Computers, Materials &
Continua

Tech Science Press

Doi:10.32604/cmc.2025.069208

ARTICLE

DPIL-Traj: Differential Privacy Trajectory Generation Framework with Imitation Learning

Huaxiong Liao[1,2], **Xiangxuan Zhong**[2], **Xueqi Chen**[2], **Yirui Huang**[3], **Yuwei Lin**[2], **Jing Zhang**[2,*] and **Bruce Gu**[4]

[1]Faculty of Data Science, City University of Macau, Macau, 999078, China
[2]School of Computer Science and Mathematics, Fujian Provincial Key Laboratory of Big Data Mining and Applications, FujianUniversity of Technology, Fuzhou, 350118, China
[3]Fujian Province Key Laboratory of Information Security and Network Systems, College of Computer Science and Big Data, Fuzhou University, Fuzhou, 350108, China
[4]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, 250353, China

*Corresponding Author: Jing Zhang. Email: jing165455@126.com

Received: 17 June 2025; Accepted: 01 September 2025; Published: 10 November 2025

ABSTRACT: The generation of synthetic trajectories has become essential in various fields for analyzing complex movement patterns. However, the use of real-world trajectory data poses significant privacy risks, such as location re-identification and correlation attacks. To address these challenges, privacy-preserving trajectory generation methods are critical for applications relying on sensitive location data. This paper introduces DPIL-Traj, an advanced framework designed to generate synthetic trajectories while achieving a superior balance between data utility and privacy preservation. Firstly, the framework incorporates Differential Privacy Clustering, which anonymizes trajectory data by applying differential privacy techniques that add noise, ensuring the protection of sensitive user information. Secondly, Imitation Learning is used to replicate decision-making behaviors observed in real-world trajectories. By learning from expert trajectories, this component generates synthetic data that closely mimics real-world decision-making processes while optimizing the quality of the generated trajectories. Finally, Markov-based Trajectory Generation is employed to capture and maintain the inherent temporal dynamics of movement patterns. Extensive experiments conducted on the GeoLife trajectory dataset show that DPIL-Traj improves utility performance by an average of 19.85%, and in terms of privacy performance by an average of 12.51%, compared to state-of-the-art approaches. Ablation studies further reveal that DP clustering effectively safeguards privacy, imitation learning enhances utility under noise, and the Markov module strengthens temporal coherence.

KEYWORDS: Privacy-preserving; trajectory generation; differential privacy; imitation learning; Markov chain

1 Introduction

Artificial Intelligence has transformed numerous fields by enabling intelligent decision-making, automation, and data-driven insights. Its applications span healthcare, finance, transportation, and smart cities, where it aids in analyzing complex systems and optimizing resources. Within this context, trajectory data plays a vital role in practical applications. For example, in the context of smart cities, mobility trajectories can reveal the spatial-temporal patterns of human activities, offering insights into urban planning, transportation optimization, and crowd management [1]. For service providers, trajectory data enables demand forecasting, personalized service recommendations, and the design of efficient resource allocation strategies. Despite its utility, real-world trajectory data raises significant privacy concerns. Sensitive information, such as visited locations, travel routes, and activity patterns, can be inferred, leading to privacy breaches through re-identification and correlation attacks [2,3]. Privacy protection has also been studied from cryptographic perspectives, including homomorphic encryption and secure computation [4]. Thus, privacy protection enables safe use of trajectory data for applications like mobility analysis and location-based services without exposing sensitive information [5].

In recent years, machine learning methods based on techniques such as Generative Adversarial Networks (GANs) [6,7] and Variational Auto-Encoders (VAEs) [8,9] have made some progress in learning the underlying movement patterns and generating or synthetic trajectories or prediction. They typically train on real trajectory data to capture the spatio-temporal characteristics of trajectories [10]. GANs generate realistic trajectories by leveraging adversarial learning between a generator and a discriminator [11]. On the other hand, VAEs model the latent distributions of trajectories through an encoding and decoding process, enabling the generation of diverse trajectory samples [12]. However, these approaches still have significant shortcomings in the area of privacy protection. Existing studies usually assume that large-scale, centrally stored real trajectory data is available for training. However, they often fail to sufficiently consider the risk of information leakage that may arise due to the privacy sensitivity of trajectory data. Moreover, these methods perform poorly in the trade-off between privacy protection and data utility, making it difficult to balance both. Overall, three problems are evident in the existing research: (1) Data centralization and privacy breaches; (2) Risk of extrapolation during training; (3) Insufficient balance between privacy protection and practicality.

To address these challenges, this paper presents DPIL-Traj, an innovative framework that integrates differential privacy, imitation learning, and Markov chain modeling for privacy-preserving trajectory generation. Specifically, the contributions of this paper are summarized as follows:

(1) A Differential Privacy Clustering mechanism is proposed to protect sensitive user information by adding Laplace noise to the original trajectory data and performing clustering operations. This approach reduces the risk of re-identification attacks by ensuring that the clustering results do not expose sensitive spatial or temporal patterns, making the generated trajectories suitable for privacy-sensitive applications.

(2) A Imitation Learning framework for trajectory generation is designed to accurately capture the complex spatio-temporal features of trajectories and achieve high-quality trajectory simulation by incorporating trajectory shape similarity and positional accuracy into the optimization objective through a reward function. In addition, the system incorporates a generative adversarial mechanism to further enhance the realism and diversity of the generated trajectories.

(3) In addition, a Markov-based Trajectory Generation model is introduced to model the state transfer law of trajectories, and the dynamic modeling capability in the trajectory generation process is enhanced by the state transfer probability matrix. The method effectively balances the randomness and goal-orientedness of trajectory generation, so that the generated trajectories are more in line with the needs of practical application scenarios.

(4) Extensive experiments validate the effectiveness of the DPIL-Traj framework, achieving a 19.85% improvement in utility performance and a 12.51% increase in privacy performance. These findings underscore the potential of DPIL-Traj as an effective solution for privacy-preserving trajectory generation in a wide range of applications. Ablation studies further show that imitation learning delivers the greatest utility improvement, DP clustering secures privacy with minimal accuracy loss, and the Markov component enhances temporal coherence.

The remainder of this article is organized as follows: Section 2 reviews the related work. Section 3 presents the preliminary concepts. Section 4 introduces the DPIL-Traj framework. Section 5 discusses the experiments and results. Finally, Section 6 summarizes the current work and highlights potential future directions.

2 Related Work

2.1 Differential Privacy in Trajectory Data Security

Differential privacy is widely used in the distribution and protection of trajectory and location data, minimizing the leakage of sensitive user information by adding random noise. Applying differential privacy during trajectory publication prevents attackers from inferring user identity through trajectory patterns or location associations.

Several studies have explored innovative methods to enhance trajectory data security using differential privacy. For example, Yin et al. propose a privacy-preserving approach that satisfies differential privacy constraints, models a multilevel location information tree, and uses a Laplace Mechanism to add noise to the access frequency of the selection data [13]. Similarly, Zhang et al. use reinforcement learning to compute the optimal Laplacian bounds and add bounded Laplacian noise to the hashmap storing sensitive locations as a way to achieve differential privacy [14]. In another approach, Yang et al. propose a new approach to provide local distance privacy protection for users participating in cluster analysis. The user's records are mapped into a one-dimensional distance space, and the records in this distance space are made indistinguishable from each other [15]. Building on this, Yuan et al. propose a Local Differential Privacy K Prototype (LDPK) mechanism that first uses local differential privacy to perturb user data, and then completes the clustering through the interaction between the server and the user [16]. Huang et al. propose a novel local differential privacy mechanism that incorporates Geo-aware grid techniques, which use the piecewise mechanism to perturb the user's trajectory before training the synthesized trajectory [17]. Furthermore, Sun et al. propose a novel solution for synthesizing private and realistic trajectories, defining moveable constraints and integrating them into differential private trajectory synthesis to generate realistic trajectories, which can be treated as a post-processing of differential privacy mechanism [18]. Additionally, Wang et al. address publishing correlated non-numerical data under differential privacy by modeling attribute correlations to enhance utility while preserving privacy. This work extends differential privacy applications beyond numerical data and complements existing methods [19].

However, these above methods perturb the trajectory points or paths by adding noise, which protects user privacy but also leads to significant distortion of the trajectory shape and weakens the utility of the trajectory data. For example, in trajectory clustering or pattern extraction tasks, the cumulative effect of noise may completely mask the spatial characteristics of the trajectories. This can affect the reliability of the analysis results. In contrast, this paper preprocesses trajectory data through differential privacy clustering. It uses cluster centers instead of the original trajectory points, reducing the impact of noise on the shape of the trajectory. In addition, the system combines imitation learning optimization in the trajectory generation stage, which further improves the spatial accuracy and practicality of the generated trajectories.

2.2 Imitation Learning in Trajectory Generation

Imitation learning is widely used to learn decision rules for generators from expert trajectories with the goal of generating or predicting trajectories with high similarity and utility. Generative Adversarial Imitation Learning (GAIL) combines Generative Adversarial Networks (GANs) with imitation learning to be able to generate realistic spatio-temporal trajectories that capture the dynamic features of trajectories. The similarity between the generated trajectories and the real trajectories is measured by optimizing the reward function, which is usually defined by metrics such as trajectory shape characteristics and spatial distribution.

Imitation learning has emerged as a promising approach for generating synthetic trajectories while addressing privacy concerns. For example, a Generative Adversarial Imitation Learning (GAIL) approach is proposed to discover privacy data security risks in the Industrial Internet of Things (IoT) by training privacy-preserving agents using a large amount of privacy-preserving expert data [20]. Expanding further, Choi et al. apply imitation learning to develop a generative model for urban vehicle trajectory data, which reproduces synthetic data by mimicking the decision-making process [21]. In a similar vein, Wang et al. employs generative modeling, which combines imitation learning to represent urban vehicle trajectories as a partially observable Markov decision process, to capture their intrinsic distribution [22]. Moreover, Wang et al. propose PateGail, a powerful mobile trajectory generator based on GAIL and federated learning, which is able to extract hidden human decision-making processes to generate rational mobile trajectories while protecting user privacy through differential privacy guarantees [23].

The aforementioned imitation learning methods have made some progress in the field of trajectory generation, demonstrating their potential for learning expert trajectory decision rules and generating high-quality trajectories. However, these methods still have some obvious limitations in practical applications: current methods show certain deficiencies in dealing with the complex dynamic characteristics of trajectories. For trajectories with complex spatio-temporal interactions, many methods are difficult to accurately capture the dynamic dependencies between trajectory points. In addition, the current imitation learning methods mainly pursue the authenticity and practicality of the generated trajectories, but under the high privacy protection requirements, the quality and applicability of the generated trajectories are often significantly affected, and it is difficult to realize an effective balance between privacy protection and data practicality.

2.3 Markov Chain in Dynamic Modeling

Markov models are commonly used in many domains for decision-related tasks, especially in dynamic systems or stochastic environments, such as path planning and navigation, dynamic resource allocation, simulation and generative scenarios. Markov models are able to capture dynamic changes between states and find optimal solutions through reward functions and transfer probability optimization strategies.

Markov models have been extensively applied in trajectory analysis and privacy preservation due to their capability to model dynamic transitions. For example, Zhang et al. use Markov models to predict users' next locations based on their mobility patterns. Building on this, they further design algorithms to achieve spatial K-anonymity using the predicted locations [24]. Meanwhile, Ko et al. develop the Constrained Markov Decision Process (CMDP) problem in their proposed Location Privacy Guaranteed Offloading Algorithm (LPGA), which is transformed into an equivalent Linear Programming (LP) model to achieve the optimal offloading policy [25]. In addition, Wang et al. present a new method for trajectory data synthesis, PrivTrace. The key insight is the use of a new method for choosing between first- and second-order transformation information for the next step of prediction [26]. Furthermore, Chen et al. propose a novel differential privacy framework for protecting symbolic system-generated trajectories,and extend these differential privacy mechanisms to Markov chains to ensure that all privately generated words are feasible with respect to the dynamics of the underlying Markov chain [27].

The above study demonstrates the important role of Markov models in location privacy preservation and trajectory generation. However, there are limitations in these approaches: insufficient modeling capability for complex trajectory dynamic laws, difficulty in balancing privacy preservation and trajectory utility, and privacy risks associated with the reliance of some schemes on centralized data. Therefore, in this paper, we will combine differential privacy with Markov chain models to achieve privacy protection while improving the ability to model the dynamic characteristics and diversity of trajectories in order to optimize the quality and security of trajectory generation.

3 Preliminary

Definition 1 (Trajectory). *In the field of location privacy protection, a trajectory of length n refers to a sequence of location information of an object or subject within a certain time range. A trajectory T can be defined as a time series $T = \{(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)\}$, in which (x_i, y_i) is the spatial coordinate of the ith position point of the trajectory, indicating the position of the user at time point t_i.*

Definition 2 (ϵ-Differential Privacy). *A randomized algorithm $\mathcal{A}$ satisfies ϵ-differential privacy if for any two adjacent datasets D and D′ (where adjacent means that they differ by at most one element), and for all subsets S of $\mathcal{A}$ satisfy Eq. (1):*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D') \in S] \tag{1}$$

where $\Pr[\mathcal{A}(D) \in S]$ represents the probability that the output of the algorithm $\mathcal{A}$ applied to dataset D falls into the set S.

Definition 3 (Global Sensitivity). *The sensitivity of a function, often called global sensitivity, quantifies the maximum amount of change that the function f can produce in its output when any single entry in the dataset is changed. This concept is crucial for applying differential privacy. The global sensitivity of the function f, expressed mathematically, is given by Eq. (2):*

$$\Delta f = \max_{D,D'} |f(D) - f(D')| \tag{2}$$

where D and D′ are any adjacent datasets.

Definition 4 (Laplace Mechanism). *The Laplace Mechanism for a function f, which queries a database, involves adding noise that follows a Laplace distribution to the output of f. The probability density function (PDF) of the Laplace distribution is given by Eq. (3):*

$$\text{Lap}(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right) \tag{3}$$

where x represents the variable, and b is the scale parameter of the Laplace distribution. The scale parameter b is typically set to $\frac{\Delta f}{\epsilon}$, where Δf is the global sensitivity of f, and ϵ is the privacy budget. To implement differential privacy, one adds a noise term to a function f(D), which calculation method is shown by Eq. (4) in a differentially private manner:

$$f_{\text{private}}(D) = f(D) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right) \tag{4}$$

Here, the term Lap $\left(\frac{\Delta f}{\epsilon}\right)$ represents noise drawn from a Laplace distribution centered at zero with a scale parameter $\frac{\Delta f}{\epsilon}$, where Δf is the global sensitivity of the function f, and ϵ is the privacy budget.

We assume the attacker has the following knowledge:

(1) The attacker has access to processed trajectory datasets containing noisy and attempts to exploit these transformations to infer original sensitive trajectory points or user identities.

(2) The attacker analyzes the synthetic trajectories generated by the Imitation Learning framework, attempting to reconstruct spatio-temporal features or infer trajectory patterns by exploiting the reward function optimization and adversarial mechanisms.

(3) The attacker utilizes advanced probabilistic modeling techniques to exploit the Markov Chain's state transfer probability matrix, aiming to infer dynamic trajectory transitions or uncover goal-oriented trajectory patterns that could compromise user privacy.

An attacker with the above background knowledge aims to simulate a Re-identification Attack on the DPIL-Traj system. The task of the DPIL-Traj framework is to effectively protect user privacy by mitigating re-identification risks, ensuring that the attacker cannot accurately reconstruct or infer sensitive trajectory data while maintaining the utility of synthetic trajectories.

The key notations are summarized in Table 1.

Table 1: Key notations

Notation	Definition	Notation	Definition
T_o	Original trajectory	ϵ	Privacy budget
K	Number of clusters	Δf	Sensitivity
μ_i	Cluster centers	x_i	Position point
η_i	Laplace noise	T_o'	Trajectory with noise
α, β	Reward weights	λ	Learning rate
π^*	Optimized policy network	s_t	State of the environment at time t
I	Other status information	a_t	Action at time t
Z_i	Score of action a	R_t	Reward at time step t
s_0	Initial state	P	Transition probability matrix
L	Maximum trajectory length	C_{result}	Clustering result
T_s	Synthetic Trajectory	T_a	Generated agent trajectory

4 Methodology

4.1 System Model

As shown in Fig. 1, the system model of the DPIL-Traj is divided into three main parts: users, trusted third party, and imitation learning generator.

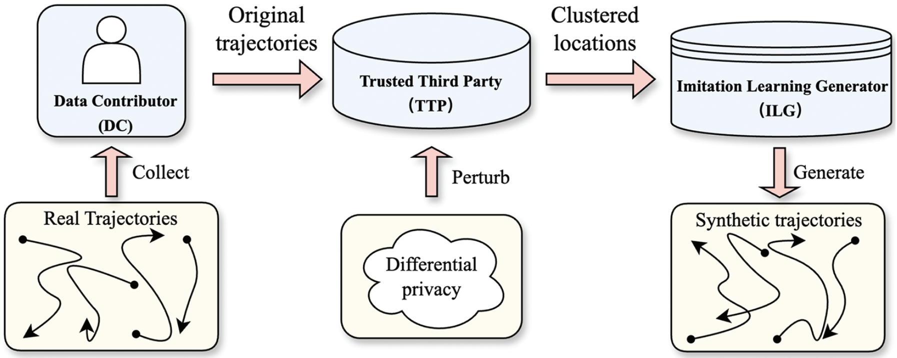


Figure 1: The system model of DPIL-Traj privacy protection method

Data Contributor (DC): DC represents the individuals or systems that upload original trajectory data to the trusted third party for further processing.

Trusted Third Party (TTP): TTP is a trusted external server that performs certain computing tasks. The core algorithm of the trusted third server in this paper is the K-means clustering with differential privacy, and then pass the clustering results to the imitation learning generator.

Imitation Learning Generator (ILG): ILG is the core module of the system for trajectory generation based on clustering data. It uses the clustering results provided by a trusted third-party server to learn the spatial pattern and location distribution of the trajectory, and finally generates a synthetic trajectory based on the learned trajectory pattern.

The DPIL-Traj framework proposed in this paper consists of the following three phases. The pseudocode for the entire algorithm is shown in Algorithm 1, which invokes three core procedures. The K-means clustering with Differential Privacy is described in Procedure 1 of Section 4.2, the Model Training with Imitation Learning is described in Procedure 2 of Section 4.3, and the Trajectory Generation with Markov Chain Model is described in Procedure 3 of Section 4.4.

Algorithm 1: Privacy-preserving trajectory generation framework

```
Input: Original trajectory $T_o$, privacy budget $\varepsilon$, number of clusters $K$, reward weights $\alpha, \beta$, learning rate $\lambda$, transition probability matrix $P$, maximum trajectory length $L$
Output: Generated agent trajectory $T_a$
Perform Procedure 1 (K-means clustering with differential privacy) to obtain noisy cluster centers $T_o'$;
Add Laplace noise to $T_o$, then execute K-means clustering on $T_o$ to compute cluster centers
    $T_o' = \{\mu_1, \mu_2, \ldots, \mu_K\}$;
Perform Procedure 2 (Model Training with Imitation Learning) using $T_o'$;
Train a global policy network by initializing states $s_t = [\mu_t, t, I]$ and optimizing the reward function:
    $R_t = \alpha \cdot \text{cos\_sim}(T_o' - T_a) - \beta \cdot \|(x_{at}, y_{at}) - (\mu_x, \mu_y)\|$
Perform Procedure 3 (Trajectory Generation with Markov Chain Model) to generate $T_a$;
Use the transition probability matrix $P$ and trained global policy network to iteratively sample next states
    $s_{t+1} \sim P(s_{t+1}|s_t)$ and generate actions $a_t$ for trajectory construction.
return $T_a$
```

The DPIL-Traj framework operates in a structured process comprising three interconnected phases, each designed to ensure privacy preservation, effective learning, and high-quality trajectory generation. These modules interact in a sequential and interdependent manner. Firstly, the differential privacy clustering module first processes the original trajectory data to produce privacy-preserving cluster centers. Secondly, these centers are then used as input features for the imitation learning module, which learns a global policy network. Finally, the Markov-based generation module synthesizes trajectories by leveraging the learned policy and environmental feedback. These phases are detailed as follows:

Phase 1 (Differential Privacy Clustering): After receiving the original trajectory T_o uploaded by the DC, the TTP uses differential privacy technology to process it to obtain T_o', and then clusters the location points through the K-means clustering algorithm to obtain a set of privacy-protected cluster centers, which are uploaded to the ILG. The clustering result can be described by the index C_{result} representing each data point and its cluster center, as shown in the module ① in Fig. 2. To achieve this goal, a K-means clustering with Differential Privacy is designed in this paper, which is described in Procedure 1 of Section 4.2.

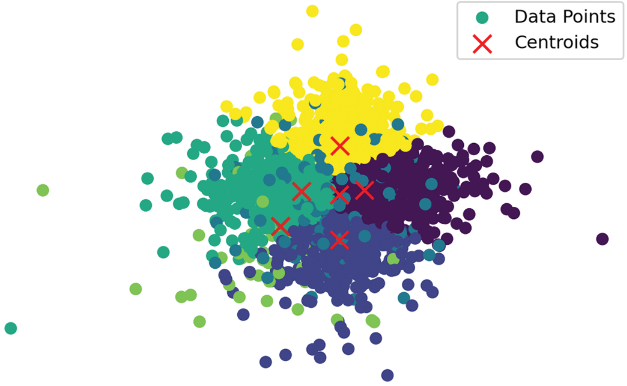


Figure 2: The framework of DPIL-Traj privacy protection method

Phase 2 (Imitation Learning): After receiving the cluster center result C_{result}, the ILG extracts useful features from it as the input of the global policy network (Global Policy Net) and trains it. During the training process, the imitation learning model evaluates the quality of the synthetic trajectory it generates through the reward function, thereby optimizing the network's generation strategy. This phase is shown in the module ②. To achieve this goal, a Model Training with Imitation Learning is designed in this paper, which is described in Procedure 2 of Section 4.3.

Phase 3 (Markov-Based Trajectory Generation): As shown in the module ③, selecting a possible initial state T_s^1, and gradually advancing the state according to the actions predicted by the generative adversarial imitation learning model and the feedback of the environment until the end point or the preset trajectory length is reached, and the synthetic trajectory $T_s = \{T_s^1, T_s^2, \ldots, T_s^n\}$. To achieve this goal, a Trajectory Generation with Markov Chain Model is designed in this paper, which is described in Procedure 3 of Section 4.4.

In the following three sections, the implementation of the three phases of the DPIL-Traj privacy framework is proposed, including three procedures: Differential Clustering, Imitation Learning and Trajectory Generation.

4.2 K-Means Clustering with Differential Privacy

In data clustering analysis, if noise is not added, even if the clustering results themselves do not directly disclose the specific information of the individual, it is still possible to infer certain characteristics of the individual through analysis of the clustering results [28]. Therefore, before performing K-means clustering on the location, noise is added to protect the individual from being identified or inferred, thereby achieving the standard of differential privacy.

Differential Privacy (DP). In order to achieve differential privacy, we must first determine the calculation of global sensitivity. In the system proposed in this paper, clustering is performed based on the Euclidean distance of the location points, so the global sensitivity is defined as the maximum distance between any two points in the data set, and its calculation method is defined in Eq. (5).

$$\Delta f = \max_{x,x' \in D} \|x - x'\| \tag{5}$$

where D represents the entire dataset, x and x' are any two points in the dataset, and $\|x - x'\|$ represents the Euclidean distance between the two points. No matter which two points in the data change, the added noise is sufficient to protect any query results involving these two points from revealing specific data point information. The scale parameter b of the Laplace noise is calculated based on the global sensitivity, and the Laplace noise is added to the data independently, thereby ensuring the user's location privacy.

K-means Clustering. Due to the huge amount of trajectory data, it is inefficient and impractical to learn every location point in every trajectory. Therefore, it is proposed to use the K-means clustering algorithm to split a large amount of trajectory data into smaller clusters. By identifying similar data points and grouping them, the main features and trends in the data can be understood more clearly.

Randomly select K data points as the initial cluster centers. For each location point in the data set, calculate its distance to each cluster center and assign it to the cluster represented by the nearest cluster center. For each cluster, recalculate the center of all points in the cluster (usually the arithmetic mean) and set it as the new cluster center. Repeat the above steps until the change in the cluster center is less than the threshold θ. The procedure for System Initialization is shown in Procedure 1.

After differential clustering, the coordinate points are shown as in Fig. 3. This approach ensures enhanced privacy protection while retaining the essential spatial characteristics required for downstream analysis and trajectory generation.

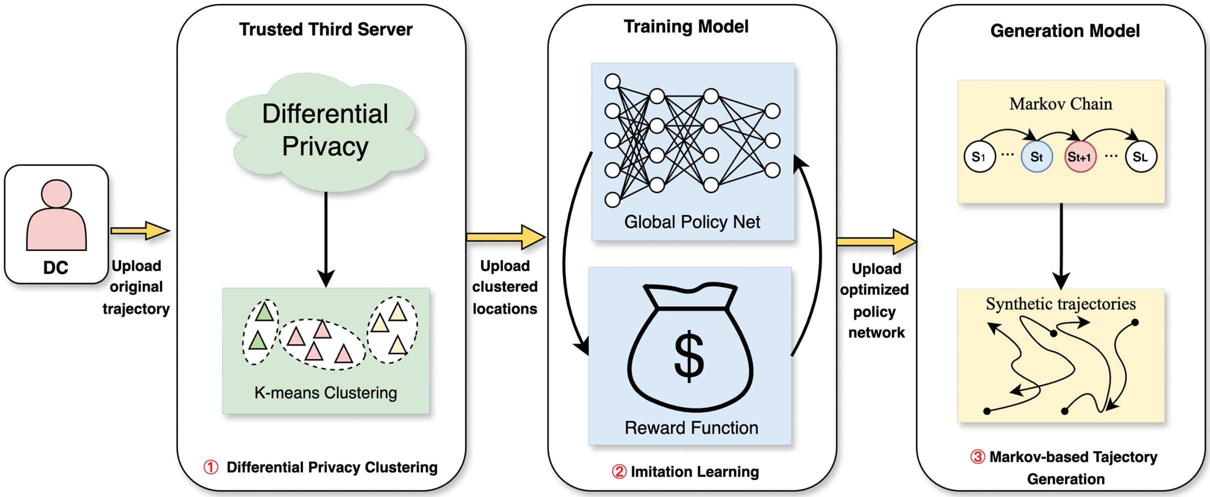


Figure 3: Data clustering results after adding noise

Procedure 1: K-means clustering with differential privacy

```
Input: Original trajectory $T_o$, privacy budget $\varepsilon$, number of clusters $K$, sensitivity $\Delta f$
Output: Cluster centers $\mu_1, \mu_2, \ldots, \mu_K$, clustered data
for each position point $x_i \in T_o$ do
    Compute the sensitivity $\Delta f$ ;
    Sample Laplace noise $\eta_i \sim \text{Laplace}\left(0, \frac{\Delta f}{\epsilon}\right)$
    $x_i' = x_i + \epsilon_i$;
end for
Randomly select $K$ initial cluster centers from the noisy data points $x_i'$;
for each iteration do
    for each noisy data point $x_i'$ do
        Assign $x_i'$ to the closest cluster $C_k$, where
```

(Continued)

Procedure 1 (continued)

```
        $k = \arg\min_k \|x_i' - \mu_k\|$
10:  end for
11:  for each cluster $C_k$ do
12:      Update the cluster center $\mu_k$ as the mean of the points in $C_k$:
             $\mu_k = \frac{1}{|C_k|} \sum_{x_i' \in C_k} x_i'$
13:    end for
14: end for
15: return $T_o' = \{\mu_1, \mu_2, \ldots, \mu_K\}$;
```

4.3 Model Training with Imitation Learning

Imitation Learning (IL) is an approach to learning tasks by observing expert demonstrations, which is used to generate synthetic trajectories and optimize the behavior of the generated trajectories by imitating the learning process. Here, we elaborate on the implementation of Imitation Learning in terms of Global Policy Networks and Reward Functions. IL framework is designed to operate on clustered trajectories derived from DP-K-means, rather than raw inputs. This privacy-aware setting guides the policy to learn behavior patterns from abstracted, noise-protected representations instead of sensitive location data.

Global Policy Networks. The global policy network is a central component in imitation learning, responsible for outputting a probability distribution of actions based on the current state. For the problem of generating trajectories, this network learns how to generate actions that match the trajectory generation goal based on different input positions, speeds, and other environmental features. A global policy network is typically a deep neural network that accepts input information from the environment e.g., position, velocity, historical trajectory, and outputs a probability distribution of actions. The network structure usually includes three components: Input Layer, Hiddden Layer and Output Layer.

(1) Input Layer. Input the state of the environment s_t at time t, including the position coordinates after differential privacy clustering, expressed in terms of the cluster center coordinates after clustering, assuming that the cluster center at time t is μ_t, the coordinates of the cluster center $\mu_t = (x_{\mu_t}, y_{\mu_t})$ can be used as input. Thus s_t of the input layer inputs can be represented as Eq. (6).

$$s_t = [\mu_t, t, I] \tag{6}$$

where I denotes other status information.

(2) Hiddden Layer. Contains a number of fully connected layers and activation functions to extract high-level features of the state.

(3) Output Layer. The output layer is a softmax layer whose goal is to realize a path that conforms to the goal trajectory. By computing the probability distribution of each possible action, it output the action of the next state s_{t+1} based on the current state s_t as Eq. (7).

$$P(a_t = a_i \mid S_t) = \frac{\exp(Z_i)}{\sum_{j=1}^{n} \exp(Z_i)} \tag{7}$$

where Z_i is the score of action a, n is the size of the action space, indicating the number of all possible actions. $P(a_t = a_i \mid s_t)$ is the probability of choosing an action a_i in the current state s_t.

To ensure semantic fidelity and spatial alignment under differential privacy, design a reward function that combines global cosine similarity to preserve trajectory shape and localized Euclidean distance to anchor positions to privacy-protected cluster centers.

Reward Functions. In imitation learning, it is important to make the agent trajectories as close as possible to the expert trajectories. Shape similarity and position distance can well represent the similarity between the agent trajectories and the expert trajectory. Assuming the agent trajectory is $T_a = a_1, a_2, \ldots, a_t$, and the expert trajectory is after differential cluetered $T_o' = \mu_1, \mu_2, \ldots, \mu_K$. Therefore, the reward function at time t can be designed as shown in Eq. (8)

$$R_t = \alpha \cdot cos\_sim(T_o' - T_a) - \beta \cdot \|(x_{at}, y_{at}) - (x_{\mu_t}, y_{\mu_t})\| \tag{8}$$

where $cos\_sim(T_o' - T_a) = \frac{\sum_{t=1}^{n}(a_t \cdot \mu_t)}{\|T_o'\|\|T_a\|}$, which measures the cosine similarity between the agent trajectory and the expert trajectory to compare the overall shape of the trajectories. $-\|\cdot\|$ denotes that the distance between trajectory points is calculated using Euclidean distance, and a negative sign indicates that the smaller the distance, the higher the reward. α and β were used to control for the effects of the two rewards on overall learning. The procedure for Imitation Learning is shown in Procedure 2.

Procedure 2: Model training with imitation learning

```
Input: Original trajectory $T_o$, cluster centers $\mu_t$, trajectory with noise $T_o'$, reward weights $\alpha, \beta$, learning rate $\lambda$
Output: Optimized policy network $\pi^*$
Initialize global policy network with input, hidden, and output layers
for each training episode do
    Initialize state $s_t = [\mu_t, t, I]$ using cluster centers, time, and additional information
    for each time step $t$ do
        Compute action probabilities $P(a_t|s_t)$ from the global policy network
        Select action $a_t$ based on $P(a_t|s_t)$
        Update state $s_{t+1}$ based on $a_t$
        Append $s_{t+1}$ to agent trajectory $T_a$
    end for
    Compute reward $R_t = \alpha \cdot \text{cos\_sim}(T_o' - T_a) - \beta \cdot ||(x_{at}, y_{at}) - (\mu_x, \mu_y)||$
    Optimize global policy network using $R_t$
end for
return $\pi^*$
```

4.4 Trajectory Generation with Markov Chain Model

Combined with the action selection mechanism of the strategy network, the use of Markov chain model to generate synthetic trajectories can make the trajectory generation process both stochastic and goal-oriented, and closer to the distribution of the real trajectories. Instead of simply selecting a static cluster center, the model dynamically transitions between states based on learned probabilities, enabling the generation of entire trajectories rather than discrete points. Therefore, in this paper the dynamic transfer of the state of the intelligent body is modeled by a Markov chain model, and the selection of the next state is performed based on the state transfer probability matrix, thus generating a synthetic trajectory.

Importantly, the Markov model operates on clustered states generated by the differentially private K-means algorithm, rather than on raw location points, ensuring that no additional privacy leakage is introduced in the temporal modeling process.

Let the set of states of the trajectory be $S = \{s_1, s_2, \ldots, s_n\}$, each state contains position coordinates and a timestamp, the transfer probability from state s_t to state s_{t+1} is defined by matrix P, as shown in Eq. (9):

$$P = \begin{bmatrix} P(s_1|s_1) & P(S_2|s_1) & \cdots & P(s_n|s_1) \\ P(s_1|s_2) & P(S_2|s_2) & \cdots & P(s_n|s_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(s_1|s_n) & P(s_2|s_n) & \cdots & P(s_n|s_n) \end{bmatrix} \tag{9}$$

where $P(s_j|s_i)$ denotes the probability that state S_i transfers to state S_j in the next step, $\sum_j P(s_j|s_i) = 1$, i.e., the sum of all state transfer probabilities is 1.

Set the initial state as s_0, initialize the null trajectory T_a, and transfer from the current state s_t to the next state s_{t+1} according to the transfer probability matrix P of Markov chain. Combined with the optimized policy network, calculate the action probability $P(a_t|s_t)$ in the current state s_t, select the action a_t, and update the state to s_{t+1}. The process is repeated until the trajectory length reaches L. The pseudo-code is shown in Procedure 3.

Procedure 3: Trajectory generation with Markov chain model

```
Input: Optimized policy network $\pi^*$, transition probability matrix $P$, maximum trajectory length $L$
Output: Generated agent trajectory $T_a$
Initialize state $s_t = s_0$
for each time step $t = 1$ to $L$ do
    Sample next state $s_{t+1} \sim P(s_{t+1}|s_t)$
    Compute action probabilities $P(a_t|s_t)$ from $\pi^*$
    Select action $a_t$ based on $P(a_t|s_t)$
    Update state $s_{t+1} = \text{update\_state}(s_t, a_t)$
    Append state $s_{t+1}$ to trajectory $T_a$
    Set $s_t = s_{t+1}$
end for
return $T_a$
```

5 Performance Analysis

In this section, the performance of DPIL-Traj is assessed across multiple aspects, including its ability to simulate real trajectories and preserve privacy. Additionally, the comparisons are made with other state-of-the-art algorithms.

5.1 Experiment Setup

In this paper, the GPS-based GeoLife real trajectory dataset is used for experiments. The dataset contains 17,621 trajectories of 181 users, and each point contains timestamp, latitude, longitude and altitude information [29]. In order to have a reliable evaluation of our proposed DPIL-Traj, we select the following trajectory generation algorithms to be compared with:

(1) **TrajGail** [21] uses a generative adversarial framework to learn the latent distribution of urban vehicle trajectory data.

(2) **PateGail** [23] utilizes a powerful generative adversary imitation learning model to simulate human mobility, training the model collectively based on mobility data, while personal identifiers are trained locally.

(3) **GeoPM-DMEIRL** [17] employs a segmentation mechanism to protect user trajectories and combines reinforcement learning and Markov chain modeling to produce synthetic trajectories.

(4) **SPSD** [30] generates privacy-preserving trajectories by selecting semantic-aware dummy locations based on visiting time, geographic distance, and motion direction, ensuring k-anonymity and high similarity to real trajectories.

To verify the effectiveness of the three integrated components, ablation experiments were conducted by separately removing each module and evaluating its impact.

(1) **DPIL-Traj-w/o-IL** denotes the variant where the imitation learning module is removed. In this setting, the model generates trajectories solely based on the cluster centroids obtained from the differential privacy clustering module and the Markov chain model, treating these centroids as compressed abstractions of the original trajectory characteristics.

(2) **DPIL-Traj-w/o-DP** refers to the variant where the differential privacy clustering module is removed. In this version, the model takes the original trajectory T_o as input directly, without applying any differential privacy mechanisms or K-means clustering.

(3) **DPIL-Traj-w/o-M** refers to the variant where the Markov-based trajectory generation module is removed. In this case, the model relies solely on the differential privacy clustering and imitation learning modules, generating trajectories by directly producing a sequence of actions through the policy network.

All experiments are conducted on macOS Sequoia v15.2 with a hardware configuration consisting of a 12-core M2 Max CPU and 32 GB of RAM. Furthermore, the computational workload is supported by a dedicated server equipped with NVIDIA RTX 3090 GPU to enhance model training efficiency and data processing capabilities. DPIL-Traj repeated the experiments 10 times for each dataset and averaged the results. The main parameters used in experiments are summarized in Table 2.

Table 2: Parameters setting

Parameter	Setting
Privacy budget ϵ	0.1–2.0
Learning rate λ	0.001
Batch size	64
Number of clusters K	6
Global sensitivity Δf	1
Users in GeoLife	180

5.2 Evaluation Metrics

To comprehensively evaluate the performance of the DPIL-Traj framework, we consider a variety of evaluation metrics that assess different aspects of trajectory generation. These metrics are designed to measure the accuracy, consistency, and realism of the generated trajectories, as well as the model's ability to preserve privacy [23]. Below, we detail the specific metrics used in our evaluation and their corresponding definitions.

(1) **Distance** measures the distance between the generated trajectory and the real trajectory, and relatively small values indicate that the generated trajectory is closer to the real trajectory, reflecting the realism of the model-generated trajectory.

(2) **Radius** measures the extent of the spatial distribution of trajectory points, with smaller values indicating that the trajectory points are more concentrated and more consistent with the actual trajectory pattern.

(3) **Daily-loc** indicates the number of generated trajectory locations per day, which correlates with the consistency of actual movement behavior. Lower values may indicate that the generated trajectories are more consistent with actual daily behavior.

(4) **Duration** measures the error in the duration of actions in the generated trajectory, with smaller values indicating the model's ability to learn temporal patterns.

(5) **I-rank** measures the deviation of the ranking of important locations in the trajectory from the true trajectory, with smaller values indicating that the generated trajectory is more consistent with the true trajectory in capturing key locations.

(6) **G-rank** measures the consistency of the global ordering of the generated trajectories with the real trajectories, with smaller values indicating a higher quality of the overall trajectory generation.

To further validate the privacy of the proposed framework, the four algorithms are evaluated for security using the following metrics:

MAE (Mean Absolute Error) [31]. A metric used to assess the difference between predicted and true values, measuring the accuracy of a model by calculating the average of the absolute values of all errors. Its mathematical definition as shown in Eq. (10):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{10}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value and n is the sample size. It evaluates the similarity between the generated trajectories and the real trajectories.

HD (Hausdorff Distance) [32]. A geometric metric used to compare the similarity of two point sets. It defines the maximum and minimum distance from a point in one set to the nearest point in another set. Its mathematical definition as shown in Eq. (11):

$$d_H(A, B) = \max\left\{\sup_{a\in A}\inf_{b\in B}\|a - b\|, \sup_{b\in B}\inf_{a\in A}\|b - a\|\right\} \tag{11}$$

where A and B are two point sets, $\|a - b\|$ denotes the Euclidean distance between two points. It measures the shape similarity between the generated trajectory and the real trajectory.

MI (Mutual Information) [33]. Mutual information is a measure of how much information is shared between two random variables. It reflects the extent to which the value of one variable reduces uncertainty about the other. Its mathematical definition as shown in Eq. (12):

$$I(X; Y) = \sum_{x\in X}\sum_{y\in Y} p(x, y)\log\frac{p(x, y)}{p(x)p(y)} \tag{12}$$

where $p(x, y)$ is the joint probability distribution, $p(x)$ and $p(y)$ are marginal probability distributions. It can be used to assess the correlation between generated trajectories and real trajectories in terms of spatial distribution characteristics.

5.3 Ablative Experiments

The contribution of each module was evaluated through ablation experiments on the DPIL-Traj. The results of the experiment are given by the Table 3. Specifically, it achieves a distance of 0.0457 and a radius of 0.0360, indicating that the combined model generates point distributions and spatial extents most similar to real trajectories. Furthermore, it significantly outperforms all variants in terms of daily-loc, with a value of 0.0517, and duration, with a value of 0.0072, suggesting that integrating all three modules enables better simulation of daily mobility patterns and temporal dynamics. In addition, DPIL-Traj achieves the lowest i-rank (0.0081) and g-rank (0.0076), indicating better consistency and global realism. These results further highlight the full model's advantage in preserving spatial order and mobility patterns.

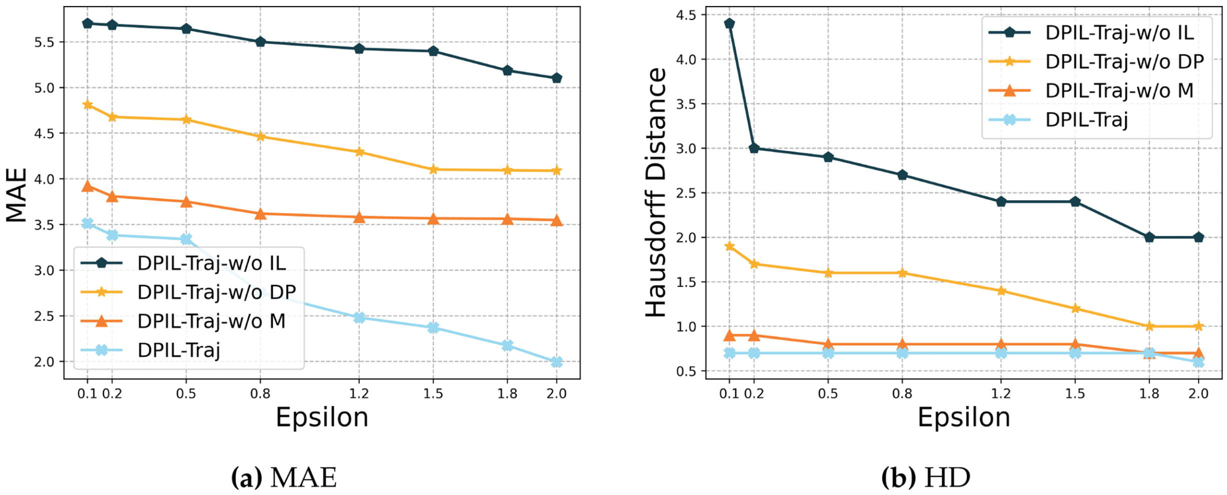
Table 3: Comparison of DPIL-Traj variants

Method	**Distance**	**Radius**	**Daily-loc**	**Duration**	**i-rank**	**g-rank**
DPIL-Traj-w/o IL	0.0474	0.0375	0.0693	0.0111	0.0177	0.0139
DPIL-Traj-w/o DP	0.0462	0.0371	0.0660	0.0106	0.0176	0.0139
DPIL-Traj-w/o M	0.0460	0.0370	0.0589	0.0098	0.0110	0.0086
DPIL-Traj	0.0457	0.0360	0.0517	0.0072	0.0081	0.0076
Improvement	0.65%	2.70%	12.22%	26.53%	26.36%	11.63%

Overall, these results confirm that the full DPIL-Traj model—integrating Differential Privacy Clustering (DP), Imitation Learning (IL), and Markov-based Trajectory Generation (M)—achieves the best trade-off between accuracy, behavioral realism, and privacy preservation.

Fig. 4 illustrates the performance of different model variants on MAE and HD. DPIL-Traj consistently achieves the lowest MAE, as it effectively captures the decision-making rules of real trajectories through imitation learning, while also incorporating differential privacy clustering and Markov chain modeling, so that the generated trajectories can be closer to the real data in terms of spatial and dynamic characteristics. Likewise, it performs best on HD, benefiting from the Markov model's ability to represent dynamic transition patterns and the refinement of trajectory shapes through imitation learning. These findings demonstrate that the combination of the three modules significantly improves the accuracy and shape similarity of the generated trajectories.

We further evaluated the privacy protection capability of each model variant using the MI metric. As shown in Fig. 5, the MI value increases as the privacy budget ϵ becomes larger, indicating that a higher privacy budget allows the model to access more information from the original trajectories, thereby increasing the correlation between generated and real data. Among the variants, DPIL-Traj-w/o-DP shows the highest MI, confirming that the removal of the differential privacy module leads to a notable rise in privacy leakage risk. Although DPIL-Traj-w/o-IL and DPIL-Traj-w/o-M also yield higher MI values than the full model, they remain lower than the variant without DP, suggesting that both imitation learning and Markov modeling contribute to privacy protection to some extent.



(a) MAE **(b)** HD

Figure 4: Ablation study: MAE, HD in varying privacy budget ϵ

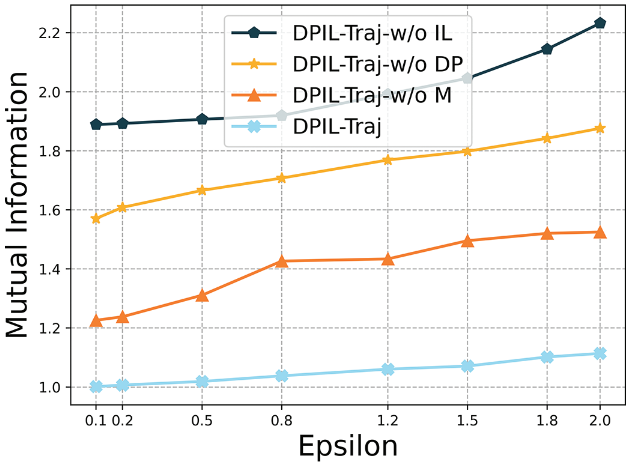


Figure 5: Ablation study: MI in varying privacy budget ϵ

Overall, DPIL-Traj achieves the lowest MI, demonstrating its strong privacy-preserving ability, primarily due to the incorporation of the differential privacy mechanism that effectively reduces statistical dependence on real trajectories.

5.4 Comparison with Existing Work

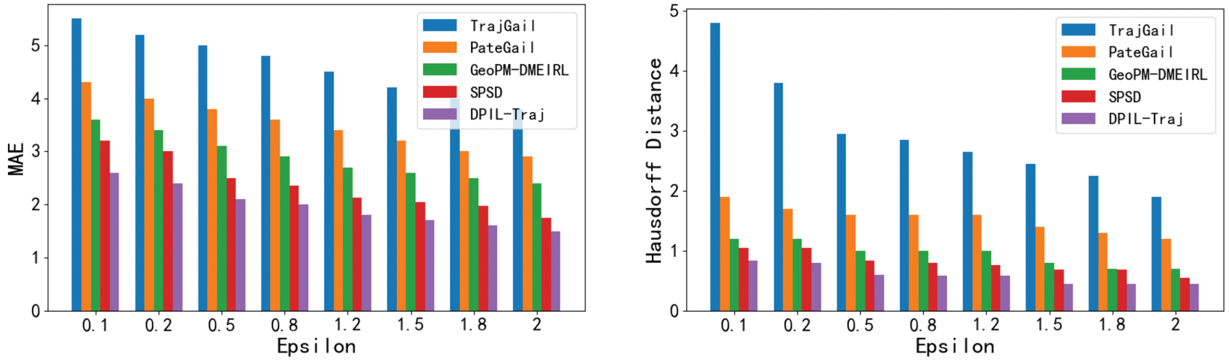
Firstly, we analyze the utility of our framework, Table 4 shows the performance of the DPIL-Traj method compared to the other four trajectory generation models (TrajGail, PateGail, GeoPM-DMEIRL, SPSD) on the Geolife dataset. TrajGail performs the worst on all the metrics, especially on duration (0.0275) and daily-loc (0.2938), which are much higher than DPIL-Traj. Analysis reveals that it lacks a privacy protection mechanism and is not able to effectively capture time-dynamic features and daily behavioral patterns. In contrast, PateGail combines GAIL and privacy-preserving mechanisms to simulate the user's mobile behavior in a distributed manner, and thus it improves on all metrics, especially on radius (0.0399) and g-rank (0.0285). However, it still fails to adequately capture temporal features, resulting in duration (0.0141) and daily-loc (0.1695) being significantly higher than DPIL-Traj. In addition, the cost of privacy protection may affect the fidelity of trajectory generation.GeoPM-DMEIRL generates trajectories through trajectory segmentation and Markov chain modeling while incorporating a reinforcement learning optimization strategy. It thus performs better on duration (0.0131) and irank (0.0194) compared to TrajGail and PateGail, showing the ability to model temporal dynamics and critical locations. However, the spatial distribution properties are weaker, with radius (0.0385) and g-rank (0.0277) not performing as well as DPIL-Traj. SPSD introduces semantic- and direction-aware dummy generation with a k-anonymous guarantee to protect trajectory privacy. It outperforms TrajGail and PateGail in spatial metrics such as distance (0.0474) and radius (0.0371). However, it struggles to model temporal behaviors, with daily-loc (0.0954) and duration (0.0104) still notably higher than DPIL-Traj. This indicates that despite enhanced spatial plausibility, SPSD lacks fine-grained temporal adaptation, leading to reduced behavioral fidelity compared to DPIL-Traj.

Table 4: Comparison of different trajectory generation models on geolife dataset

Method	Distance	Radius	Daily-loc	Duration	i-rank	g-rank
TrajGail	0.0515	0.0409	0.2938	0.0275	0.0503	0.0451
PateGail	0.0497	0.0399	0.1695	0.0141	0.0295	0.0285
GeoPM-DMEIRL	0.0487	0.0385	0.1213	0.0131	0.0194	0.0277
SPSD	0.0474	0.0371	0.0954	0.0104	0.0112	0.0098
DPIL-Traj	0.0457	0.0360	0.0517	0.0072	0.0081	0.0076
Improvement (TrajGail)	11.26%	11.98%	82.40%	73.82%	83.89%	83.14%
Improvement (PateGail)	8.04%	9.77%	69.50%	48.94%	72.54%	73.33%
Improvement (GeoPM-DMEIRL)	6.16%	6.49%	57.36%	45.04%	58.25%	72.57%
Improvement (SPSD)	3.58%	2.96%	45.80%	30.76%	27.67%	22.44%

Our DPIL-Traj approach combines a combination of differential privacy clustering, imitation learning and Markov trajectory generation. It shows significant improvement over TrajGail on daily-loc and duration, indicating that its mechanism of combining differential clustering, imitation learning and Markov modeling is able to better reproduce daily behavioral patterns. It outperforms PateGail on distance (8.04%) and radius (9.77%), indicating that its privacy-preserving mechanism has less impact on trajectory realism. It outperforms GeoPM-DMEIRL on i-rank (58.25%) and g-rank (72.57%), indicating that it generates trajectories with better global ordering consistency and key location identification. DPIL-Traj achieves an all-round optimization in terms of trajectory realism, spatial distribution and temporal dynamics, which is significantly better than the other three models. Compared to SPSD, DPIL-Traj achieves 45.80% improvement in daily-loc, 30.76% in duration, 27.67% in i-rank, and 22.44% in g-rank. This indicates that while SPSD leverages semantic-aware dummy strategies for privacy, it lacks the fine-grained temporal modeling and dynamic behavior learning embedded in DPIL-Traj.

Then we compared the performance of the four algorithms in terms of MAE and HD to further explore utility performance. From Fig. 6a, it can be seen that as the privacy budget ϵ increases, the MAE values of each model gradually decrease. This indicates that the error between the generated trajectory and the real trajectory is significantly reduced and the utility of the model is improved under higher privacy budgets. Under all privacy budgets, DPIL-Traj consistently maintains the lowest MAE values, especially in the case of low privacy budgets (e.g., $\epsilon = 0.1$), its performance advantage is especially prominent. This indicates that DPIL-Traj can more accurately generate trajectories that are close to the real trajectories. Specifically, our proposed method improves 55.23% over PateGail, 43.15% over TrajGail, 28.09% over GeoPM-DMEIRL, and 18.75% over SPSD in terms of MAE. GeoPM-DMEIRL, despite incorporating trajectory segmentation and reinforcement learning for policy optimization, relies on Markov chain-based modeling, which limits its capacity to capture long-range temporal dependencies and complex user mobility patterns. In contrast, DPIL-Traj leverages attention-based temporal modeling and personalized user priors, enabling it to generate trajectories that align more closely with real-world behaviors. Fig. 6b shows the performance of the different models on the HD metrics. DPIL-Traj shows a clear advantage on HD, especially at low privacy budgets (e.g., $\epsilon = 0.1$ and $\epsilon = 0.5$), and the maximum deviation between the generated trajectory and the real trajectory point set is significantly smaller than that of the comparison models. This indicates that DPIL-Traj is able to better capture the overall shape and features of the real trajectories, ensuring high spatial accuracy and consistency of the generated trajectories. Specifically, the DPIL-Traj proposed in this paper improves the performance of HD by 75.35% compared to PateGail, 57.61% compared to TrajGail, 28.48% compared to GeoPM-DMEIRL, and 20.95% compared to SPSD. This indicates that although SPSD improves utility through semantic-aware dummies, it still falls short of the fine-grained temporal modeling and robust privacy-utility balance achieved by DPIL-Traj.



(a) MAE

(b) HD

Figure 6: MAE, HD in varying privacy budget ϵ

In summary, the DPIL-Traj model proposed in this paper significantly outperforms existing methods in both MAE and HD metrics, indicating that the generated trajectories are not only closer to the real trajectories in terms of positional accuracy, but also have a higher degree of reducibility in terms of overall shape and spatial consistency. This superiority reflects the good performance in utility of our method.

In order to validate the privacy of the proposed methods, we use MI to evaluate the privacy preserving effect of all the algorithms. As shown in Fig. 7, it can be seen that over all privacy budgets ϵ, the MI values of DPIL-Traj are always significantly lower than those of the other baseline models. This suggests that DPIL-Traj can effectively reduce the statistical correlation between generated trajectories and real trajectories, thus better protecting users' privacy. As the privacy budget increases, the MI value of each model increases, but DPIL-Traj has the smallest increase and always maintains a relatively low MI value. At lower privacy budgets, the MI values of the models vary widely. DPIL-traj has the lowest MI value, indicating that under stronger privacy protection, DPIL-Traj leaks the least amount of raw trajectory information. For example, DPIL-Traj shows a 44.41% improvement in MI compared to PateGail, showing a significant reduction in the statistical correlation between its generated trajectories and the real trajectories. At moderate or high privacy budgets, the MI increased for all models, but DPIL-Traj still maintained the lowest MI. This suggests that although the increase in privacy budget allows the model to utilize more information to generate trajectories, DPIL-Traj still manages to avoid overexposing the original trajectory information while increasing its utility. The performance of DPIL-Traj on MI metrics validates the effectiveness of its privacy preservation. By introducing differential privacy mechanism and imitation learning, DPIL-Traj not only significantly reduces the statistical dependence of the generated trajectories on the original trajectories, but also maintains a stable privacy-preserving performance under multiple privacy budgets.

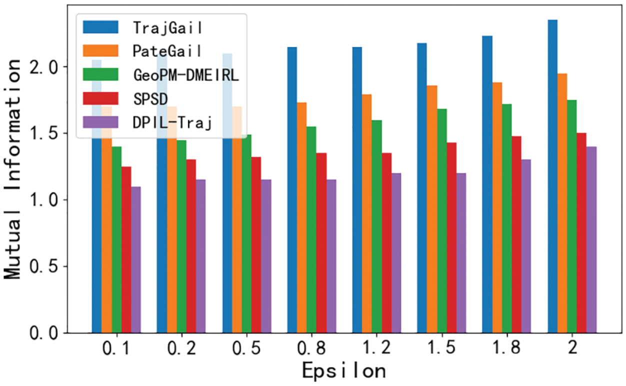


Figure 7: MI in varying privacy budget ϵ

Similarly, DPIL-Traj achieves an 11.24% improvement in MI compared to SPSD, indicating a further reduction in the statistical dependence between generated and real trajectories. Although SPSD introduces semantic-aware dummy trajectories to enhance privacy, it still retains a notable degree of correlation with the original data. In contrast, DPIL-Traj, by integrating differential privacy mechanisms with imitation learning, is able to more effectively obscure sensitive trajectory patterns. This allows DPIL-Traj to maintain stronger privacy guarantees, especially under strict privacy budgets, while still preserving the overall utility of the generated data.

6 Conclusion

In this work, we proposed DPIL-Traj, a privacy-preserving trajectory generation framework that integrates differential clustering, imitation learning, and Markov chain modeling to achieve a balance between utility and privacy. Comprehensive experiments conducted on benchmark datasets demonstrate the superiority of DPIL-Traj. Compared to the state-of-the-art methods, our system achieves the lowest MAE and HD with an average value improvement of 19.85%, indicating improved accuracy and shape similarity of the generated trajectories. In addition, the MI results were improved by 12.51%, validating the strong privacy protection capability of our system, which greatly reduces the risk of privacy leakage.

By introducing differential privacy in the clustering process, our framework effectively protects user location data while maintaining high utility of the data. The imitation learning module captures complex decision patterns to generate realistic synthetic trajectories, while the Markov chain ensures that the generated trajectories maintain dynamic spatio-temporal characteristics. In conclusion, DPIL-Traj provides an effective and efficient solution for privacy-preserving trajectory generation, offering a promising approach for applications requiring high-quality synthetic data. Future work will explore incorporating external factors, such as environmental contexts or social influences, to further improve the adaptability and utility of the generated trajectories.

Acknowledgement: Not applicable.

Funding Statement: This study was supported by the Natural Science Foundation of Fujian Province of China (2025J01380); National Natural Science Foundation of China (No. 62471139); the Major Health Research Project of Fujian Province (2021ZD01001); Fujian Provincial Units Special Funds for Education and Research (2022639); Fujian University of Technology Research Start-up Fund (GY-S24002); Fujian Research and Training Grants for Young and Middle-aged Leaders in Healthcare (GY-H-24179).

Author Contributions: Huaxiong Liao: Conceptualization, Supervision, Methodology. Xiangxuan Zhong: Writing—original draft, Methodology. Xueqi Chen: Investigation, Validation. Yirui Huang: Supervision, Validation. Yuwei Lin: Software. Jing Zhang: Writing—review and editing, Validation. Bruce Gu: Supervision, Software. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data for this research can be made available upon request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Min M, Zhu H, Li S, Zhang H, Xiao L, Pan M, et al. Semantic adaptive geo-indistinguishability for location privacy protection in mobile networks. IEEE Trans Vehicular Technol. 2024;73(6):9193–8. doi:10.1109/tvt.2024.3354881.
2. Kumar GS, Premalatha K, Maheshwari GU, Kanna PR, Vijaya G, Nivaashini M. Differential privacy scheme using Laplace mechanism and statistical method computation in deep neural network for privacy preservation. Eng Appl Artif Intell. 2024;128(2):107399. doi:10.1016/j.engappai.2023.107399.
3. Cai K, Zhang J, Hong Z, Shand W, Wang G, Zhang D, et al. Where Have You Been? A study of privacy risk for point-of-interest recommendation. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2024 Aug 25–29; Barcelona, Spain. p. 175–86.
4. Kumar GS, Premalatha K, Maheshwari GU, Kanna PR. No more privacy Concern: a privacy-chain based homomorphic encryption scheme and statistical method for privacy preservation of user's private and sensitive data. Expert Syst Appl. 2023;234(1):121071. doi:10.1016/j.eswa.2023.121071.
5. Wang H, Xu Z, Jia S. Cluster-indistinguishability: a practical differential privacy mechanism for trajectory clustering. Intell Data Anal. 2017;21(6):1305–26. doi:10.3233/ida-163098.
6. Rao J, Gao S, Kang Y, Huang Q. LSTM-TrajGAN: a deep learning approach to trajectory privacy protection. arXiv:2006.10521. 2020.
7. Hao R, Hussain R, Parra-Ullauri JM, Vasilakos X, Nejabati R, Simeonidou D. GAN-based privacy abuse attack on federated learning in IoT networks. In: IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS); 2024 May 20–23; Vancouver, BC, Canada. p. 1–2.
8. Long Q, Wang H, Li T, Huang L, Wang K, Wu Q, et al. Practical synthetic human trajectories generation based on variational point processes. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2023 Aug 6–10; Long Beach, CA, USA. p. 4561–71.
9. Benaglia R, Porrello A, Buzzega P, Calderara S, Cucchiara R. Trajectory forecasting through low-rank adaptation of discrete latent codes. arXiv:2405.20743. 2024.
10. Ma X, Ding Z, Zhang X. ST-TrajGAN: a synthetic trajectory generation algorithm for privacy preservation. Future Gener Comput Syst. 2024;161(3):226–38. doi:10.1016/j.future.2024.07.011.
11. Zhang J, Huang Q, Huang Y, Ding Q, Tsai PW. DP-TrajGAN: a privacy-aware trajectory generation model with differential privacy. Future Gener Comput Syst. 2023;142(2):25–40. doi:10.1016/j.future.2022.12.027.
12. Xu B, Wang X, Li S, Li J, Liu C. Social-cvae: pedestrian trajectory prediction using conditional variational auto-encoder. In: International Conference on Neural Information Processing. Cham, Switzerland: Springer; 2023. p. 476–89.

13. Yin C, Xi J, Sun R, Wang J. Location privacy protection based on differential privacy strategy for big data in industrial internet of things. IEEE Trans Ind Inform. 2018;14(8):3628–36. doi:10.1109/tii.2017.2773646.
14. Zhang J, Huang Y-R, Huang Q-H, Li Y-Z, Ye X-C. Hasse sensitivity level: a sensitivity-aware trajectory privacy-enhanced framework with Reinforcement Learning. Future Gener Comput Syst. 2023;142(1–2):301–13. doi:10.1016/j.future.2023.01.008.
15. Yang M, Huang L, Tang C. K-means clustering with local distance privacy. Big Data Min Analyt. 2023;6(4):433–42. doi:10.26599/bdma.2022.9020050.
16. Yuan L, Zhang S, Zhu G, Alinani K. Privacy-preserving mechanism for mixed data clustering with local differential privacy. Concurr Comput. 2023;35(19):e6503. doi:10.1002/cpe.6503.
17. Huang Y-R, Zhang J, Hou H-M, Ye X-C, Chen Y. GeoPM-DMEIRL: a deep inverse reinforcement learning security trajectory generation framework with serverless computing. Future Gener Comput Syst. 2024;154:123–39. doi:10.1016/j.future.2024.01.001.
18. Sun X, Ye Q, Hu H, Wang Y, Huang K, Wo T, et al. Synthesizing realistic trajectory data with differential privacy. IEEE Trans Intell Transp Syst. 2023;24(5):5502–15. doi:10.1109/tits.2023.3241290.
19. Wang H, Wang H. Differentially private publication for correlated non-numerical data. Comput J. 2022;65(7):1726–39.
20. Huang C, Chen S, Zhang Y, Zhou W, Rodrigues JJ, de Albuquerque VHC. A robust approach for privacy data protection: ioT security assurance using generative adversarial imitation learning. IEEE Internet Things J. 2021;9(18):17089–97. doi:10.1109/jiot.2021.3128531.
21. Choi S, Kim J, Yeo H. TrajGAIL: generating urban vehicle trajectories using generative adversarial imitation learning. Transport Res Part C Emerg Technol. 2021;128:103091. doi:10.1016/j.trc.2021.103091.
22. Wang M, Cui J, Wong YW, Chang Y, Wu L, Jin J. Urban vehicle trajectory generation based on generative adversarial imitation learning. IEEE Trans Vehicular Technol. 2024;73(12):18237–49. doi:10.1109/tvt.2024.3437412.
23. Wang H, Gao C, Wu Y, Jin D, Yao L, Li Y. PateGail: a privacy-preserving mobility trajectory generator with imitation learning. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence; 2023 Feb 7–14; Washington, DC, USA. p. 14539–47.
24. Zhang S, Li X, Tan Z, Peng T, Wang G. A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services. Future Gener Comput Syst. 2019;94(5):40–50. doi:10.1016/j.future.2018.10.053.
25. Ko H, Lee H, Kim T, Pack S. LPGA: location privacy-guaranteed offloading algorithm in cache-enabled edge clouds. IEEE Trans Cloud Comput. 2020;10(4):2729–38. doi:10.1109/tcc.2020.3030817.
26. Wang H, Zhang Z, Wang T, He S, Backes M, Chen J, et al. {PrivTrace}: differentially private trajectory synthesis by adaptive markov models. In: 32nd USENIX Security Symposium (USENIX Security 23); 2023 Aug 9–11; Anaheim, CA, USA. p. 1649–66.
27. Chen B, Leahy K, Jones A, Hale M. Differential privacy for symbolic systems with application to Markov Chains. Automatica. 2023;152(10):110908. doi:10.1016/j.automatica.2023.110908.
28. Yang M, Tjuawinata I, Lam KY. K-means clustering with local d_X-privacy for privacy-preserving data analysis. IEEE Trans Inf Forensics Secur. 2022;17:2524–37. doi:10.1109/tifs.2022.3189532.
29. Zheng Y, Zhang L, Xie X, Ma WY. Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web; 2009 Apr 20–24; Madrid, Spain. p. 791–800.
30. Huang H, Sun H, Wu W, Wang C, Liu W, Miao W, et al. Synthetic privacy-preserving trajectories with semantic-aware dummies for location-based services. IEEE Trans Serv Comput. 2025;18(3):1811–24. doi:10.1109/tsc.2025.3556642.
31. Zhu Y, Ye Y, Wu Y, Zhao X, Yu J. Synmob: creating high-fidelity synthetic GPS trajectory dataset for urban mobility analysis. Adv Neural Inform Process Syst. 2023;36:22961–77.
32. Liu Q, Yu J, Han J, Yao X. Differentially private and utility-aware publication of trajectory data. Expert Syst Appl. 2021;180(7):115120. doi:10.1016/j.eswa.2021.115120.
33. Li Q, Gundersen JS, Heusdens R, Christensen MG. Privacy-preserving distributed processing: metrics, bounds and algorithms. IEEE Trans Inf Forensics Secur. 2021;16:2090–103. doi:10.1109/tifs.2021.3050064.