CSGAN: Modality-Aware Trajectory Generation via Clustering-based Sequence GAN

Minxing Zhang[1], Haowen Lin[2], Shun Takagi[3], Yang Cao[4], Cyrus Shahabi[2], Li Xiong[1]
[1]Emory University, [2]University of Southern California
[3]Kyoto University, [4]Hokkaido University
[1]{minxing.zhang,lxiong}@emory.edu, [2]{haowenli,shahabi}@usc.edu
[3]takagi.shun.45a@st.kyoto-u.ac.jp, [4]yang@ist.hokudai.ac.jp

***Abstract*—Human mobility data is useful for various applications in urban planning, transportation, and public health, but collecting and sharing real-world trajectories can be challenging due to privacy and data quality issues. To address these problems, recent research focuses on generating synthetic trajectories, mainly using generative adversarial networks (GANs) trained by real-world trajectories. In this paper, we hypothesize that by explicitly capturing the modality of transportation (e.g., walking, biking, driving), we can generate not only more diverse and representative trajectories for different modalities but also more realistic trajectories that preserve the geographical density, trajectory, and transition level properties by capturing both cross-modality and modality-specific patterns. Towards this end, we propose a Clustering-based Sequence Generative Adversarial Network (CSGAN) that simultaneously clusters the trajectories based on their modalities and learns the essential properties of real-world trajectories to generate realistic and representative synthetic trajectories. To measure the effectiveness of generated trajectories, in addition to typical metrics that measure how well the trajectories preserve density and trajectory level statistics, we define several new metrics for a comprehensive evaluation, including modality distribution and transition probabilities both globally and within each modality. Our extensive experiments with real-world datasets show the superiority of our model in various metrics over state-of-the-art models.**

***Index Terms*—Generative Adversarial Networks, Clustering, Reinforcement Learning, Synthetic Trajectory Generation**

I. Introduction

The recent growth in location-based technology, such as mobile devices and sensors equipped with GPS, has led to an unprecedented increase in the analysis and management of human mobility data. This data, often represented as a series of ordered points, reflects the physical–behavioral trace of an individual's movement. Understanding the mobility patterns of a population has significant implications for a wide range of applications, including transportation, epidemiological modeling, and public health [1]. For instance, pandemic risk evaluation via mobility data during COVID-19 can help understand, estimate, and mitigate the disease spread [2]. In addition, analyzing individual movements can provide insight into traffic or public transportation systems and help address traffic congestion and urban planning. Recommendation systems also rely on population flow data to identify effective advertising locations. Despite the value of mobility data, obtaining and sharing large-scale real-world trajectories can be challenging due to privacy and commercial concerns [3]. As a result, generating synthetic, realistic trajectories has become a valuable and important research problem to either scale up or protect the privacy of the original trajectory data so that they can be used for downstream tasks.

To address the synthetic trajectory generation task, existing methods can be mainly categorized into 1) earlier Markov-based models [4], which rely on simplified mobility assumptions; 2) deep predictive models [5], which can learn more complex sequential patterns; and 3) more recent state-of-the-art generative adversarial network (GAN)-based models [6] which can generate more realistic trajectories based on the generator-discriminator adversarial game. To better generate trajectories that are represented as discrete location sequences in contrast to grid-based data such as images, [6] proposes a recurrent neural network (RNN)-based sequence GAN and leverages reinforcement learning (policy gradient) and Monte Carlo search to generate discrete sequences. Followup works such as [7] extend sequence GAN and attempt to capture the mobility regularity via incorporating the urban structure.

While these GAN-based models generate sequences that preserve the spatiotemporal statistics of the original trajectories to some level (e.g., global density statistics such as the visiting probability of a location or trajectory-level statistics such as average daily travel distance per trajectory), they do not consider important semantic information such as the modality of the trajectories. Real-world trajectories always consist of various modalities, including transportation modalities, such as walking, biking, or driving, or more implicit moving purposes, e.g., shopping, going to work, or sightseeing. While there are common mobility regularity and transition patterns shared across these modalities, there are also modality-specific characteristics and patterns. For example, trajectories of different transportation modalities may have different average speeds, accumulative distances, number of distinct visits, and sequential transition patterns (e.g., transitions on pedestrian-only streets for walking trajectories). Without considering this information explicitly, the resulting trajectories 1) may not capture the modality distributions and may not be diverse and representative of the different modalities, and 2) may not capture the *modality-specific* characteristics and may generate unrealistic trajectories that do not correspond to the real-world

modality or moving behaviors.

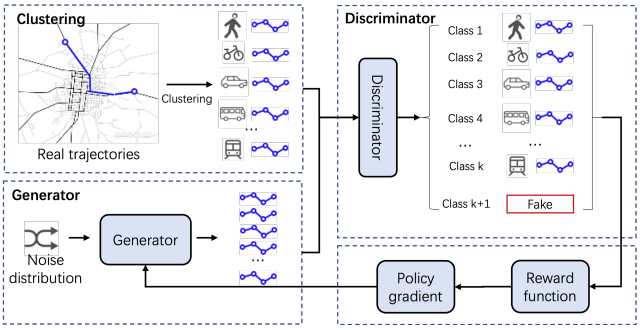


Fig. 1: Proposed CSGAN framework

Contributions. Towards this end, we hypothesize that by explicitly capturing the modality in the trajectories (e.g., walking, biking, driving) and learning from both cross-modality and modality-specific patterns, we can generate 1) more diverse and representative trajectories and 2) more realistic trajectories across all modalities. Existing approaches that do not consider modality only learn global patterns but not modality-specific patterns. A naive approach that trains an independent GAN for each modality will also not work well since it will miss the cross-modality patterns. Thus, we propose a Clustering-based Sequence Generative Adversarial Network (CSGAN) that simultaneously clusters the trajectories based on their modalities and learns both cross-modality and modality-specific properties to generate realistic and representative synthetic trajectories. The key contributions are summarized as follows:

1) We propose a novel modality-aware Clustering-based Sequence Generative Adversarial Network (CSGAN) to generate realistic human mobility data. As shown in Figure 1, we first cluster the real (training) trajectories into k clusters based on a variety of features that capture their modality. A sequence GAN is then trained where the generator generates synthetic trajectories, and the discriminator is inspired by the semi-supervised GAN and trained to classify a real trajectory into one of the k clusters (modalities) and a generated trajectory into the $(k+1)$-st ("fake") class. A reinforcement learning framework is used to train the network where we design a reward function to reward the generator for generating a trajectory that can be classified into *any one* of the k modalities (real classes).
2) To have a comprehensive evaluation of the generated trajectories, we propose three metrics to measure how well the synthetic trajectories capture the modality distribution of the real trajectories. In addition to the typical metrics that measure how well the trajectories preserve density and trajectory-level statistics, we also introduce a transition probability metric by building the transition matrix to measure how well the synthetic trajectories capture transitional information.
3) We conduct comprehensive experimental analysis on two real-world datasets with different mobility characteristics to validate the effectiveness of the proposed model. Our results show that CSGAN achieves superior results compared to state-of-the-art methods in preserving the statistical properties of the original trajectories both globally and within each modality. It also outperforms existing methods for downstream predictive tasks using the generator.

The rest of the paper is organized as follows. We first review the related work in Section 2. Then, we formulate the problem of synthetic trajectory generation in Section 3. Next, we present the proposed CSGAN framework in Section 4 and report the experimental evaluation in Section 5. Finally, we conclude the paper in Section 6.

II. Related Work

A. Synthetic Trajectory Generation

The generation of realistic human trajectories, under the category of sequence data, has been a long-standing research problem. Markov models are widely used in synthetic trajectory generation, including first-order MC [4], which constructs a transitional matrix to capture the first-order transition probability from one location to another; HMM [8], which utilizes the discrete emission probability; and IO-HMM [9], which combines transition and emission models to maximize the likelihood of observed sequences. Compared with the Markov model-based methods with simplifying assumptions, recent research resorts to model-free or deep learning methods to better capture the underlying correlations among sequence data. Deep predictive models are utilized for trajectory generation, which treat the problem as a next location prediction task given historically visited locations. For example, [5] applies Gated Recurrent Units to predict the next location given historically visited locations.

More recently, Generative Adversarial Networks (GAN)-based methods are being used and show superior performance than deep predictive models thanks to their dual generator-discriminator architecture. [10] proposes a generative model for location trajectories that can capture high-order geographic and semantic features of human mobility, such as density statistics. It uses location-based representation instead of temporal representation of trajectories, and the generator and the discriminator use Convolutional Neural Networks (CNNs), hence can not sufficiently model the sequential transitions of the trajectories. [11] presents an end-to-end LSTM-TrajGAN model to generate synthetic trajectory data, which captures the sequential information via LSTM. To better learn the sequence information for trajectories represented as discrete sequences, SeqGAN [6] proposes a reinforcement learning framework that treats the output of the discriminator as a reward sent back to the generator. [7] extends SeqGAN by leveraging the self-attention networks as the backbone of the generator and incorporating prior knowledge of human mobility patterns via the urban structure (derived from both the original training trajectories and external information such as Points of Interest (POIs)) during the generation process. Instead

of generating a discrete sequence of visits for regular time intervals, DeltaGAN [12] further extends SeqGAN to generate continuous time points and time-conditioned locations to better capture temporal irregularity in human mobility by leveraging spatiotemporal point process. While these works preserve the spatiotemporal statistics to some level, none of them consider modality information in real-life trajectories explicitly.

We focus on generating location sequences for regular time intervals in this paper and propose a novel framework that explicitly models the modality in trajectories. We show that it outperforms the state-of-the-art methods [6] [7] for generating sequences of locations, and the generated trajectories are both more representative in modalities and more realistic by capturing both cross-modality and modality-specific patterns. We note that our clustering-based framework is general and can be integrated with the extended frameworks that incorporate additional external data and generate irregular sequences, which will be interesting for future work.

B. Trajectory Clustering

Trajectory clustering is an effective method for analyzing trajectory data to detect groups of similar trajectories, e.g., consistently moving together or having similar transportation modalities or moving purposes. Existing trajectory clustering methods can be classified as: unsupervised, supervised, and semi-supervised [13]. Unsupervised methods aim to derive the hidden correlation among unlabeled trajectory data and include traditional methods such as density clustering [14], hierarchical clustering [15], and spectral clustering [16], and more recent deep learning or auto-encoder based methods [17]. Our framework uses clustering to cluster the training trajectories into different modalities and can leverage any existing clustering methods. In this paper, we experimented with several basic clustering methods based on different features derived from the trajectories to demonstrate the feasibility and advantage of our proposed framework; it would be interesting for future work to incorporate more advanced mobility behavior clustering methods with additional context information such as POIs.

III. Preliminaries

A. Problem Definition

Definition 1: ***Individual spatiotemporal trajectory.*** It is defined as a list of visiting records $Y = [y_1, y_2, y_3, ..., y_i, ..., y_n]$, where y_i denotes the i-th visit of the trajectory, which is a tuple (t_i, x_i), t_i denotes the timestamp of the i-th visit, x_i denotes the user's location of the i-th visit, which can be a geographical coordinate (lat, lon) or a region identification (ID).

Based on the above notation, the synthetic trajectory generation process with regular time intervals is defined as follows:

Definition 2: ***Synthetic trajectory generation.*** Given that each visit of the trajectory lasts for a regular time interval, the generation of each synthetic trajectory with a θ-parameterized generator can be expressed as the continuous generation of the location of each visit:

$$p_\theta(\hat{Y}) = \prod_{i=1}^{n} p_\theta(\hat{x}_i|\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{i-1}) \quad (1)$$

where p_θ denotes the probability distribution of the generator, $\hat{x}_i$ denotes the generated user's location of the i-th visit, $\hat{Y}$ denotes the generated trajectory with regular time intervals.

B. Preliminaries

Generative Adversarial Network. It consists of θ-parameterized generator G_θ and a ϕ-parameterized discriminator D_ϕ to play a "Two Player Game": the generator G_θ and discriminator D_ϕ are trained together. The generator generates a batch of trajectories, and these, along with real trajectories, are provided to the discriminator and classified as real or fake. The generator is trained to fool the discriminator in terms of being unable to distinguish the generated trajectories from the real ones (minimizing the classification accuracy of the discriminator). In contrast, the discriminator is trained to classify the real trajectories as real and generated trajectories as fake (maximizing classification accuracy). Formally, the min-max optimization objective can be expressed as:

$$\min_{G_\theta} \max_{D_\phi} E_{Y\sim p_d(Y)}[log(D_\phi(Y))] + E_{\hat{Y}\sim G_\theta(\hat{Y})}[log(1-D_\phi(\hat{Y}))] \quad (2)$$

where p_d denotes the probability distribution of the real trajectories.

IV. Proposed Framework

A. Overview

Our proposed CSGAN framework, as illustrated in Figure 1, comprises three main components: a clustering component (Section IV-B), a generator (Section IV-C), and a discriminator (Section IV-D). The clustering component groups the real (training) trajectories into k clusters based on their modalities. The generator generates synthetic trajectories, which are assigned as the $\{k+1\}$-st fake class. The discriminator functions as a multi-class classifier, taking all the real-life and synthetic trajectories from the generator as input. The discriminator is trained to classify a real trajectory into its associated cluster (one of the k real classes or modalities) and a generated trajectory into the $(k+1)$-st ("fake") class. The output of the discriminator goes through a semi-supervised reward function and is sent back to optimize the generator so that the generator is rewarded for generating a trajectory that is classified into *any one* of the k real classes. We explain each component in detail in the following subsections.

B. Modality-based Clustering

To capture the different transportation modalities in real-life trajectories, we leverage clustering to group similar trajectories together and assign them to one of the k real classes based on their cluster membership. We can leverage a variety of clustering methods based on different features such as 1) the raw location sequences, 2) derived features, 3) explicit

annotations, and 4) additional context information such as POIs associated with the locations. We present and evaluate two clustering methods based on derived features and explicit annotations in this paper as a demonstration of transportation modality. When POI information is available, we can leverage more advanced methods to cluster the trajectories into different moving behaviors or purposes.

Derived feature-based clustering. Intuitively, the most determining feature of transportation modality is the moving speed. Hence, we first cluster the trajectories based on their average moving speed directly computed from the trajectories based on consecutive locations and time elapsed. We then incorporate additional derived features, such as accumulative daily travel distance and the number of distinct visits, which may further help recognize the modality. Given the derived features, we apply the K-means Clustering algorithm [18] using Euclidean distance metric.

Explicit annotations-based clustering. For some collected trajectories (such as the PeopleFlow dataset collected in Japan), there may be explicit annotations for the transportation mode for each visit (e.g., walking, running, car). For an individual trajectory $Y = [y_1, y_2, y_3, ..., y_i, ..., y_n]$ as a sequence of visits, we have an annotation or explicit feature at each visit y_i. In this way, trajectory Y can be represented as a feature vector $[f_1, f_2, f_3, ..., f_i, ..., f_n]$ with f_i denoting the modality of the i-th visit. We use the Jaccard distance as the distance metric, as each element in the feature vector represents a categorical modality, to conduct clustering.

C. Generator

We leverage Recurrent Neural networks (RNNs) as the backbone of our generator G_θ to generate synthetic trajectories while capturing the sequential transition patterns. Assuming that the location visits have regular time intervals (the consecutive locations can be the same, indicating the person is not moving in that interval), G_θ is tasked to generate a sequence of location visits. It first generates the starting location by randomly selecting from the entire probability distribution of the locations. Then, the selection of the next location $\hat{x}_i$ is based on the previously generated locations $\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{i-1}$. G_θ consists of an embedding function $e(\cdot)$ to map the sequence of previously generated locations into embedding representations, a mapping function $g(\cdot)$ to map the embedded sequence into hidden states, and finally, a predicting function $z(\cdot)$ to map the hidden states to the probability distribution of locations, which can be written as:

$$\begin{aligned} p(\hat{x}_i|\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{i-1}) &= G_\theta(\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{i-1}) \\ &= z(g(e(\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{i-1}))) \end{aligned} \tag{3}$$

After obtaining the embedding representations, the proposed function $g(\cdot)$, which is the Gated Recurrent Unit (GRU), maps the embedding representations of the previously generated locations to a sequence of hidden states $h_1, h_2, h_3, ..., h_{i-1}, h_i$, which can be written as:

$$h_i = g(h_{i-1}, e(\hat{x}_{i-1})) \tag{4}$$

Finally, the predicting function $z(\cdot)$ maps the h_i into the output probability distribution of locations with a softmax output later to determine the most probable next location, which can be expressed as:

$$p(\hat{x}_i|\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{i-1}) = z(h_i) \tag{5}$$

D. Discriminator

Given that we obtain the cluster label for each real trajectory by splitting the entire real trajectories into k clusters, each trajectory is assigned a distinct modality corresponding to one of the k real classes. For the synthetic trajectories from the generator, we assign them the $(k+1)$-st label denoting the fake class. Our proposed discriminator D_ϕ functions as a multi-class classifier that aims to distinguish 1) whether a trajectory is real or fake and 2) given it is real, the specific class out of the k real classes it belongs to. More specifically, given a trajectory generated from the generator, D_ϕ aims to classify it into the $(k+1)$-st fake cluster; given a trajectory from the real ones, D_ϕ aims to classify it into the specific modality it belongs to among the k real classes.

To capture the complete sequence information, we first leverage bidirectional Recurrent Neural Networks (RNNs) to comprehensively evaluate the input trajectory, and then followed by dense layers to output the probability of being classified into each class, which can be written as:

$$p_D(Y) = D_\phi(Y) = z_d(g_d(e_d(Y))) \tag{6}$$

where $p_D(Y)$ denotes the output probability distribution of $k+1$ classes corresponding to input trajectory Y, $e_d(\cdot)$ denotes an embedding function, $g_d(\cdot)$ denotes a mapping function, and $z_d(\cdot)$ denotes a predicting function.

Given an input trajectory, similar to our proposed generator G_θ, our discriminator D_ϕ first consists of an embedding function $e_d(\cdot)$, which takes the locations of the input trajectory Y and outputs the embedded representations. Then, D_ϕ leverages a function $g_d(\cdot)$, a bidirectional GRU, to map the embedding representations of the locations to the hidden state. Finally, the predicting function $z(\cdot)$, which consists of a stack of fully connected layers, maps the hidden state into the output probability distribution of the $k+1$ classes to determine the most probable class.

E. Model Training

Reinforcement Learning-based Training for Diverse Modality. The traditional training algorithm of GANs via gradient back-propagation does not perform well due to the discrete nature of the generator's output [7]. Thus, we adopt the reinforcement learning approach [6] to address this issue. More specifically, we treat our proposed generator as the agent, the group of currently generated locations as the state, generating the next location based on the previously generated locations as the action, and the probability of "fooling" the discriminator as the reward. Our generator $G_\theta(\hat{x}_i|\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{i-1})$ aims to maximize its expected end reward.

To support the generation of diverse trajectories of different modalities, we propose a novel reward function based on the output of the discriminator. According to Equation 6, the output of the discriminator is the probability distribution of $k+1$ classes (k real class and the $(k+1)$-st fake class) given an input trajectory Y. Thus, the summation of the probability for a generated trajectory $\hat{Y}$ corresponding to the k real classes represents the probability of "fooling" the discriminator and thus should be treated as the reward, which can be written as:

$$R_D(\hat{Y}) = \sum_{c \in C_r} p_D^c(\hat{Y}) \tag{7}$$

where $R_D(\hat{Y})$ denotes the reward gained from the generated trajectory $\hat{Y}$ based on the output of the discriminator, C_r denotes the group of k real classes, and $p_D^c(\hat{Y})$ denotes the probability of the trajectory $\hat{Y}$ being classified into class c by the discriminator. In other words, the generator is being rewarded not for a particular modality but being rewarded as long as it generates a trajectory that looks like *any* real modalities. Alternatively, we can also create a conditioned generator where we can input a desired modality and then use the output probability corresponding to the desired class (modality) by the discriminator as the reward if we need to generate trajectories corresponding to specific modalities.

For the discriminator, since the objective is to minimize the multi-class classification loss by classifying the real trajectories into one of the k real classes and generated trajectories as fake, we have:

$$\min_\phi -E_{Y \sim p_d(Y)}[\log(p_D^{c_t}(Y))] - E_{\hat{Y} \sim G_\theta(\hat{Y})}[log(1 - \textstyle\sum_{c \in C_r} p_D^c(\hat{Y}))] \tag{8}$$

where $p_D^{c_t}(Y)$ denotes the probability of the discriminator to classify the trajectory Y into its associated class c_t given a real trajectory, C_r denotes the group of k real clusters, and $p_D^c(\hat{Y})$ denotes the probability of the discriminator to classify the generated trajectory $\hat{Y}$ into cluster c.

Model Pre-training. Due to the complicated nature of human mobility data, training a powerful generator with a large number of parameters is time-consuming. Thus, to accelerate the training process and improve the overall model's performance, we perform model pre-training on both the generator and the discriminator following the previous work [7]. We pre-train the generator with a part of the real trajectories via maximum likelihood estimation (MLE) by minimizing the negative log-likelihood loss between the generated and real ones. To pre-train the discriminator, we equally mix the real trajectories with the generated trajectories and minimize the negative log-likelihood loss between the predicted cluster labels and the ground-truth cluster labels (one of the k clusters for the real trajectory and the $(k+1)$-st cluster for the generated trajectory). Our entire CSGAN algorithm is illustrated in Algorithm 1.

Algorithm 1: CSGAN for Modality-Aware Synthetic Trajectory Generation

Data: Real set of trajectories $\mathbf{Y}$, noise distribution $P_\mathbf{Z}$, number of clusters (modalities) k, batch size b, total number of iterations T, number of iterations T_G to train the generator, number of iterations T_D to train the discriminator

Initialize parameters of the generator G_θ and the discriminator D_ϕ;
Perform clustering on $\mathbf{Y}$ and obtain k centroids;
Pre-train G_θ via MLE using a subset of $\mathbf{Y}$;
Pre-train D_ϕ via minimizing the negative log-likelihood loss;
for $t = 1 : T$ **do**
  for $t = 1 : T_G$ **do**
    Use G_θ to generate b synthetic trajectories $\{G_\theta(\mathbf{z}_i)\}_{i=1}^b$ from $P_\mathbf{Z}$;
    Assign them to "fake" (the $(k+1) - st$ class);
    Compute the reward of the b generated trajectories via Equation 7 and update θ via policy gradient;
  end
  for $t = 1 : T_D$ **do**
    Sample b real trajectories $\{Y_i\}_{i=1}^b$ from $\mathbf{Y}$;
    Obtain their cluster labels w.r.t to the k centroids;
    Use G_θ to generate b synthetic trajectories $\{G_\theta(\mathbf{z}_i)\}_{i=1}^b$ from $P_\mathbf{Z}$ and assign them the fake label;
    Update D_ϕ w.r.t the NLL via Equation 8
  end
end

V. Evaluation

We conduct comprehensive experiments utilizing real-world datasets and aim to answer the following questions:

RQ1. With the modality-aware clustering-based generation, how effective is our CSGAN model in generating realistic synthetic trajectories compared with the state-of-the-art approaches?

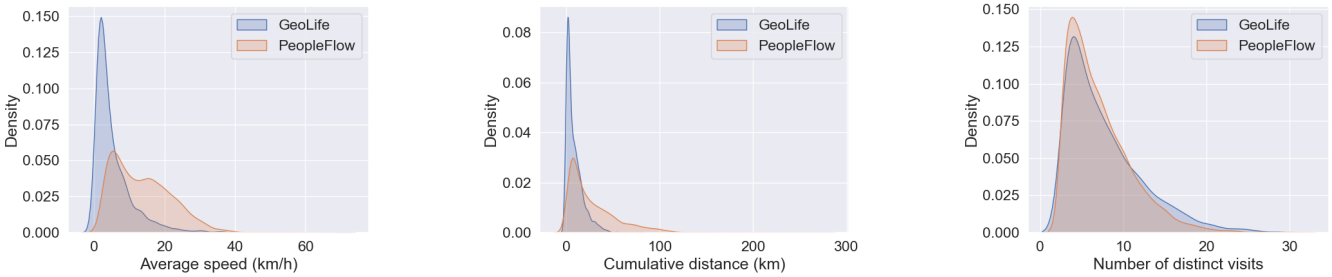
RQ2. How do different clustering strategies impact our proposed CSGAN framework?

RQ3. How does CSGAN perform on the downstream task, e.g., next location prediction, compared to the state-of-the-art approaches?

A. Experimental Setup

Data. We experiment on two real-world datasets with different characteristics (open GeoLife Dataset [19] and semi-open PeopleFlow Dataset [20]) to verify the effectiveness of our proposed model.

- GeoLife Dataset: This GPS trajectory dataset was collected in the (Microsoft Research Asia) Geolife project by 182 users over three years (from April 2007 to August 2012). We select a portion of the entire GeoLife data for evaluation (2756 daily trajectories in 2008 from 6:00 am to 8:00 pm with 15 minutes as the time interval, i.e., each trajectory has 56 visiting locations)
- PeopleFlow Dataset: This data is based on 2008 Tokyo Metropolitan Area PT Data (provided by Tokyo Metropolitan Circle Transportation Planning Council) and is lent by the University of Tokyo CSIS. We select a portion of the entire PeopleFlow data via the same processing technique as GeoLife, resulting in 6183 trajectories.



(a) Average speed (b) Accumulative distance (c) Number of distinct visit

Fig. 2: Dataset comparison (GeoLife and Peopleflow)

TABLE I: GeoLife and PeopleFlow datasets

Characteristics	GeoLife	PeopleFlow
Number of Trajectories	2756	6183
Period	1 year (2008)	1 year (2008)
Visit Interval	every 15 minutes	every 15 minutes
Average Speed (km/h)	5.324±5.744	13.592±8.303
Average Accumulated Distance (km)	8.401±9.340	27.662±26.494
Average Distinct Visits	7.713±4.558	7.123±3.895

We show the dataset characteristics in Table I, and a detailed view of the distribution of average speed, accumulative distance, and the number of distinct visits per trajectory in each dataset in Figure 2.

Comparison Methods. We compare CSGAN with the state-of-the-art methods SeqGAN and MoveSim as well as a baseline Cluster-wise SeqGAN.

- SeqGAN [6]: it utilizes reinforcement learning and Monte Carlo search to generate discrete sequences of trajectories.
- MoveSim [7]: it extends SeqGAN, utilizes self-attention networks as the generator, and incorporates additional urban structures to regularize the generation via mobility regularity. We note that the original work includes three kinds of urban structures: the physical distance between all location pairs, functional similarity between locations based on the correlation between the POI distribution, and historical transitions between locations. The first and the third can be directly computed from the training trajectories, while the second POI information is an ancillary attribute unavailable from the datasets. Thus, we implement MoveSim without the second urban structure for a fair comparison.
- Cluster-wise SeqGAN: we also implement a baseline cluster-wise SeqGAN to consider modality, which conducts clustering on the real trajectories and then trains an individual SeqGAN model on each cluster. While the above two represent the approaches that learn global patterns without considering modality, this represents the approaches that learn only modality-specific patterns.

For all the methods, we also perform clustering on the original and generated trajectories and evaluate how the cluster (modality) distributions match (see Evaluation Metrics later in this section). We experiment with different clustering strategies using different sets of features, including 1) a single derived feature of average speed (dubbed with "-S"), 2) multiple derived features including average speed, travel distance, and distinct visits (dubbed with "-M") for the GeoLife dataset, and 3) explicit per-visit annotations of transportation modes that are available for the PeopleFlow dataset (dubbed with "-E").

Implementation Details. Our CSGAN model leverages clustering to divide real trajectories into k categories. To determine the optimal k for each dataset, we leverage an extended elbow method [21]. For clustering based on the single feature average speed, we set $k = 4$ for the GeoLife Dataset (likely corresponding to walking, biking, bus, and car) and $k = 6$ for PeopleFlow Dataset (due to mixed transportation modes, e.g., walking and bus). For clustering based on multiple derived features, we set $k = 7$ for the GeoLife Dataset. Finally, for clustering based on the explicit annotations, we set $k = 4$ for the PeopleFlow Dataset.

The generator and discriminator are configured with the embedding size and hidden dimension of 32 and 64, respectively. The generator is pre-trained for 150 epochs, and the discriminator is pre-trained for 75 epochs. A dropout of 0.2 is applied, and adversarial training lasts for 75 epochs, with a learning rate of $1e^{-2}$ and batch size of 32.

Evaluation Metrics. We evaluate the quality of the synthetic trajectories by verifying whether the various statistical properties at the geographical, individual trajectory, transition, and modality levels are preserved.

1) Geographical density-based statistics:

- $P(r)$: Probability of a trajectory visiting location r.
- $P(r, t)$: Probability of a trajectory visiting location r at time t.

2) Individual trajectory level statistics:

- $P(d)$: Probability of the accumulated distance of a trajectory being d.
- $P(v)$: Probability of the number of distinct visits of a trajectory being v.

Following previous work, we compute the Jensen-Shannon Divergence between the probability distribution of the real trajectories and that of generated trajectories for each of the

above distributions, which can be written as:

$$JSD(P_{re}||P_{gen}) = H(\frac{P_{re}+P_{gen}}{2}) - \frac{H(P_{re})+H(P_{gen})}{2} \tag{9}$$

where P_{re} and P_{gen} are the two probability distributions for real and generated trajectories, respectively, and H is the Shannon information. The lower the divergence, the better the generated trajectories preserving the original distributions.

3) Transition statistics:

- $P(r_1, r_2)$: Probability of a trajectory transitioning from location r_1 to location r_2. While a main goal of the synthetic trajectory generation is to learn and preserve the sequential information, most existing works do not evaluate how the sequential transition probability is preserved. Given the entire Q regions, we build the transitional matrix $P \in \mathbb{R}^{Q*Q}$ for both real and synthetic trajectories, where the element corresponding to row r_1 and column r_2 of the matrix denotes the transition probability from location r_1 to location r_2. Then we take the Frobenius norm of the difference between the two transition matrices:

$$||P_r - P_g||_F = \sqrt{\sum_{r_1=1}^{Q}\sum_{r_2=1}^{Q}|P_r(r_1,r_2) - P_g(r_1,r_2)|^2} \tag{10}$$

  where P_r and P_g denote the transition matrix of the real and generated trajectories, respectively. The lower the norm, the better the transition is preserved.

4) Modality level statistics:

- $P(c_i^0)$: Proportion of trajectories within each modality (cluster c_i) with respect to the centroids from the real trajectories. In other words, for generated trajectories, we assign each of them to the nearest centroid from the real trajectories. Then, we compute the JSD between the two distributions.
- $P(c_i^1)$: The difference between this one and the above is that we perform clustering on the generated trajectories separately and find a set of synthetic centroids (which might be different from those from the real trajectories). $P(c_i^1)$ denotes the proportion of trajectories within each modality using the corresponding centroid in real and synthetic trajectories, respectively. The centroids are matched and ordered as explained below. We also compute the JSD between the two distributions.
- C: Cluster centroids or modality representatives. Given the vector of k centroids from real trajectories and generated trajectories, we compute the minimum accumulated pair-wise distance among all permutations (closest match). The lower the value, the better the generated trajectories preserve the modality representatives.

B. RQ1: Effectiveness Comparison with the Baselines

Global Comparison. We show the evaluation metrics for different methods for the overall dataset in Table II. We first dive into the results with clustering based on average speed. Our proposed model CSGAN performs consistently the best over all the metrics on both datasets. More specifically, on GeoLife data, for geographical and trajectory statistics, CSGAN exceeds the baselines on average 33% in $P(r)$, 19% in $P(r,t)$, 25% in $P(d)$, and 15% in $P(v)$; for transitional probability, CSGAN outperforms the baselines on average 53% in $P(r_1, r_2)$; for modality patterns, CSGAN excels over the baselines on average 49% in $P(c_i^0)$, 76% in $P(c_i^1)$, and 53% in C. Similarly, on PeopleFlow data, CSGAN excels over the baselines on average 20% in $P(r)$, 9% in $P(r,t)$, 48% in $P(d)$, and 46% in $P(v)$; 49% in $P(r_1, r_2)$; 56% in $P(c_i^0)$, 59% in $P(c_i^1)$, and 53% in C.

As expected, the highest performance gain is observed for the modality-level metrics, which demonstrates the power of our method in capturing and representing the modalities in the generated trajectories. Moreover, we also observe a significant performance gain on the transitional probability metrics. The explanation is intuitive: given a trajectory with a specific modality, e.g., walking, a user cannot travel a large distance, and thus, there is a limited number of potential destinations. By our modality-aware generation, the modality-specific transition can be better learned and preserved.

Modality-specific Comparison. In addition to the overall comparison, we also show the comparison for each modality for both datasets to verify whether the trajectories within each modality are realistic. Table III shows the results. On both GeoLife and PeopleFlow data, CSGAN consistently outperforms the baselines over all the metrics for each modality. For instance, in cluster 2 of GeoLife, CSGAN excels over the baselines on average 29% in $P(r)$, 16% in $P(r,t)$, 57% in $P(d)$, 44% in $P(v)$, and 62% in $P(r_1, r_2)$. In cluster 1 of PeopleFlow, CSGAN shows improvements of 19% in $P(r)$, 15% in $P(r,t)$, 36% in $P(d)$, 25% in $P(v)$, and 30% in $P(r_1, r_2)$. This verifies that CSGAN learns not only global patterns but also modality-specific patterns across all modalities.

Baseline Comparison. Comparing the baseline approaches with each other, SeqGAN outperforms MoveSim for most metrics. This can be due to two reasons: 1) most of the performance gain of MoveSim, as reported in the original work, may be due to the auxiliary POI information (which we did not use for a fair comparison), 2) MoveSim may require a large training dataset due to its more complex model architecture and the training data in our experiments is smaller than that used in the original MoveSim work (1 year vs. 5 years). Cluster-wise SeqGAN, while achieving the best performance for preserving the distinctive visits $P(v)$, does not perform as well as SeqGAN in general because it only learns from each modality without learning from the patterns shared among different modalities. In summary, this verifies that by capturing both the global and modality-specific patterns, CSGAN is able to generate trajectories that are both 1) more diverse and representative as reflected in the modality metrics and 2) more realistic as reflected in the trajectory and transitional metrics.

TABLE II: Global comparison with baselines on GeoLife and PeopleFlow data with different clustering techniques. The table shows the average statistics of 5 experiments. The best performance is in boldface. The second-best is underlined.

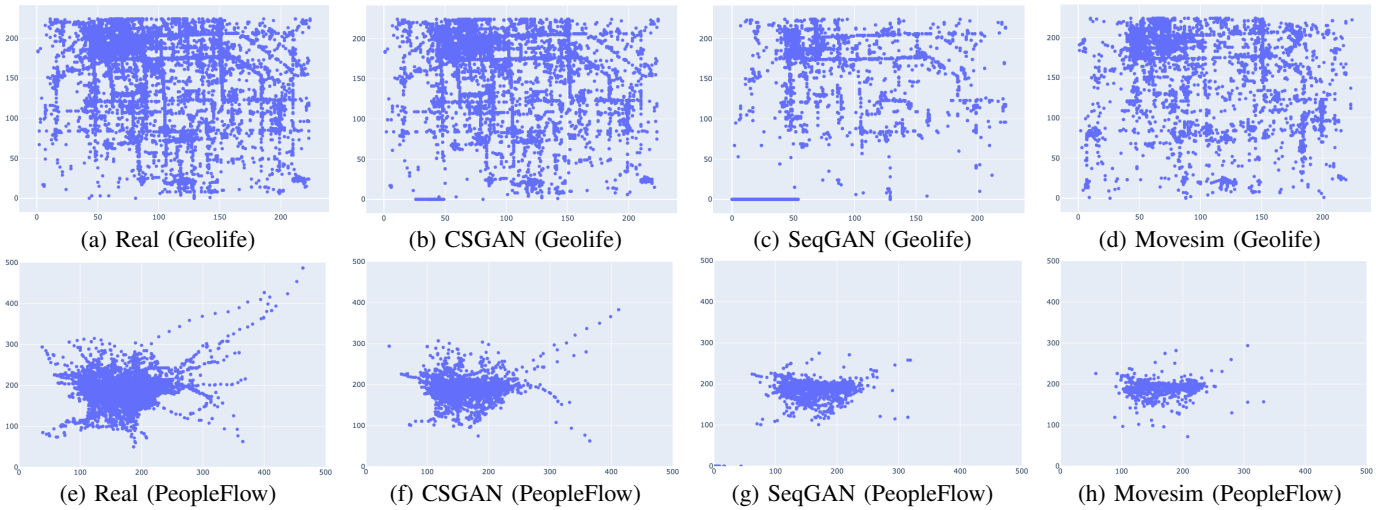
	Geographical density-based statistics		**Individual trajectory level statistics**		**Transition statistics**	**Modality level statistics**		
Methods	P(r)	P(r,t)	P(d)	P(v)	P(r1, r2)	$P(c_i^0)$	$P(c_i^1)$	C
GeoLife								
SeqGAN-S	0.407	0.478	0.208	0.288	0.100	0.162	0.220	18.682
Cluster-wise SeqGAN-S	0.506	0.562	0.313	**0.234**	0.082	0.402	0.298	83.441
Movesim-S	0.522	0.579	0.263	0.390	0.116	0.303	0.209	**7.934**
CSGAN-S	**0.319**	**0.439**	**0.195**	0.258	**0.047**	**0.147**	**0.058**	17.128
SeqGAN-M	0.407	0.478	0.208	0.288	0.100	0.678	0.111	48.384
Movesim-M	0.522	0.579	0.263	0.390	0.116	0.341	0.134	59.993
CSGAN-M	**0.289**	**0.354**	**0.098**	**0.120**	**0.032**	**0.197**	**0.065**	**41.055**
PeopleFlow								
SeqGAN-S	0.378	0.437	0.368	0.275	0.092	0.406	0.167	29.694
Cluster-wise SeqGAN-S	0.344	0.406	0.363	0.317	0.105	0.311	0.194	31.046
Movesim-S	0.344	0.396	0.524	0.602	0.100	0.289	0.151	23.420
CSGAN-S	**0.284**	**0.376**	**0.218**	**0.215**	**0.050**	**0.146**	**0.070**	**13.136**
SeqGAN-E	0.378	0.437	0.368	0.275	0.092	0.117	0.250	5.000
Movesim-E	0.344	0.396	0.524	0.602	0.100	0.338	0.401	**3.000**
CSGAN-E	**0.288**	**0.380**	**0.244**	**0.216**	**0.040**	**0.083**	**0.144**	**3.000**

TABLE III: Modality-specific comparison with baselines on GeoLife and PeopleFlow data with clustering based on the derived global feature **average speed**. The table shows the average statistics of 5 experiments. The best performance is in boldface. The second-best is underlined.

				Geographical density-based statistics		**Individual trajectory level statistics**		**Transition statistics**
Cluster	Centroid Speed	Proportion	Method	P(r)	P(r,t)	P(d)	P(v)	P(r1, r2)
GeoLife								
Cluster_1	2.498	55%	SeqGAN-S	0.505	0.555	0.297	0.432	0.191
			Cluster-wise SeqGAN-S	0.607	0.633	0.496	**0.276**	0.135
			CSGAN-S	**0.403**	**0.487**	**0.237**	0.402	**0.058**
Cluster_2	7.314	34%	SeqGAN-S	0.507	0.589	0.225	0.169	0.075
			Cluster-wise SeqGAN-S	0.563	0.626	0.336	0.252	0.140
			CSGAN-S	**0.382**	**0.509**	**0.120**	**0.118**	**0.041**
Cluster_3	14.784	10%	SeqGAN-S	0.540	0.597	**0.293**	**0.210**	0.097
			Cluster-wise SeqGAN-S	0.700	0.721	0.477	0.335	0.248
			CSGAN-S	**0.507**	**0.586**	0.319	0.229	**0.092**
Cluster_4	30.747	1%	SeqGAN-S	0.650	0.672	0.459	0.358	0.358
			Cluster-wise SeqGAN-S	**0.487**	**0.560**	0.705	0.423	0.335
			CSGAN-S	0.632	0.660	**0.446**	**0.333**	**0.328**
PeopleFlow								
Cluster_1	4.497	27%	SeqGAN-S	0.460	0.482	0.576	0.597	0.160
			Cluster-wise SeqGAN-S	0.364	0.408	0.548	0.407	0.104
			CSGAN-S	**0.334**	**0.379**	**0.360**	**0.374**	**0.092**
Cluster_2	9.141	21%	SeqGAN-S	0.512	0.558	0.411	0.347	0.155
			Cluster-wise SeqGAN-S	0.424	0.475	0.454	0.339	0.093
			CSGAN-S	**0.361**	**0.438**	**0.243**	**0.215**	**0.080**
Cluster_3	14.590	20%	SeqGAN-S	0.553	0.614	0.382	0.251	0.109
			Cluster-wise SeqGAN-S	0.492	0.557	0.492	0.368	0.124
			CSGAN-S	**0.433**	**0.521**	**0.264**	**0.162**	**0.072**
Cluster_4	19.924	17%	SeqGAN-S	0.585	0.648	0.460	0.338	0.165
			Cluster-wise SeqGAN-S	0.511	0.586	0.442	0.265	0.129
			CSGAN-S	**0.451**	**0.556**	**0.258**	**0.146**	**0.073**
Cluster_5	25.868	11%	SeqGAN-S	0.564	0.639	0.432	0.273	0.122
			Cluster-wise SeqGAN-S	0.540	0.622	0.546	0.321	0.135
			CSGAN-S	**0.471**	**0.587**	**0.325**	**0.181**	**0.087**
Cluster_6	34.627	4%	SeqGAN-S	0.576	0.654	0.461	0.220	0.114
			Cluster-wise SeqGAN-S	0.628	0.673	0.716	0.396	0.267
			CSGAN-S	**0.553**	**0.646**	**0.455**	**0.184**	**0.093**

TABLE IV: Comparison with baselines on GeoLife and PeopleFlow data on the task of next location prediction. The best performance is in boldface.

Model	Accuracy@1	Accuracy@2	Accuracy@3	Accuracy@4	Accuracy@5	Accuracy@6	Accuracy@7	Accuracy@8
GeoLife								
SeqGAN	0.842	0.913	0.934	0.944	0.951	0.956	0.959	0.963
Movesim	0.612	0.641	0.653	0.662	0.665	0.670	0.673	0.678
CSGAN	**0.880**	**0.930**	**0.944**	**0.954**	**0.960**	**0.964**	**0.967**	**0.970**
Peopleflow								
SeqGAN	0.831	0.882	0.905	0.916	0.927	0.932	0.937	0.941
Movesim	0.815	0.822	0.826	0.831	0.835	0.839	0.841	0.843
CSGAN	**0.888**	**0.912**	**0.921**	**0.927**	**0.933**	**0.936**	**0.940**	**0.942**



(a) Real (Geolife) (b) CSGAN (Geolife) (c) SeqGAN (Geolife) (d) Movesim (Geolife)

(e) Real (PeopleFlow) (f) CSGAN (PeopleFlow) (g) SeqGAN (PeopleFlow) (h) Movesim (PeopleFlow)

Fig. 3: Population density (GeoLife and Peopleflow)

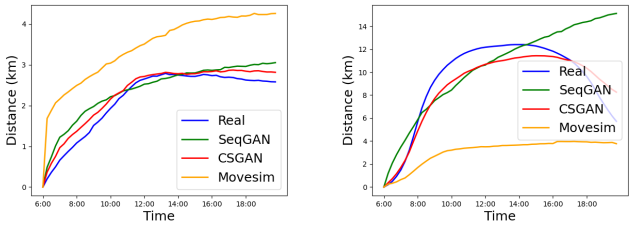


Fig. 4: Absolute distance to origin (GeoLife (left) and PeopleFlow (right))

Dataset Comparison. Comparing the GeoLife and PeopleFlow Datasets, GeoLife is less diverse in modality compared to PeopleFlow, as shown in Figure 2. This difference contributes to the different orders of performance in Table II and Table III. For Geolife, SeqGAN outperforms Cluster-wise SeqGAN since the modality is less diverse, and hence it is more important to learn from the global patterns. In contrast, for PeopleFlow, which is more diverse in modality, SeqGAN tends to perform less satisfying and is surpassed by the Cluster-wise SeqGAN, which explicitly learns modality-specific patterns within each cluster.

C. RQ1: Visualization

We show several visualizations to illustrate how the methods compare with each other in preserving the patterns in the original trajectories.

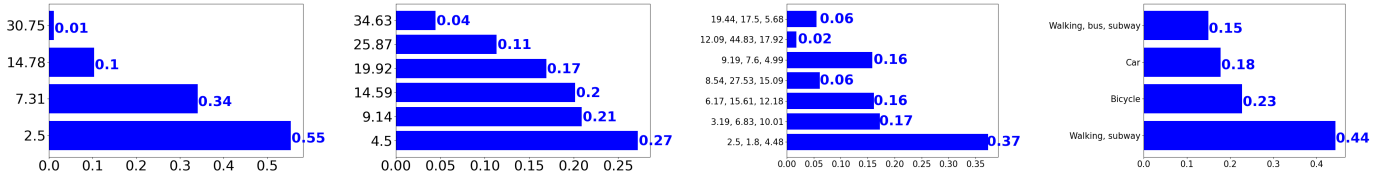
Population Density. We plot the population density (the aggregate density from 6:00 am to 8:00 pm) for GeoLife and PeopleFlow data. Figure 3 shows the density on the map (divided into grids) using real and generated trajectories for both datasets. We can see that CSGAN best mimics the real data by capturing both the overall distribution and the outliers. Places with high density and low density are both captured by our model. In contrast, the baseline models fail to capture such information and tend to lose both the overall distribution (GeoLife) and outliers (PeopleFlow).

Absolute Distance to Origin. We plot the absolute distance to the starting point from 6:00 am to 8:00 pm for every 15 minutes using the real and synthetic trajectories in Figure 4. We can see that CSGAN can better capture the moving behaviors: for GeoLife data, people tend to be far away from their starting location till 1:00 pm and then stay around the same region till 8:00 pm. For PeopleFlow data, people tend to be away from the starting point from 6:00 am to 4:00 pm and then return to the origin (typically their home), which is reflected by the downward trend of the blue curve. In both cases, CSGAN (red) closely follows the trend of the real data (blue), while both SeqGAN and MoveSim deviate from the trends.

D. RQ2: Impact of Different Clustering Methods

Next, we explore the impact of different clustering techniques on the performance of CSGAN against baselines. In Table II, methods with "-S" denotes clustering based on a single feature (average speed); methods with "-M" denotes clustering based on multiple features (average speed, accumulative distance, the number of distinct visits); and methods with "-E" denotes clustering based on explicit annotations for each visit (transportation mode) provided by the PeopleFlow dataset. Figure 5 shows the proportion of trajectories within each cluster via these clustering methods, respectively. For example, PeopleFlow demonstrates several clusters of both single and mixed modalities: (Walking, bus, subway), Car, Bicycle, (Walking, subway).

Table II shows that CSGAN with different clustering methods consistently outperforms the baselines. Moreover, we find that with more sophisticated clustering techniques or features, as shown by switching from a single feature (average speed) to multiple features, CSGAN tends to be more powerful. For the PeopleFlow dataset, the explicit annotation offers some performance gain in some metrics, at the same time, also verifies that clustering based on the trajectories alone without the annotations indeed captures the modality well.



(a) Single feature (GeoLife) (b) Single feature (PeopleFlow) (c) Multiple features (GeoLife) (d) Explicit features (PeopleFlow)

Fig. 5: The proportion of trajectories in each cluster. The x-axis denotes the proportion of trajectories, and the y-axis denotes the feature of each cluster's centroid. (a) and (b) are based on clustering via average speed. (c) is based on average speed, accumulative distance, and number of distinct visits (from left to right), and (d) is based on transportation modes vector.

E. RQ3: Next Location Prediction

To further verify the utility of our CSGAN model, we study the next location prediction as a downstream task using the trained generator from CSGAN and baseline methods. We leverage the metric: accuracy@k, which denotes whether the ground-truth next location exists in top k predicted locations given the predicted probability distribution of the entire Q locations from the generator. We set k from 1 (the ground-truth location is exactly the predicted next location) to 8. The test data is selected from the real trajectories.

We report the next location prediction results in Table IV. CSGAN consistently outperforms SeqGAN and MoveSim on both datasets, exceeding the best baseline by 5% on GeoLife and 7% on PeopleFlow when $k = 1$. More importantly, as k gets smaller, our model's advantage is more obvious. This further verifies the advantage of CSGAN in learning the sequential patterns from the data by capturing the modality.

VI. Conclusion

We proposed a novel and general framework, the modality-aware Clustering-based Sequence Generative Adversarial Network (CSGAN), which can generate representative and realistic synthetic trajectories by capturing real-world modalities. CSGAN leverages clustering and adopts semi-supervised losses to capture real-life modality patterns in a GAN setting and a novel reward function for training the network via reinforcement learning. To comprehensively evaluate the quality of the synthetic trajectories, we introduce several new metrics to measure how the synthetic trajectories preserve transitional and modality properties in addition to the typical density and trajectory level properties. Experiments on two real-world datasets demonstrate the consistent and superior performance of CSGAN. Our future works include incorporating clustering methods for moving behaviors, integrating the framework with other methods that consider additional contextual information, such as POIs and temporal irregularity of the trajectories, and extending the framework to model co-movements via multi-agent reinforcement learning.

References

[1] E. Toch, B. Lerner, E. Ben-Zion, and I. Ben-Gal, "Analyzing large-scale human mobility data: a survey of machine learning methods and applications," *Knowledge and Information Systems*, vol. 58, pp. 501–523, 2019.

[2] T. Gan, W. Li, L. He, and J. Li, "Intracity pandemic risk evaluation using mobile phone data: The case of shanghai during covid-19," *ISPRS International Journal of Geo-Information*, vol. 9, no. 12, p. 715, 2020.

[3] F. Giannotti and D. Pedreschi, *Mobility, data mining and privacy: Geographic knowledge discovery*. Springer Science & Business Media, 2008.

[4] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating next-cell predictors with extensive wi-fi mobility data," *IEEE transactions on mobile computing*, vol. 5, no. 12, pp. 1633–1649, 2006.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[6] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[7] J. Feng, Z. Yang, F. Xu, H. Yu, M. Wang, and Y. Li, "Learning to simulate human mobility," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 3426–3433.

[8] J. Krumm, E. Horvitz *et al.*, "Locadio: Inferring motion and location from wi-fi signal strengths." in *mobiquitous*, 2004, pp. 4–13.

[9] M. Yin, M. Sheehan, S. Feygin, J.-F. Paiement, and A. Pozdnoukhov, "A generative model of urban activities from cellular data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1682–1696, 2017.

[10] K. Ouyang, R. Shokri, D. S. Rosenblum, and W. Yang, "A non-parametric generative model for human trajectories." in *IJCAI*, vol. 18, 2018, pp. 3812–3817.

[11] J. Rao, S. Gao, Y. Kang, and Q. Huang, "Lstm-trajgan: A deep learning approach to trajectory privacy protection," 2020. [Online]. Available: https://arxiv.org/abs/2006.10521

[12] N. Xu, L. Trinh, S. Rambhatla, Z. Zeng, J. Chen, S. Assefa, and Y. Liu, "Simulating continuous-time human mobility trajectories."

[13] J. Bian, D. Tian, Y. Tang, and D. Tao, "A survey on trajectory clustering analysis," *arXiv preprint arXiv:1802.06971*, 2018.

[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[15] A. Gaidon, Z. Harchaoui, and C. Schmid, "Activity representation with motion hierarchies," *International journal of computer vision*, vol. 107, no. 3, pp. 219–238, 2014.

[16] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognition*, vol. 41, no. 3, pp. 1012–1029, 2008.

[17] M. Yue, Y.-Y. Chiang, and C. Shahabi, "Vambc: A variational approach for mobility behavior clustering," in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part IV 21*. Springer, 2021, pp. 453–469.

[18] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[19] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li, *Geolife GPS trajectory dataset - User Guide*, geolife gps trajectories 1.1 ed., July 2011, geolife GPS trajectories 1.1. [Online]. Available: https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/

[20] Y. Sekimoto, R. Shibasaki, H. Kanasugi, T. Usui, and Y. Shimazaki, "Pflow: Reconstructing people flow recycling large-scale social survey data," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 27–35, 2011.

[21] P. Bholowalia and A. Kumar, "Ebk-means: A clustering technique based on elbow method and k-means in wsn," *International Journal of Computer Applications*, vol. 105, no. 9, 2014.