mobile Network Operator					Foursquare				
Metrics (JSD)	Distance	Radius	Interval	DailyAct	ActType	Distance	Radius	Interval	DailyAct	ActType
Pre-trained	0.168	0.222	0.0094	0.053	0.025	0.124	0.126	0.062	0.22	0.027
Jointly-trained	0.092	0.148	0.011	0.011	0.030	0.086	**0.121**	0.064	0.172	0.029
Hybrid-trained	**0.071**	**0.142**	**0.0077**	**0.0084**	**0.004**	**0.082**	0.138	**0.054**	**0.093**	**0.0063**

Table 2: Performance of different training methods in terms of JSD. Lower JSD value indicates a better performance. Bold means the best results and underline means the second-best results.

4.3 Study of the Training Process (RQ2)

As discussed in Section 3.5, the spatiotemporal dynamics can be learned in three optional ways: (1) Pre-train $\mathcal{G}_{st}$ with MLE and fix it in the adversarial learning, (2) Hybrid train $\mathcal{G}_{st}$ with MLE and policy net π_θ with PPO algorithm iteratively, (3) Jointly train $\mathcal{G}_{st}$ together with the policy net π_θ in one model by PPO algorithm. We illustrate the results of different training processes in Table 2.

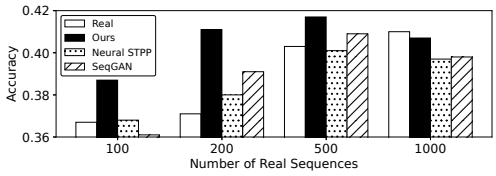
As we can observe, the training method affects the final performance of the ActSTD framework. In general, the hybrid training technique achieves the best performance in most cases. The combined training method, together with likelihood optimization and adversarial learning, not only gives rise to samples that resemble real cases but also increases the likelihood of the states and actions.

4.4 Case Study (RQ3)

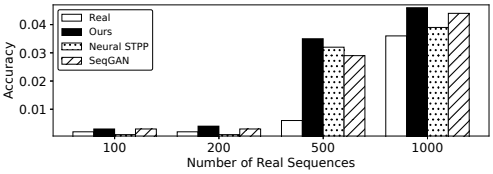
In addition to the dataset-level and individual-level evaluations, we also conduct case studies to assess the utility of the generated activity trajectories. We consider a scenario where real-world activity records from large mobile operators cannot be directly shared with other companies for practical applications due to privacy issues, which is quite common in user-related applications. In these cases, the ActSTD framework can be applied to generate artificial activity trajectories that retain the utility of real-world data. Thus, the generated data can be utilized to augment limited scales of real-world activity data in practical applications.

Here we perform location-based activity prediction [36, 52], which is a classical and well-performed application on activity data. Specifically, we leverage an LSTM with an attention mechanism as the prediction model. To compare the performance of data augmentation, we also apply the data generated by the two best baselines, Neural STPP and SeqGAN. As a result, the training samples of the prediction model can be divided into four categories: (1) only real-world data, (2) real-world data augmented by our framework, (3) real-world data augmented by Neural STPP, and (4) real-world data augmented by SeqGAN. Then the prediction model is trained with the four types of inputs and evaluated on the same test set. Furthermore, we vary the number of real-world data as $\{100, 200, 500, 1000\}$ to investigate the performance gain of the data augmentation under different circumstances. Meanwhile, we all add 1000 generated activity trajectories for different amounts of real data. We use the Mobile Network Operator data in this section, considering the data sparsity of the Foursquare data. Both the activity type and the spatial location are predicted in the experiments.

As we can observe in Figure 6, the prediction model trained with the augmented data exhibits better prediction outcomes. Besides, the results show that the relative performance gain of data augmentation is more obvious with small-scale real-world data,



(a) Activity prediction



(b) Location prediction

Figure 6: Results of Activity prediction with training data augmented by generated data. For different number of real-world data, *i.e.*, 100, 200, 500, and 1000, we all add 1000 generated activity trajectories for data augmentation.

such as the results of the activity type prediction only with 100 or 200 real-world sequences. Moreover, as Figure 6(a) illustrates, the data with only 200 real-world sequences while augmented by the ActSTD framework has achieved better performance than the data with even 1000 real-world data. These experimental results prove the utility of the generated data in practical applications.

4.5 SEIR Modeling of COVID-19 (RQ4)

To testify the utility of the synthetic data in supporting epidemic modeling, we leverage the activity from the mobile operator dataset to simulate the epidemic spreading with the SEIR model following recent works [9, 26]. We follow the setup of the detailed parameters of SEIR simulation as [9, 26]. Specifically, there are eight key parameters, including close contact ratio (c), transmission period (T), incubation period (T_i), infection period (T_f), reproduction rate (R_0), transmission probability (β), infectious rate (α), recovery rate (r). Table 3 shows the parameter values in the simulation experiment.

Parameters	c	T	T_i	T_f	R_0	β	α	r
Value	0.2	$5.8d$	$5.2d$	$11d$	2.2	R_0/T	$1/T_i$	$1/T_f$

Table 3: Setup of parameters for modeling COVID-19 spreading following [26], where d denotes days.

To simulate the epidemic spreading with activity trajectory data, we first construct a contact network by adding an edge between two individuals if they stay in the same location and take the same activity. Then in the simulation process, we assume that infected or

exposed people contact with s susceptible people with edges in the contact network each day. The probability of two individuals with an edge becoming a close contact is set as c. The transmission probability β is calculated as the basic reproduction rate divided by the transmission period (5.8 days), which is the average number of days from onset to first medical visit and isolation [26]. The infectious rate from exposed (α) is estimated as the inverse of the incubation period, which is the average time exposed but not infectious (5.2 days in [26]). The spreading process is formulated as follows:

$$\dot{S} = -sc\beta,\ \dot{E} = sc\beta - \alpha E,\ \dot{I} = \alpha E - rI,\ \dot{R} = rI \tag{15}$$

where S is the number of the susceptible, E is the number of the exposed, I is the number of the infectious, and R is the number the removed. Based on the above dynamic mechanism, we can simulate the epidemic spreading and obtain the daily number of the exposed, the infectious, and the removed. In the experiment, 50 individuals are initialized as the exposed randomly. Considering that the random initialization of exposed individuals will give rise to different transmissions, we perform simulations ten times in each experiment to avoid the influence of the random seed.

We conduct the simulation experiment on three types of data, including real-world data, synthetic data of ActSTD, and synthetic data of SeqGAN. Among them, real-world data serve as the ground truth, and we evaluate the MAPE of different estimated populations (E, I, R). Figure 7 shows the comparison results of the average error across the simulation period. As we can observe, synthetic data generated by our framework delivers closer simulation results to the real data compared with the baseline methods, which reduce the MAPE by over 50% on three kinds of populations.

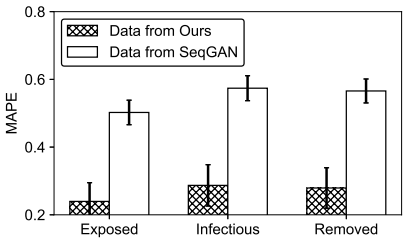


Figure 7: Simulation results of COVID-19 spreading with different synthetic data.

5 RELATED WORK

Applications of Activity Data. Location-based activity data have benefited a plethora of applications including mobility prediction [8, 12], location or activity recommendation [28, 41], health care [21], activity scheduling [16], friend recommendations, and social link predictions. The data generated by our proposed simulation framework can facilitate the above-mentioned applications without privacy concerns and large data scales. Besides, our framework is also useful to investigate and understand individuals' decision-making processes and describe insightful activity patterns [30, 37]. For example, with realistic activity data, researchers can analyze the behavioral patterns [38, 47] and further explain why users choose to perform the activity at the time and location [37]. while extremely valuable of the activity data, publicly available datasets [1, 37, 50] suffer from data sparsity and small scales [23]. Under these circumstances, our framework is able to enrich the real-world datasets by generating artificial datasets with retaining data fidelity and utility.

Deep Generative Models. Recurrent Neural Networks (RNNs) have been widely used to predict the sequential activities with considering the spatio-temporal contexts [22, 29]. Although RNN-based models can be utilized to generate activity sequences, they cannot consider the long-term influence and thus fail to perform a realistic simulation. Recently, Generative Adversarial Network (GAN) [13] has been proposed to learn a good generator in an adversarial manner. Shortly afterwards, its sequential invariant, SeqGAN [54], is proposed to address sequence generation tasks. To mimic the decision-making process of agents, Generative Adversarial Imitation Learning (GAIL) developed the initial imitation learning by integrating the GAN framework, which has been applied in various decision-making scenarios [40, 55]. However, the decision-making process of the activity choice is complex, and it is essential to learn effective states that describe human activities. Therefore, our proposed framework takes both the spatio-temporal dynamics and the policy function to simulate the activity decision.

Learning Differential Equations. Differential equations, such as ordinary differential equation and stochastic differential equation, are leveraged to model the dynamics in many physical, biological and economic systems [20, 27, 32, 34]. Recently, learning parameters of differential equations with neural networks has been successful for many physics-based problems [31, 42, 48]. Specifically, spatiotemporal dynamics are captured with dependencies and propagation effects by utilizing differential equations in practical applications, including traffic flow forecasting [7], fluid simulation [44], earthquake and epidemic modeling [3]. Inspired by these works, we take the first step to capture spatiotemporal dynamics underlying activity trajectories to facilitate activity simulation.

6 CONCLUSION

In this paper, we propose a novel generative adversarial-based framework, ActSTD, to simulate activity trajectories by capturing underlying spatiotemporal dynamics. By incorporating the dynamics of continuous flow and instantaneous updates in time and space, our proposed simulation framework benefits the generation of massive high-quality activity trajectories. Extensive experiments demonstrate the superior performance of the proposed framework in generating artificial data with both fidelity and utility. Moreover, it achieves better performance in modeling the COVID-19 spreading. As for future work, we will investigate the interactions between individuals in the simulation process and extend the framework to various practical applications.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under grant 2020YFA0711403, the National Nature Science Foundation of China under 61971267, 61972223, 62171260, U1936217, and Young Elite Scientists Sponsorship Program by CIC (Grant No. 2021QNRC001).

REFERENCES

[1] Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362* (2016).
[2] Y. Bengio and P. Frasconi. 1996. Input-output HMMs for sequence processing. *IEEE Transactions on Neural Networks* 7, 5 (1996), 1231–1249. https://doi.org/10.1109/72.536317

[3] Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. 2020. Neural spatio-temporal point processes. *arXiv preprint arXiv:2011.04583* (2020).
[4] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *NIPS* 31 (2018).
[5] Seongjin Choi, Jiwon Kim, and Hwasoo Yeo. 2021. TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning. *Transportation Research Part C: Emerging Technologies* 128 (2021), 103091.
[6] Peter J Diggle. 2006. Spatio-temporal point processes: methods and applications. *Monographs on Statistics and Applied Probability* 107 (2006), 1.
[7] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *KDD*. 364–373.
[8] 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*. 1459–1468.
[9] Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. 2020. Learning to simulate human mobility. In *KDD*. 3426–3433.
[10] Shai Fine, Yoram Singer, and Naftali Tishby. 1998. The hierarchical hidden Markov model: Analysis and applications. *Machine learning* 32, 1 (1998), 41–62.
[11] Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *ISIT 2004*. IEEE, 31.
[12] Qiang Gao, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2019. Predicting human mobility via variational attention. In *The World Wide Web Conference*. 2750–2756.
[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
[14] Aditya Grover, Manik Dhar, and Stefano Ermon. 2018. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
[15] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).
[16] Serge P Hoogendoorn and Piet HL Bovy. 2004. Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological* 38, 2 (2004), 169–190.
[17] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).
[18] Junteng Jia and Austin R Benson. 2019. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems* 32 (2019).
[19] 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. *PNAS* 113, 37 (2016), E5370–E5378.
[20] Douglas Samuel Jones, Michael Plank, and Brian D Sleeman. 2009. *Differential equations and mathematical biology*. Chapman and Hall/CRC.
[21] Ingibjörg H Jonsdottir, Lars Rödjer, Emina Hadzibajramovic, Mats Börjesson, and Gunnar Ahlborg Jr. 2010. A prospective study of leisure-time physical activity and mental health in Swedish health care workers and social insurance officers. *Preventive medicine* 51, 5 (2010), 373–377.
[22] Joo-Chang Kim and Kyungyong Chung. 2019. Prediction model of user physical activity using data characteristics-based long short-term memory recurrent neural networks. *TIIS* 13, 4 (2019), 2060–2077.
[23] Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Ovi Chris Rouly, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2020. Location-based social network data generation based on patterns of life. In *MDM*. IEEE, 158–167.
[24] Vo S Korolyuk, SM Brodi, and AF Turbin. 1975. Semi-Markov processes and their applications. *Journal of Soviet Mathematics* 4, 3 (1975), 244–280.
[25] Jason T Ladner, Nathan D Grubaugh, Oliver G Pybus, and Kristian G Andersen. 2019. Precision epidemiology for infectious disease control. *Nature medicine* 25, 2 (2019), 206–211.
[26] Shengjie Lai, Nick W Ruktanonchai, Liangcai Zhou, Olivia Prosper, Wei Luo, Jessica R Floyd, Amy Wesolowski, Mauricio Santillana, Chi Zhang, Xiangjun Du, et al. 2020. Effect of non-pharmaceutical interventions to contain COVID-19 in China. *nature* 585, 7825 (2020), 410–413.
[27] Anthony W Leung. 2013. *Systems of nonlinear partial differential equations: applications to biology and engineering*. Vol. 49. Springer Science & Business Media.
[28] Guoqiong Liao, Shan Jiang, Zhiheng Zhou, Changxuan Wan, and Xiping Liu. 2018. POI recommendation of location-based social networks using tensor factorization. In *MDM*. IEEE, 116–124.
[29] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI conference on artificial intelligence*.
[30] Xuelian Long, Lei Jin, and James Joshi. 2012. Exploring trajectory-driven local geographic topics in foursquare. In *Proceedings of the 2012 ACM conference on ubiquitous computing*. 927–934.
[31] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. 2018. Pde-net: Learning pdes from data. In *ICML*. PMLR, 3208–3216.
[32] VP Maksimov. 2006. Theory of functional differential equations and some problems in economic dynamics. In *Proceedings of the Conference on Differential and Difference Equations and Applications*. Hindawi Publishing Corporation New York, 74–82.
[33] Emile Mathieu and Maximilian Nickel. 2020. Riemannian continuous normalizing flows. *arXiv preprint arXiv:2006.10605* (2020).
[34] Remigijus Mikulevicius and Boris L Rozovskii. 2004. Stochastic Navier–Stokes equations for turbulent flows. *SIAM Journal on Mathematical Analysis* 35, 5 (2004), 1250–1310.
[35] Bjarke Frost Nielsen, Kim Sneppen, Lone Simonsen, and Joachim Mathiesen. 2021. Differences in social activity increase efficiency of contact tracing. *The European Physical Journal B* 94, 10 (2021), 1–11.
[36] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2012. Mining user mobility features for next place prediction in location-based services. In *ICDM*. IEEE, 1038–1043.
[37] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. 2011. An empirical study of geographic user activity patterns in foursquare. In *ICWSM*, Vol. 5. 570–573.
[38] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. 2011. Exploiting semantic annotations for clustering geographic areas and users in location-based social networks. In *ICWSM*, Vol. 5. 32–35.
[39] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. 2018. A Non-Parametric Generative Model for Human Trajectories.. In *IJCAI*. 3812–3817.
[40] Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, and Jun Luo. 2020. xgail: Explainable generative adversarial imitation learning for explainable human decision analysis. In *KDD*. 1334–1343.
[41] Yukun Ping, Chen Gao, Taichi Liu, Xiaoyi Du, Hengliang Luo, Depeng Jin, and Yong Li. 2021. User Consumption Intention Prediction in Meituan. In *KDD*. 3472–3482.
[42] Maziar Raissi. 2018. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research* 19, 1 (2018), 932–955.
[43] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
[44] Varun Shankar, Gavin D Portwood, Arvind T Mohan, Peetak P Mitra, Christopher Rackauckas, Lucas A Wilson, David Schmidt, and Venkatasubramanian Viswanathan. 2020. Learning non-linear spatio-temporal dynamics with convolutional Neural ODEs. In *NeurIPS*.
[45] 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.
[46] Dandan Tian, Yanhong Sun, Huihong Xu, and Qing Ye. 2022. The emergence and epidemic characteristics of the highly mutated SARS-CoV-2 Omicron variant. *Journal of Medical Virology* 94, 6 (2022), 2376–2383.
[47] Xiaoxuan Wang, Lizhen Wang, and Peizhong Yang. 2019. Prevalent co-visiting patterns mining from location-based social networks. In *MDM*. IEEE, 581–586.
[48] Yichen Wang, Evangelos Theodorou, Apurv Verma, and Le Song. 2018. A stochastic differential equation framework for guiding online user activities in closed loop. In *AISTATS*. PMLR, 1077–1086.
[49] Halbert White. 1982. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the econometric society* (1982), 1–25.
[50] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting user mobility and social relationships in lbsns: a hypergraph embedding approach. In *The world wide web conference*. 2147–2157.
[51] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014).
[52] Jihang Ye, Zhe Zhu, and Hong Cheng. 2013. What's your next move: User activity prediction in location-based social networks. In *ICDM*. SIAM, 171–179.
[53] Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-François Paiement, and Alexei Pozdnoukhov. 2017. A generative model of urban activities from cellular data. *IEEE TITS* 19, 6 (2017), 1682–1696.
[54] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, Vol. 31.
[55] Haosheng Zou, Hang Su, Shihong Song, and Jun Zhu. 2018. Understanding human behaviors in crowds by imitating the decision-making process. In *AAAI*, Vol. 32.

A DATASET STATISTICS

Table 4 illustrate the basic statistics of the mobile network operator dataset and the Foursquare dataset, respectively. It shows information including the number of users (#users), the number of activity types (#activity types) and the average length of activity trajectories (average length).

Statistics	Mobil dataset	Foursquare dataset
#users	10000	1000
#activity types	13	9
average length	82.92	58.56

Table 4: The basic statistics of the Mobile Network Operator dataset and the Foursquare dataset.

In addition, we preprocess the used datasets by filtering users with less than 50 activities in the Mobile Network Operator dataset and users with less than 30 activities in the Foursquare dataset. Then, we illustrate the distribution of activity trajectory length of the two datasets in Figure 8 and Figure 9.

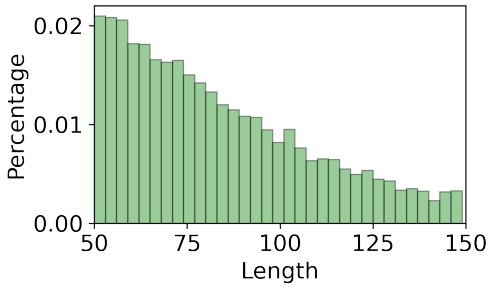


Figure 8: Distribution of the activity trajectory length in the Mobile Network Operator dataset.

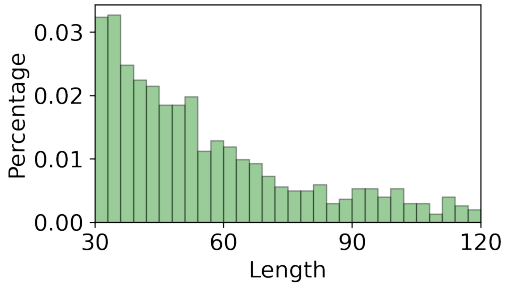


Figure 9: Distribution of the activity trajectory length in the Foursquare dataset.

B BASELINES

We provide the details of baseline methods in our experiments.

- **SMM [24].** Semi-Markov Model describes the time interval with exponential distribution with gamma prior and builds a transition matrix with Dirichlet prior to implementing a Bayesian inference.
- **TimeGeo [19].** It assumes that an individual is attracted to previously visited places based on historical preference as well as new places depending on the distance from the current position.
- **LSTM [17].** It is widely used in addressing sequences. Here this model directly predicts the time interval, activity type and spatial location based on the activity history.
- **SeqGAN [54].** It solves the sequential generation problem by introducing reinforcement learning into GAN. We apply this method to generate the activity sequence.
- **TrajGAIL [5].** It is a generative adversarial imitation learning framework for trajectory generation. We apply this method for both decisions on both activity type and spatial location choices.
- **Movesim [9].** As a mobility trajectory simulation model proposed recently, it integrates physical regularities and prior knowledge into the SeqGAN model.
- **Neural STPP [3].** It is the state-of-the-art baseline to model spatio-temporal sequences.

C TRAINING ALGORITHM

Algorithm 1 Hybrid training process of ActSTD

Input: Observed true trajectories $\mathcal{T}_E$, initial policy, discriminator, and spatiotemporal dynamics parameters $\theta_0, \phi_0, \mathcal{G}_0$.
Output: $\pi_\theta, D_\phi, \mathcal{G}_{st}$

```
for i ← 0, 1, 2... do
    Rollout activity trajectories for the agents T_G, where τ = (s, g, a) ;
    Score τ from T_G with D_φ, calculate reward based on Eq. (12);
    Update π_θ via PPO with the reward R based on Eq. (12) ;
    Update G_st via MLE with T_E based on Eq. (13) ;
    Update D_φ based on Eq. (11).
end for
```

Algorithm 1 shows the training procedure of ActSTD with hybrid techniques, and other optional training methods share a similar process. At each iteration, the policy parameters are used to generate activity a and $\mathcal{G}_{st}$ parameters are used to obtain dynamics. After that, rewards for each state-dynamics-action tuple are calculated by the discriminator parameters and are utilized to update the policy parameters via PPO algorithm [43]. Then the observed true activities $\mathcal{T}_E$ are used to optimize $\mathcal{G}_{st}$ parameters with MLE. The generated activities and observed activities subsequently serve as training data to optimize discriminator parameters D_ϕ.

D IMPLEMENTATION DETAILS

D.1 Parameter Settings

In our experiments, two-layer MLPs are used in the network architecture with a hidden size of 32. The hidden size of MLPs is both set

as 64 and the embedding size is set as 32. We perform the simulation in a mini batch of 32. The policy network and the spatiotemporal dynamics model are updated 5 epochs at each round, while the discriminator is updated 3 epochs every two rounds. The learning rate is set as 5e-5 via grid searching in a set of $\{3e-4, 1e-4, 5e-5, 1e-5\}$.

The deep learning models in our work are implemented with Pytorch and classical methods are implemented with Python. The models are all trained on a server with two CPUs and eight GPUs. Empirically, our proposed ActSTD framework can be effectively trained in less than 10 hours on a single GPU.

D.2 Evaluation Metrics

In addition to the JSD-based metrics (widely used distance metrics for distributions [9, 39]) to evaluate the dataset-level similarity, there are two other metrics used to evaluate the individual-level performance and examine the data utility in practical applications.

Accuracy. We use accuracy to evaluate the performance of individual-level activity prediction, which is the percentage of the value predicted as the true value. Higher accuracy indicates a better prediction performance.

MAPE. We use Mean Absolute Percentage Error to evaluate the simulation results in the modeling of COVID-19 spread. The metric is defined as follows:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{\hat{y_i} - y_i}{y_i} \tag{16}$$

where $\hat{y_i}$ is the simulation result with the synthetic data and y_i is the simulation result with real-world data. Lower MAPE indicates a better simulation result.