[[1]](#footnote-1)

ST-TrajGAN：A Synthetic Trajectory

Generation Algorithm for Privacy Preservation

Zinan Ding and Xuebin Ma

***Abstract*—The rapid development of location-based services (LBS) exposes large-scale trajectory data to the risk of privacy leakage. In order to enhance the trajectory privacy protection while improving the trajectory utility, this paper proposes an efficient and secure deep learning model ST-TrajGAN (Semantic and Transformer-based Trajectory Generative Adversarial Networks) for trajectory data generation and publication. First, a semantic trajectory encoding model is introduced to preprocess the trajectory points. Second, a synthetic trajectory with more uncertainty and practicality is generated by learning the spatial-temporal and semantic features of real trajectory data. In addition, a new TrajLoss loss metric function is designed to measure the trajectory similarity loss of the trained deep learning model. Finally, the effectiveness of the generated synthetic trajectories and the utility of the model is evaluated by TUL and TSP values on the real LBS datasets.**

***Index Terms*—Trajectory privacy; Generative Adversarial Networks; Location-based Services**

# I. INTRODUCTION

T

rajectory data (e.g., vehicle or pedestrian GPS trajectories) play an increasingly important role in intelligent transportation, urban planning, etc., however, direct distribution of these trajectories can compromise user privacy. For example, LPAuditor [1] can accurately identify the user's home and workplace and can infer the sensitive locations visited by the user. Therefore, it is necessary to adopt effective privacy-preserving methods to pre-process trajectories before publishing or sharing them.

Since the spatial-temporal and thematic characteristics of trajectories have the potential to correlate trajectories with their creators as strong quasi-identifiers, a common approach is to aggregate trajectory points into geographic units which avoids exposing their original locations. However, recent studies have found that aggregation is neither effective in protecting trajectory privacy, nor does it reduce the spatial resolution and effectiveness of spatial analysis [2]. For example, De Montjoye et al. [3] reduced the resolution of human mobility trace datasets through spatial-temporal aggregation to protect individuals from identification, but coarsened datasets still do not provide sufficient anonymity. Therefore, to achieve more effective trajectory privacy protection, this paper needs to deal more specifically with the spatio-temporal characteristics of trajectory data. However, the trade-off between the effectiveness of trajectory privacy protection and the practicality of spatial and temporal analysis remains difficult to control, and there is no effective solution to this problem. Moreover, current research has focused on the spatial dimension of trajectories, while other semantics (e.g., temporal and thematic attributes) have been less studied. In fact, these features have been shown to be crucial for trajectory user identification. Moreover, the current approach relies heavily on manually designed programs. Once the procedure is disclosed, one may have the opportunity to recover the original trajectory data (e.g., using reverse engineering).

Publishing synthetic trajectories has become another important method to protect the trajectory privacy. Most of the existing studies are based on tree structure or Markov model trajectories, such as AdaTrace [4], TGM [5], etc. These models generate synthetic trajectories in two stages. They first manually extract statistical features from the real trajectories, such as -plots, subsequence counts, etc., and then reconstruct the synthetic trajectories with the extracted statistical features and Markov models. However, the manually feature extraction process cannot fully capture the spatio-temporal features and the hidden Markov model cannot capture the long-term spatio-temporal correlation of the real trajectories, which leads to the poor utility of the synthetic trajectories.

Deep learning models such as GAN and VAE have made significant progress in generating images, text, etc. in recent years, so new GAN-based trajectory generation models have been derived [6]. Among them, LSTM-TrajGAN [7] is a typical end-to-end deep learning framework for generating spatio-temporal trajectories. However, LSTM-TrajGAN has the following drawbacks. First, the loss function of LSTM-TrajGAN directs its generators to generate synthetic trajectories that are as similar as possible to the real trajectories, which may lead to privacy leakage of the model-generated trajectories. Second, the LSTM-TrajGAN model uses an LSTM (Long Short-Term Memory) to generate trajectories, and this LSTM cannot capture the long-term spatio-temporal dependence characteristics of locations.

To address the above drawbacks, this paper proposes ST-TrajGAN, a GAN-based trajectory generation model with transformer decoder as the generator and LSTM as the discriminator, and a new loss function Trajloss. the new loss function can reduce the privacy leakage in the training trajectories by relaxing the similarity constraint. In addition, the transformer decoder is used to prevent the generated synthetic trajectories from generating high utility losses. Compared with LSTM-TrajGAN, the model proposed in this paper can generate synthetic trajectories, which leads to a better trade-off between privacy and utility. The work in this paper is summarized as follows:

1. In this paper, a generative adversarial model with transformer decoder as the generator is proposed to generate synthetic trajectory data, which is a simple and highly secure process while generating synthetic trajectories with higher quality.
2. A trajectory encoding model for semantic trajectory data encoding is introduced and a new TrajLoss metric function is designed to measure the trajectory similarity loss of the deep learning model.
3. The ST-TrajGAN algorithm is evaluated using real LBS datasets to verify the utility of ST-TrajGAN. The experimental results show that ST-TrajGAN has a stronger privacy-preserving ability and higher data utility compared with current methods.

The rest of this paper is organized as follows: Section 2 discusses related works; some GAN-related background knowledge and problem definitions are introduced in Section 3; In Section 4, we propose a new Synthetic Trajectory Generation algorithm ST-TrajGAN; The experimental results are reported in Section 5; Section 6 concludes the paper.

# II. Related Work

The main approaches used to protect the privacy of trajectory data are the following. The first one is based on generalization techniques such as -anonymity [8], -diversity [9], and -closeness [10]. The second one is based on perturbation techniques such as -differential privacy [11] and -differential privacy [12]. New privacy-preserving schemes utilize machine learning-based methods for large-scale trajectory data, such as trajGANs [13], LSTM-TrajGAN [14], TrajGAIL [15], and TrajVAE [16]. In this section, these models are summarized and analyzed.

## Generalization-based methods K-Anonymity

The primary target of privacy-preserving schemes for trajectory data constructed by generalization-based approaches is to generalize the quasi-identifier attributes in trajectories so that it is impossible to distinguish one trajectory from other trajectories. -anonymity approach was proposed by Sweeney et al. [17], which mainly publishes data with low precision through generalization and suppression to ensure that each record has at least the same quasi-identifier attributes as other -1 records in the dataset and reduce privacy leakage from linking attacks. Tu et al. [18] first recognizes semantic attacks in published track datasets and proposes an algorithm that provides strong privacy protection against semantic and re-identification attacks while retaining high data utility. Shaham et al. [19] propose a new metric called transition-entropy and an attack model to investigate the location privacy preservation. Zhang et al. [20] present a DKM scheme to protect the users’ trajectory privacy for continuous LBSs and utilize the dynamic pseudonym mechanism and K-anonymity to improve the users’ trajectory privacy on the LSP. Mao et al. [21] employ spatial -anonymity to enhance user privacy. According to the cells that the user are required to query, the anonymizer selects cells to form a cloaking region, which can conceal the real user’s location.

The generalization-based approach is an effective privacy-preserving scheme that is usually relatively easy to implement and it handles trajectory data that can resist a certain level of linking attacks. However, the design of the generalization-based approach relies heavily on generalization capabilities, and it is also vulnerable to privacy attacks based on powerful background information.

## Perturbation-Based Approaches

Differential Privacy (DP) [22] is a mechanism that enables the sharing of only statistical characteristics that describe the database, without disclosing information specific to individuals and this approach is proposed as a solution for statistical database privacy breaches. It is a strict and measurable privacy security technique that can effectively reduce the impact of context-based attacks. Cunninghim et al. [23] proposed an -gram based local differential privacy technique with algorithmic improvements based on publicly available points of interest in the real world to enhance privacy retention and improve the utility of perturbed shared trajectory data. Andres et al. [24] proposed geographic indistinguishability to protect the real location information of LBS users. This approach not only prevents malicious guessing of the user's location but also limits the attacker's access to relevant background knowledge. Yuan et al. [25] proposed a DPTS-tree structure for data publishing that satisfies differential privacy while avoiding information inference attacks. Wang et al. [26] propose a novel differentially private trajectory data publishing algorithm with a bounded Laplace noise generation algorithm and a trajectory merging algorithm formally prove that the proposed scheme can achieve -differential privacy.

All the above generalization-based and perturbation-based approaches have some practical value in protecting trajectory privacy, but they also have some technical drawbacks. To protect privacy, these schemes aim to obfuscate trajectory locations and increase uncertainty. However, the trade-off between privacy and utility in trajectory data remains a difficult problem that has not yet been effectively addressed. Moreover, since previous trajectory privacy-preserving schemes rely too much on manually designed programs, attackers may use reverse engineering to recover the original trajectory data if their privacy-preserving mechanisms are attacked along with the publishing mechanisms [27].

## Machine learning-based approach

With the advancement of machine learning techniques, recent studies have started to use deep learning techniques to analyze behavioral patterns and location features in trajectories and then learn their distributions to generate large-scale trajectory data. Ouyang et al. [28] proposed a nonparametric trajectory generation model (trajGANs) that represents trajectories by gridding them and learning their distributions using GANs. Rao et al. [29] proposed a deep learning model combining LSTM and GANs (LSTM-TrajGAN) for generating trajectories with similar features to replace the original trajectories for data distribution. All of the above machine learning-based approaches can perform the trajectory generation task well and ensure the integrity of trajectory data features. However, the above models still have the following problems: (1) they cannot provide effective privacy protection; (2) the synthetic trajectories are only partially compatible with the distribution of the original trajectory data.

In summary, generalization and perturbation-based approaches are usually easy to implement in various scenarios, but many parameters and strategies need to be formulated manually. In contrast, machine learning-based approaches can produce more efficient and uncertain privacy-preserving techniques without more human involvement. Therefore, the transformer is applied to our model ST-TrajGAN to protect trajectory privacy. Its basic architecture combines GANs and LSTMs to process the original trajectories to synthesize distributions that are more similar to the original data.

# III. Preliminaries

## Basic Concepts

Generative Adversarial Network (GAN) is a deep learning model designed to learn from input data and generate new data with similar features. The GAN model consists of two neural networks: the Generator and the Discriminator.



1. ST-TrajGAN architecture diagram

The goal of the discriminator is to distinguish whether a sample is from the true distribution or from the generative model , so the discriminator network is actually a dichotomous classifier. Using the label to indicate that the sample is from the true distribution and to indicate that the sample is from the generative model, the output of the discriminator network is the probability that belongs to the true data distribution, i.e.

Then the probability that the sample comes from the generative model is

Given a sample with indicating whether it comes from or , the objective function of the discriminator network is to minimize the cross-entropy, i.e.

The objective of generating the adversarial network is the opposite of the discriminator network, i.e., to let the discriminator network discriminate the samples it generates as true samples.

## Definitions

***Definition 1:*** (Sampling location [3]). The sampling point of an individual is defined as a quadratic , where each element records the sampling latitude, longitude, date and hour, respectively.

***Definition 2:*** (Trajectory [5]). A trajectory is a sequence of sampled locations in temporal order, which can be expressed as: , where denotes the user, denotes the trajectory of the user, and denotes the length of the trajectory.

***Definition 3:*** (Trajectory dataset [15]). The set of trajectories composed of multiple trajectories, consisting of , where denotes the number of trajectories in the set.

## Problem Definition

Based on the above definitions, this section formulates the mobility trajectory generation problem as follows. Given a trajectory set and random noise obeying a specific distribution, and use them as the input of the generator to obtain the features of the original trajectory dataset after a large number of cycles of training, with the goal that the generated new synthetic trajectory are indistinguishable from the original trajectory set. Due to its privacy and practicality, this new trajectory dataset can be published instead of the original trajectory dataset.

# IV. ST-TrajGAN

The Semantic and transformer-based Trajectory Generative Adversarial Network (ST-TrajGAN) proposed in this paper consists of three main components: (1) A trajectory encoding model, which encodes GPS coordinates, temporal attributes and other attributes such as point-of-interest (POI) categories; (2) A trajectory generator , which takes random noise and the original set of trajectories as inputs and generates synthetic trajectories as outputs; (3) A trajectory discriminator, which takes the original set of trajectories and the set of synthetic trajectories generated by generator as inputs and determines whether they are "real" or "synthetic".

Fig.2 depicts the entire workflow of trajectory generation and publication. The goal is to train an "intelligent" trajectory generator to generate "real" synthetic trajectories to replace the original trajectories, which preserves different privacy level in trajectory analysis tasks such as trajectory-user linking (TUL) and trajectory data mining (e.g., work/home location clustering). At the same time, the quality of multiple spatial or temporal summary analysis tasks is guaranteed. Such a



1. The overall workflow of ST-TrajGAN

framework can be used as a trajectory privacy protection layer in trajectory data collection, processing, and publishing trajectories, publishing synthetic alternatives to real trajectory data that may reveal personal privacy.

## Trajectory encoding model

1. **Trajectory Encoding**

First, this paper introduces a trajectory encoding model that transforms the original trajectories into a specific encoding space as the input to the ST-TrajGAN model. The main reason for the encoding process is that trajectory data usually contain various types of attributes, such as interval data (e.g., GPS coordinates, date and time), nominal data (e.g., POI category), and ordinal data (e.g., POI level), which can only be effectively utilized in training deep learning models by converting them into valid numerical representations. The trajectory encoding model in this paper consists of two parts: trajectory point encoding and trajectory filling.

The trajectory point encoding process is shown in Fig.3. The semantic trajectory points contain the following attributes: user ID, location, time, trajectory ID, and other optional attributes such as the POI category. For the location attribute, this paper uses the center of mass of all trajectories in the dataset to normalize all latitudes and longitudes to obtain the deviation of latitude and longitude from the center of mass. In this way, the model can better understand the spatial deviation patterns between different trajectory points. These deviation values will be used as numerical representations of trajectory points for constructing spatial embeddings.

For temporal and categorical attributes, this paper uses one-hot encoding (a machine-learning representation process that converts categorical variables into numerical variables) to encode them as high-dimensional binary vectors based on the vocabulary size of the attributes. For example, the "Day" attribute is encoded as a 7-dimensional binary vector, and "Tuesday" is represented as . Similarly, the "Month" attribute is encoded as a 12-dimensional binary vector, and the "Category" attribute is encoded as a 10-dimensional binary vector. Note that User ID, Location ID, and Trajectory ID are not encoded in this paper, because they are only used to indicate the user and trajectory to which the point belongs.



Fig.3. Trajectory encoding model



Fig.4. The neural network model of ST-TrajGAN

1. **Trajectory Filling**

After the trajectory points are encoded, all spatial, temporal, and thematic attributes of the trajectories are stored in a multidimensional matrix whose first dimension represents the index of each trajectory. Since the length of each trajectory data (i.e., the number of location points) is a variable, this paper then applies a trajectory padding technique to ensure that all trajectories have the same length as the longest trajectory. Specifically, this paper uses zero pre-padding to populate each trajectory with empty trajectory points (i.e., points whose attributes are all set to zero) until all trajectories reach the same length as the longest trajectory in the dataset. The key reason for taking this approach is that trajectory data of the same size can be used for batch processing and training deep learning models, which helps to speed up the training process. During model training and inference, the trajectory points that have been populated will be masked (i.e., cut) and therefore will not have a substantial impact on the weight update and inference results of the neural network.

## ST-TrajGAN model

Fig.4 depicts the neural network structure of the ST-TrajGAN model. The trajectory generator learns the distribution characteristics and mobility patterns of the real trajectory data, followed by synthesizing the trajectory data based on the input original trajectories and random noise. In addition, the trajectory discriminator distinguishes whether the trajectory samples come from the training set (i.e., real trajectory data) or the trajectory generator (i.e., synthesized trajectory data). The goal of the trajectory generator is to generate "high-quality" synthetic trajectories that can "fool" the trajectory discriminator, which leads to a zero-sum game between them. The generated synthetic trajectories are designed to be capable of spatial and temporal aggregation, but also have some randomness to protect sensitive information in trajectory analysis tasks with privacy concerns. This strategy is reflected in the design and optimization of the ST-TrajGAN model.

1. **Trajectory Generator**

The trajectory generator consists of four functional layers: the embedding layer, the feature fusion layer, the transformer decoder layer, and the regression/classification layer. The embedding layer is an MLP that takes the preprocessed vectors

as input, where is the length of the trajectory. After preprocessing, the trajectories are set to have the same length.

For the spatial dimension of the trajectories (i.e., for latitude and longitude deviations), MLPs with 64 nodes are used to embed each pair of them, and the trajectories for the temporal dimension are embedded according to the corresponding MLP node settings. For temporal dimensions (e.g., day and hour) and category attributes (e.g., POI categories), this model uses MLPs to embed them separately and obtain fixed-length vectors based on their vocabularies.

where are the latitude and longitude deviations of the trajectory point; ，， denote the one-hot vectors of day, hour, and category attributes of the trajectory point; ，，， represent the MLP and ReLU activation functions for embedding spatial, daily, hourly, and category attributes; ，，， are the embedding weight matrices of these MLPs; 、、 and are the embedding vectors for each attribute, respectively. Note that the embedding weight matrix is shared among all trajectory points. The final output of the embedding is obtained by merging the embedding vectors of all trajectory points as shown in (9):

The feature fusion layer is also an MLP that takes as input all embedding vectors of random noise and splices together and is used to fuse all attributes of the trajectory points to support the modeling and generation of spatio-temporal trajectories.

The transformer decoder network is chosen to capture the long-term correlation of trajectories. It consists of a location encoding, a multi-headed self-attentive model, Add and Norm layers and a feedforward neural network. The input vectors are represented by the fusion vectors and the position-encoded vectors are summed. The multi-headed self-attentive module mainly uses the self-attentive network to perform the corresponding dot product operations on the input vectors, and the multi-headed ensures that the transformer encoder can pay attention to the information in different subspaces. Specifically, the vectors are multiplied by three self-attentive model matrices, i.e., the query matrix , the key matrix , and the value matrix to obtain the key vector , the value vector , and the query vector , where ; the self-attentive model learns the feature relationships between the vectors by equations (10).

where is the dimensionality of each vector. After Add and Norm layers and feedforward neural network, the output of the transformer decoder layer .

Finally, this paper decodes the output of the transformer decoder network layer. To decode the latitude and longitude deviations, this paper uses a fully connected layer with two neurons and applies the tanh activation function. For the temporal attributes, a fully connected layer with neurons of lexical size for each attribute and a softmax normalized exponential function are used to decode into one-hot vectors corresponding to the temporal attributes. The decoding process can be roughly expressed as follows:

where is the latitude and longitude deviation of the synthetic trajectory point; denote the fully connected layers of the corresponding activation functions; ，， ( is the number of nodes of the transformer decoder and is the vocabulary size of each attribute) denote the decoding weight matrix of the fully connected layer.

According to the objective function , the loss function of the generator can adopt the Binary Cross-Entropy (BCE) loss function (). However, unlike the original GAN, this paper needs real trajectory data as input. For this reason, a new loss measure function TrajLoss is designed in this paper, based on which the corresponding loss function is established for training the generation algorithm by calculating the similarity loss between the real trajectory data and the synthetic trajectory data in multiple dimensions such as space, time and category. TrajLoss is defined as follows.

where and represent the trajectory prediction results of the underlying true labels and the discriminator, respectively; and represent the true trajectories and the corresponding synthetic trajectories; is the original binary cross-entropy loss from the discriminator; , and are the spatial similarity loss, temporal similarity loss, and category similarity loss between the true and synthetic trajectories, respectively; , , and are the weights of these losses, and different weights can be assigned according to different situations.

The spatial similarity of trajectories is measured in this paper using cosine similarity instead of the L2 loss function, which can reduce the privacy leakage of training trajectories by increasing the value of the loss function of the generator. As mentioned earlier, the loss function of the generator can directly affect the quality of the synthesized trajectories. In general, the loss function of the algorithm should be as small as possible to generate more similar trajectories. However, cosine similarity is used to measure trajectories, and the more similar two trajectories are, the larger the value will be. In this way, the value of the loss function of the generator can be limited, and the leakage of trajectory privacy by the generator can be reduced.

1. **Trajectory Discriminator**

The structure of the trajectory discriminator is very similar to that of the trajectory generator . The main differences between them are:

a) The trajectory discriminator requires only input trajectory data (no random noise);

b) A many-to-one LSTM model is used, where features with time steps are used as input and a scalar is used as output: the

where is the fused features of all trajectory points in the trajectory (i.e., ), where is the fused feature vector of the trajectory point; is the weight matrix of the LSTM model; and is the output scalar of the LSTM model.

This section uses a one-unit layer with a sigmoid activation function for binary classification (real or synthetic) of the scalar output:

where is the unit-dense layer with a sigmoid function used for binary classification and is its weight matrix; is the final output of the discriminator.

The binary classification task of the discriminator is much easier. To balance the training process for effective learning of the whole framework, this paper uses binary cross entropy loss (BCE Loss) as the loss function to determine whether the trajectory is true or false.

# VI. Experiments

In this paper, the ST-TrajGAN algorithm is compared with two deep learning algorithms proposed for trajectory privacy protection. The first algorithm is trajGAN proposed by Liu et al. [5], and the other algorithm is LSTM-TrajGAN proposed by Rao et al. [6]. We implement the above algorithms in Python 3.7 with Intel Core i5 CPU 2.3GHz and 8GB RAM, running with Windows 10.

## Datasets

This paper uses three real-world trajectory datasets Foursquare NYC, Geolife, and Gowalla datasets for experiments, and only retains user ID, trajectory ID, location, hour, date, and category attributes, and removes other attributes (such as interest point level), all attributes are shown in Table I below. There are 193 users, 3079 trajectories, and 66962 trajectory points in the dataset. In this paper, 2/3 of the data are used for the training of the ST-TrajGAN model, and 1/3 of the trajectories are used for testing.

## Training and evaluation

1. Summary of Foursquare NYC weekly trajectory dataset

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Number/Range |
| Trajectory | Integer | 3079 |
| User ID | Integer | 193 |
| Latitude | Float | （40.550852，40.988332） |
| Longitude | Float | （-74.269644，-73.685767） |
| Hour | Integer | 24 |
| Day | String | 7 |
| Category | String | 10 |

In this paper, the ST-TrajGAN model is trained on the training set with 2000 epochs using several default training hyperparameters (e.g., the learning rate is set to 0.001, the batch size is set to 256, the training count is 5000, and the optimizer is Adam). After the training, the trajectory data from the test set and the random noise are used as the input of the generator to obtain the synthetic trajectory data. Next, a state-of-the-art TUL algorithm, MARC (multifaceted trajectory classifier), is used to perform the TUL task on the test data and the synthetic data in this paper. This paper uses five commonly used metrics to evaluate the accuracy of TUL: ACC@1 (top-1 precision, which shows the ability of the model to treat correct labels as the most likely label candidates), ACC@5 (top-5 precision, which shows the ability of the model to have correct labels among the top 5 most likely label candidates), Macro-p (macro precision), Macro-R (macro recall rate, the average recall across all categories), and Macro-F1 (the summed average of Macro-P and Macro-R). For comparison, random perturbations (spatial filter: within 1km; temporal filter; within 24 hours) and Gaussian geomask (spatial filter: standard deviation = 0.001; temporal filter: within 24 hours) were also evaluated in this paper.

## Utility Measures

Since this model considers timestamp attributes during training, its temporal attributes also need to be considered when performing trajectory utility analysis. In this section, location and timestamp are used as trajectory points, and Hausdorff distance and trajectory sharing percentage are used to measure the similarity of the synthetic trajectory to the original trajectory.

1. **Hausdorff Distance (HDD):**

The Hausdorff distance is a measure of the distance between two sets and can be applied to measure the spatial dissimilarity between two trajectories. It is defined as follows:

where , is called the two-way Hausdorff distance, is called the one-way Hausdorff distance from point set A to point set B, and is the same as above.

1. **Trajectory Sharing Percentage (TSP):**

Trajectory sharing percentage measures the similarity between each synthetic trajectory and the real trajectory and finally averages all the similarities, the higher the TSP proves that the synthetic trajectory is more similar to the real trajectory.

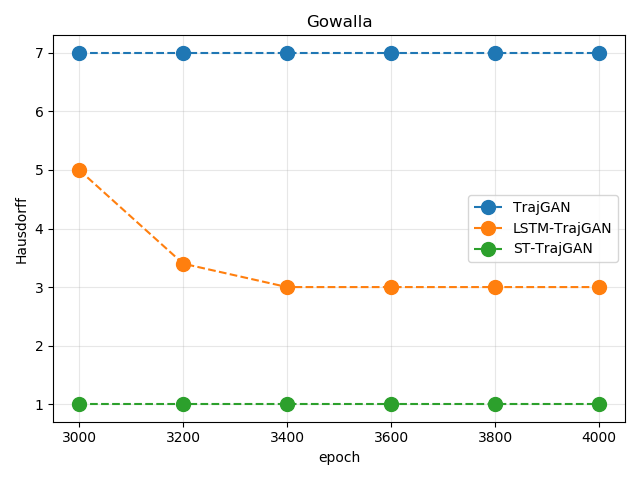
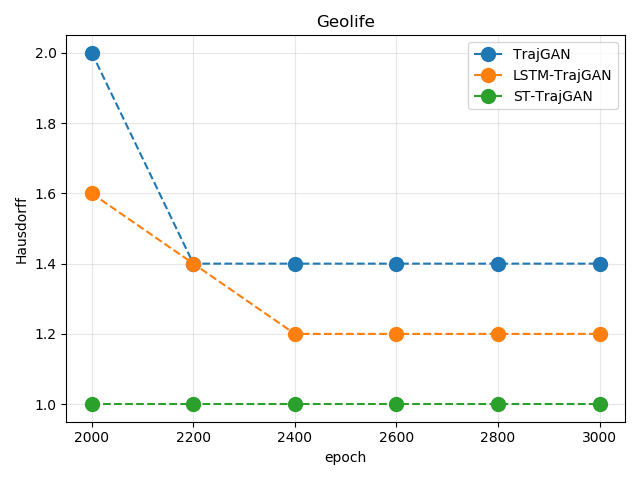
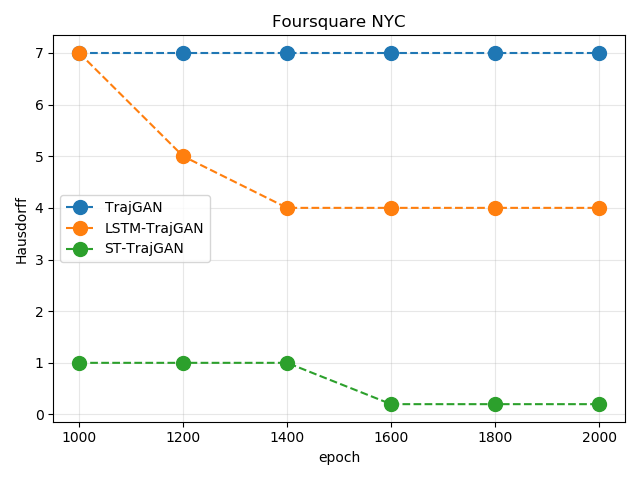
## Experiment Results

1. **Privacy Evaluation**

Table II shows the TUL values of the three different models on the three datasets. the smaller the value of TUL, the more difficult it is for the user to be linked, i.e., the more private the trajectory is. From the values in Table II, it can be seen that the TrajGAN model has the weakest privacy protection for trajectories among the three models, as it has the highest various TUL values. Numerically, compared to the LSTM-TrajGAN model in the NYC dataset, the model in this paper decreases by 21% for ACC@1, by about 62% for ACC@5, and by about 25% for Macro\_P, Macro\_R, and Macro\_F1. It is obvious that the model in this paper does greatly improve the privacy of the generated trajectories compared to the LSTM-TrajGAN model. The ST-TrajGAN model has the lowest TUL value compared to the other two models. This indicates that the trajectories generated by the ST-TrajGAN model are the most difficult to link, i.e., the privacy-preserving ability is the strongest.

1. The privacy protection effect of different models on TUL task

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | NYC | | | Geolife | | | Gowalla | | |
| Traj-  GAN | LSTM-TrajGAN | ST-TrajGAN | Traj-GAN | LSTM-TrajGAN | ST-TrajGAN | TrajGAN | LSTM-TrajGAN | ST-TrajGAN |
| ACC@1 | 0.376 | 0.137 | 0.108 | 0.309 | 0.216 | 0.181 | 0.382 | 0.189 | 0.140 |
| ACC@5 | 0.672 | 0.327 | 0.123 | 0.744 | 0.643 | 0.536 | 0.610 | 0.465 | 0.301 |
| Macro\_P | 0.334 | 0.126 | 0.094 | 0.216 | 0.170 | 0.116 | 0.241 | 0.101 | 0.064 |
| Macro\_R | 0.381 | 0.142 | 0.101 | 0.236 | 0.231 | 0.182 | 0.243 | 0.108 | 0.091 |
| Macro\_F1 | 0.359 | 0.122 | 0.095 | 0.214 | 0.152 | 0.126 | 0.365 | 0.172 | 0.141 |



1. Hausdorff distance on different datasets
2. **Utility Evaluation**
3. Hausdorff distance

The Hausdorff distances shown in Table III are the average of the Hausdoff distances over the number of iterations for each data set in Fig.5. Fig.5 shows a visualization of the Hausdorff distances. As can be seen in Fig.5, the Hausdorff distance for each model decreases as the number of training iterations increases in each of the three datasets. However, the combined comparison shows that the ST-TrajGAN model has the smallest Hausdorff distance.

This indicates that the synthetic trajectory dataset generated by the ST-TrajGAN model is most similar to the original trajectory dataset. Here, the present algorithm considers not only the spatial characteristics of trajectories but also the temporal characteristics of trajectories. In the Geolife dataset, although the Hausdorff distance of the present model is larger than that of the LSTM-TrajGAN model, the difference is not very large, only 1%. This indicates that the trajectories generated by the ST-TrajGAN model are most similar to the original trajectories when the spatio-temporal factors are considered.

1. Trajectory sharing percentage

The TSP values given in Table III are the average of all trajectories in the dataset. From Table III, it can be seen that the TSP of the model is highest for the NYC and Gowalla datasets, reaching 85.73% and 80.66%, respectively.

Fig.6 shows the TSP of TUL values for Acc@5 as well as for different datasets. where (a) and (b) are for the NYC dataset, (c) and (d) are for the Geolife dataset, and (e) and (f) are for the Gowalla dataset. From the performance of the different models in Fig.6, it can be seen that the ST-TrajGAN model has a TSP value over 80% on the NYC dataset, which is more than 20% on average than the LSTM-TrajGAN model. With the same TUL, the ST-TrajGAN model has the highest percentage of trajectory sharing on the NYC dataset. Also, ST-TrajGAN has lower TUL values, and the same trend is observed for the Geolife and Gowalla datasets. This further indicates that the ST-TrajGAN model not only improves trajectory privacy but also ensures trajectory utility.

1. Frequency distribution of visits

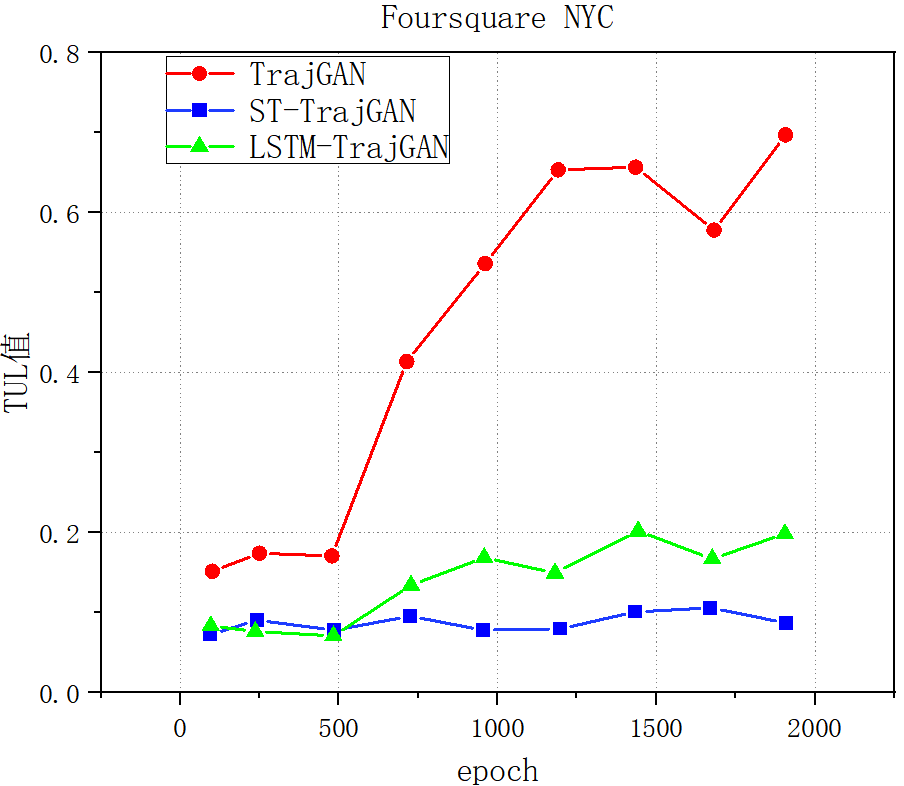
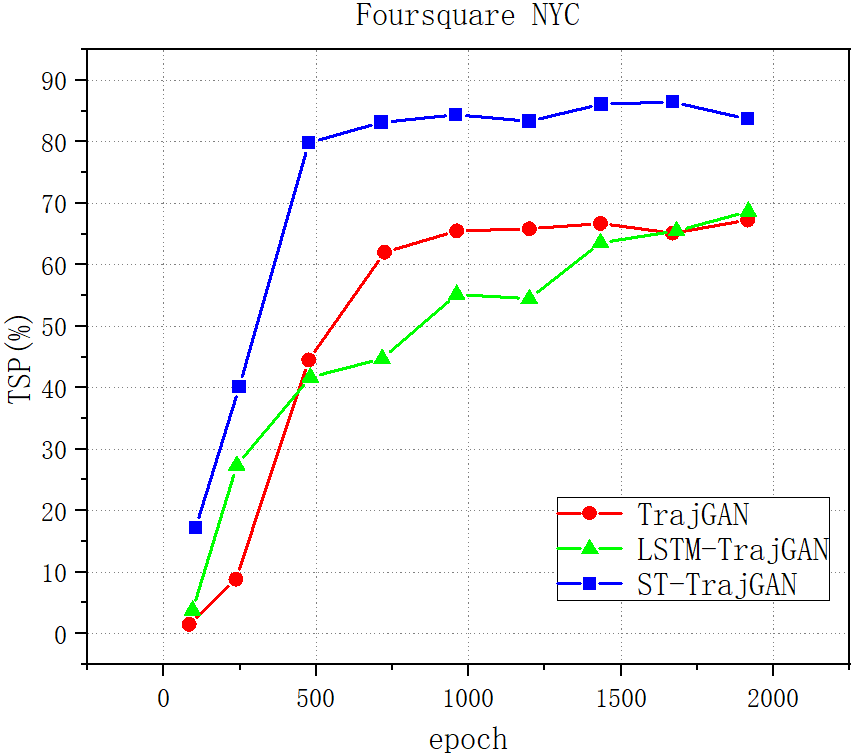
This paper also explores temporal characteristics based on the visualization of two summary metrics: the temporal access probability distribution and the total temporal access frequency distribution for each POI category. In this paper, three different methods are adopted to count the visit frequency of each POI category in the original and synthetic trajectories on a time-by-time basis and convert them into a probability distribution matrix for analyzing and comparing temporal patterns and temporal similarities.

The results show that the temporal visit probability distributions of ST-TrajGAN have a large commonality with the temporal visit probability distributions of the original data, reflecting significant temporal similarity. some of the results of ST-TrajGAN have visit probabilities close to zero because these categories rarely appear in the training dataset, and thus the model does not learn enough information to make intelligent predictions. For comparison, the temporal visit probabilities from the random perturbation and Gaussian distributions show neither temporal similarity to the original data nor significant temporal patterns beyond 24 hours (except for the event categories).

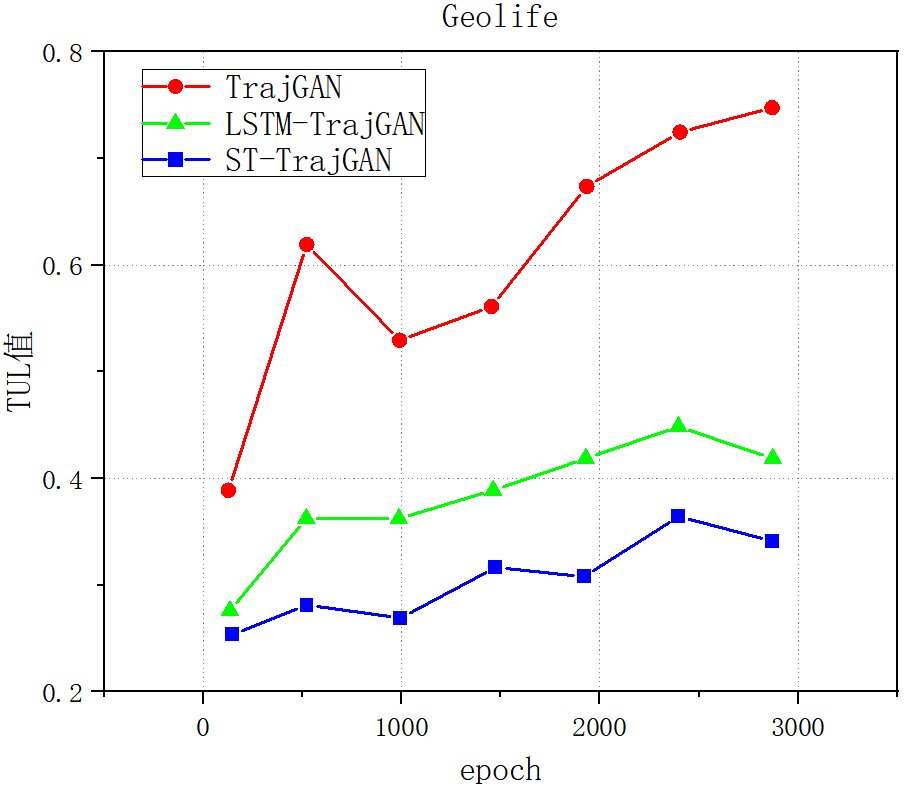
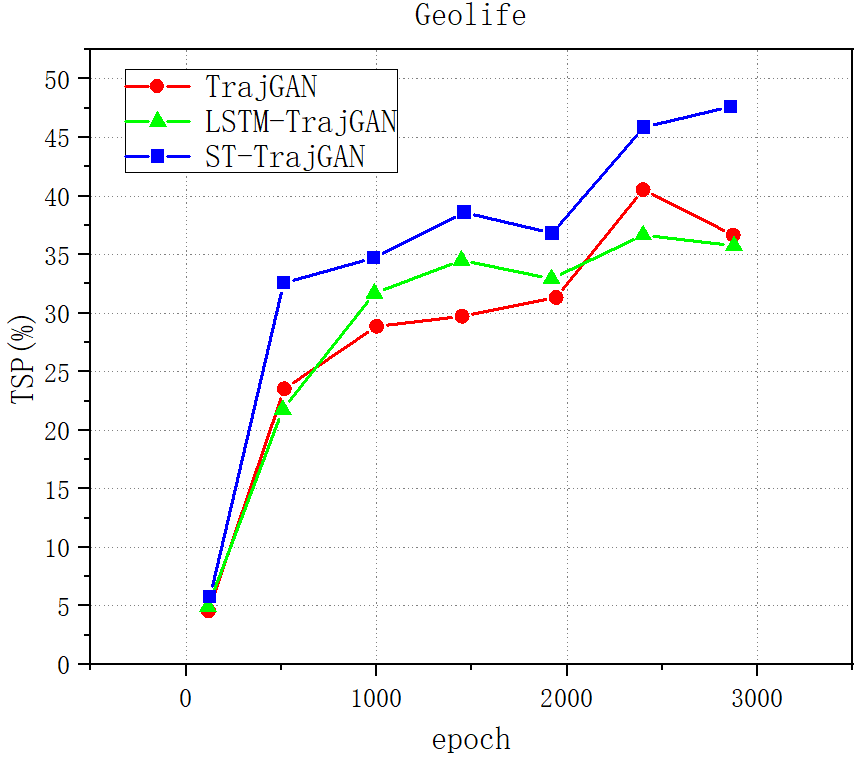
In addition, this paper investigates the overall temporal and categorical visit frequency distributions (Fig.7(a) and Fig.7(b)). The model's overall temporal visit frequency distribution (Pearson coefficient: 0.761) fits the original data (Pearson coefficient: 0.761) better than the random perturbation (0.536) and Gaussian mask (0.535). The overall categorical visit frequency distribution also fits well (0.889). Therefore, this paper concludes that the models in this paper generally preserve temporal and categorical features well. This section discusses the factors that may affect the privacy-preserving effectiveness of the ST-TrajGAN model and the trade-off between privacy-preserving effectiveness and utility. Finally, this paper discusses the limitations of the approach in this paper.

1. Spatial distribution similarity

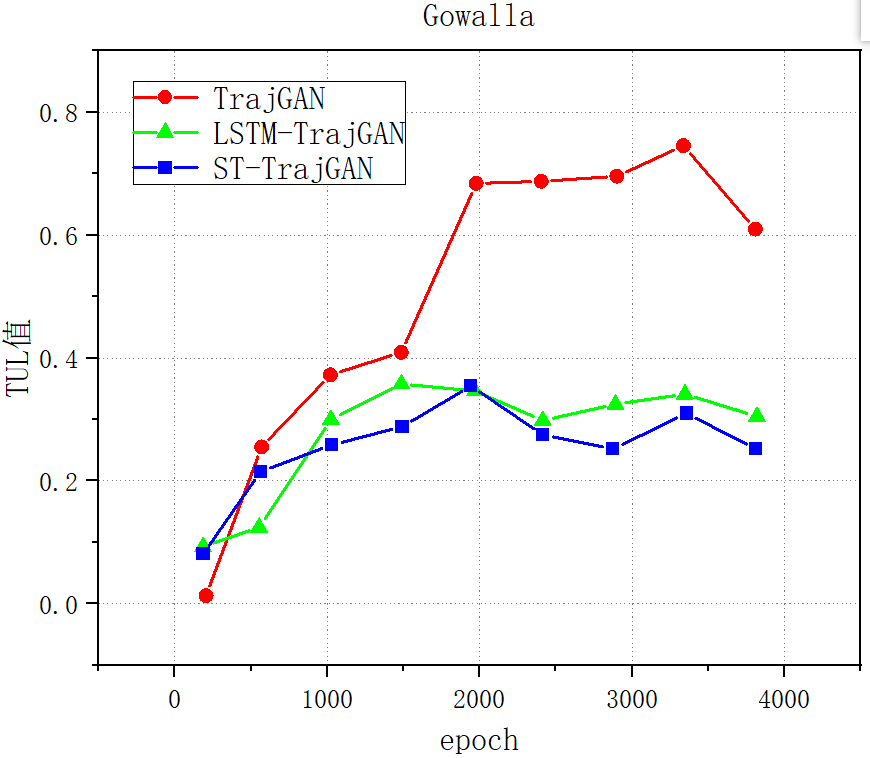
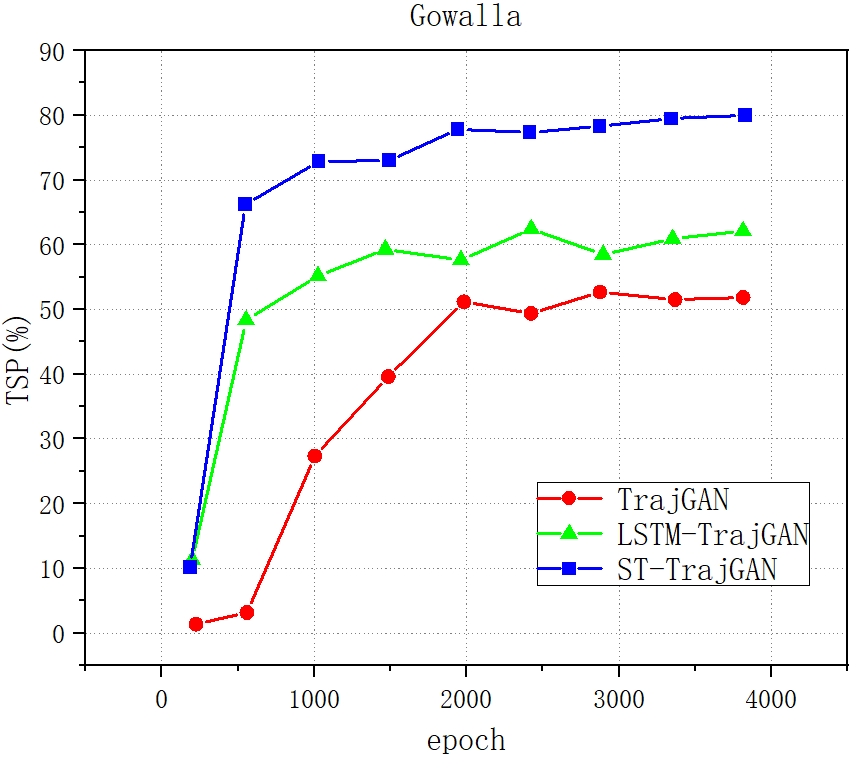
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | Hausdorff distance | | | Trajectory sharing percentage | | |
| NYC | Geolife | Gowalla | NYC | Geolife | Gowalla |
| TrajGAN | 7.115 | 1.623 | 7.102 | 69.7% | 34.7% | 53.7% |
| LSTM-TrajGAN | 5.342 | 1.064 | 3.001 | 80.6% | 40.7% | 68.9% |
| ST-TrajGAN | 0.661 | 1.002 | 1.008 | 86.5% | 41.9% | 73.1% |



|  |  |  |  |
| --- | --- | --- | --- |
| （a）TSP | | （b）TUL | |
|  |  | |

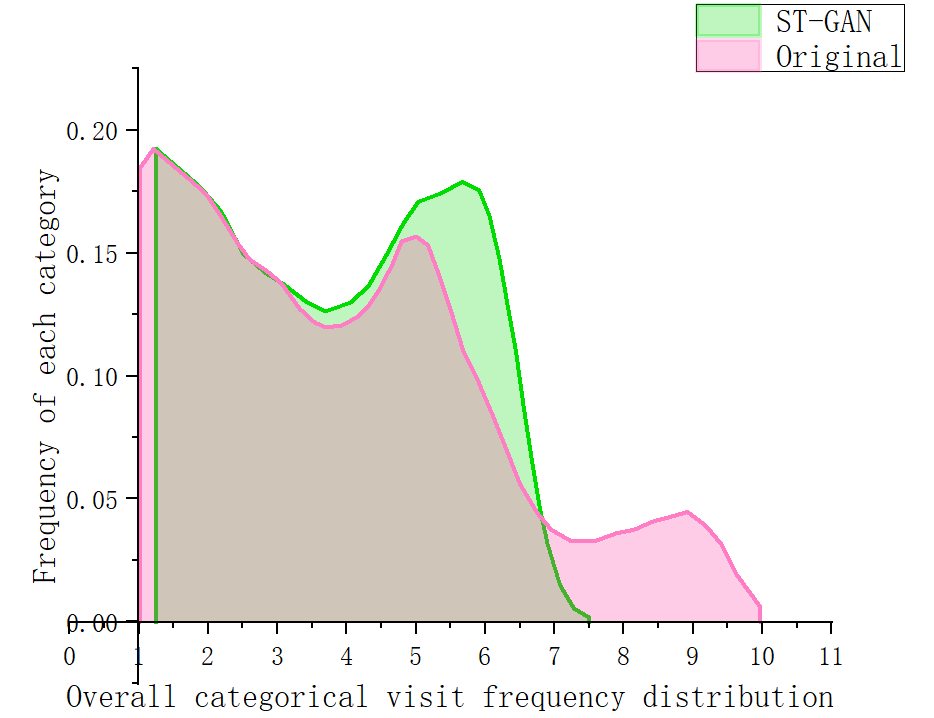
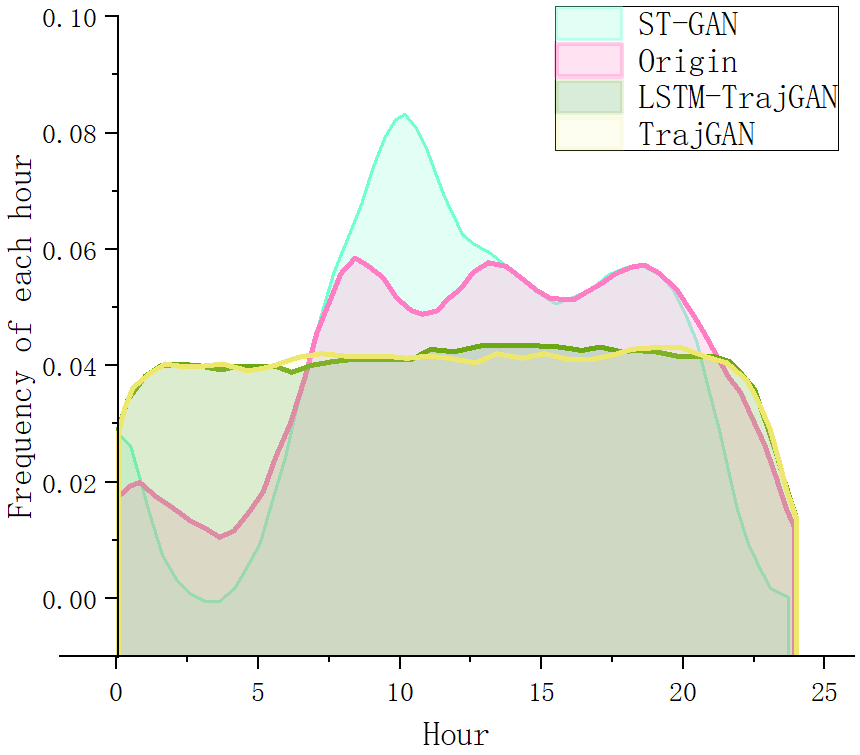


|  |  |
| --- | --- |
| （c）TSP | （d）TUL |



|  |  |
| --- | --- |
| （e）TSP | （f）TUL |

1. The relationship between TSP and TUL of the different datasets.



|  |  |
| --- | --- |
| （a） | （b） |

1. Overall time-visit frequency distribution (a) and overall categorical visit frequency distribution(b)
2. **Privacy Utility Balance**

This paper first explores how different learning rates, loss metric functions, and random noise data affect the metric scores in the TUL task compared to the baseline setting (i.e., ST-TrajGAN learning rate = 0.001; spatial dimension = 64; and TrajLoss metric function during training). As shown in Table IV, different random noise data have less impact on the metrics, which actually contributes to the potential generalizability of the proposed method for generating privacy-preserving trajectory data. We also find that the choice of learning rate may have a significant impact on the metrics. The higher the learning rate (0.002), the faster the model converges and the less uncertainty in the generated synthetic trajectories, which have more characteristics than the original trajectories, resulting in higher TUL metric scores, and vice versa. Although this is not always the case, the learning rate should be carefully set to balance trajectory utility and privacy-preserving effectiveness.

In addition, this paper investigates the contribution of the TrajLoss metric function to training. When Spatial Loss or Temporal Loss is removed from the TrajLoss function, the metric value drops sharply, which means that the synthetic trajectory does not maintain the spatial or temporal characteristics of the original trajectory. In contrast, removing the classification loss has a limited impact on the metric scores. There is no doubt that removing the entire TrajLoss function leads to the loss of spatio-temporal features, resulting in the lowest TUL metric score. It is concluded that the spatial and temporal dimensions represent the essential characteristics of the trajectory and thus need to be explicitly considered in the privacy-preserving approach.

Since the embedding of temporal and category attributes is based on the size of their vocabularies, this paper focuses on spatial embedding. Commonly used spatial embedding methods are the multilayer perceptron (MLP) and the Geohash algorithm. For example, Gupta et al. [30] used MLP to embed the location of each person to obtain a fixed-length vector and used this vector as an input to an LSTM model to generate human trajectories. Petry et al. [31] introduced a binary Geohash algorithm, where they first used the Geohash algorithm to divide the region into grid cells, then encoded the latitude and longitude as strings, and finally converted the string converted into a binary fixed-length vector as a representation of the spatial dimension of each trajectory point.

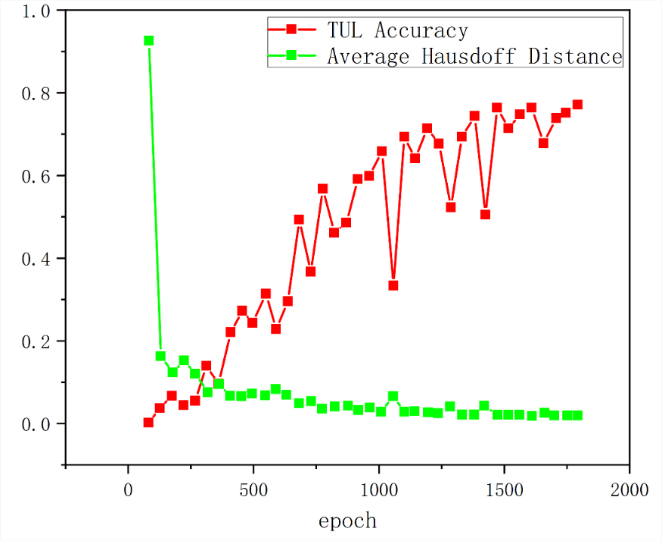
Table IV. Metrics in the TUL task based on ST-TrajGAN using different training and optimization settings and different dimensions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ST-TrajGAN | ACC@1 | ACC@5 | Macro-F1 | Macro-P | Macro-R |
| Baseline | 0.459 | 0.722 | 0.381 | 0.429 | 0.428 |
| Different Random Noise | 0.466 | 0.742 | 0.398 | 0.451 | 0.436 |
| Higher Learning Rate (0.002) | 0.843 | 0.969 | 0.824 | 0.855 | 0.828 |
| Lower Learning Rate (0.00002) | 0.052 | 0.157 | 0.029 | 0.047 | 0.054 |
| Higher Spatial dimensions (128) | 0.510 | 0.811 | 0.504 | 0.513 | 0.513 |
| Lower Spatial dimensions (32) | 0.416 | 0.703 | 0.386 | 0.412 | 0.392 |
| TrajLoss without Spatial Loss | 0.047 | 0.176 | 0.030 | 0.037 | 0.042 |
| TrajLoss without Temporal Loss | 0.093 | 0.252 | 0.076 | 0.119 | 0.089 |
| TrajLoss without Categorical Loss | 0.354 | 0.613 | 0.311 | 0.385 | 0.346 |
| No TrajLoss | 0.010 | 0.032 | 0.002 | 0.001 | 0.007 |

This paper uses MLP to embed spatial dimensions in the generator and discriminator but does so in a different way. Instead of embedding the coordinates directly, this paper first derives the deviations of latitude and longitude from the center of mass of all trajectory locations, and then embeds these deviations into a 64-dimensional vector using MLP. There are two considerations: (1) on the one hand, unlike the trajectory classification task, the goal of this paper is to generate synthetic trajectories, which means that this paper needs to decode the coordinates from the hidden features in the model, and thus the use of binary Geohash may lead to difficulties in learning an efficient representation of the coordinates, designing appropriate spatial losses, and error back-propagation; (2) on the other hand, unlike the Cartesian coordinate system describing the constrained prediction region, the prediction region in the task of this paper is at the urban scale, where the difference between the two GPS coordinates appears only after the decimal point. It would be a great challenge for the model to learn and predict coordinates with only minor variations. Therefore, this paper standardizes the coordinates so that the difference between the two locations is more important for the model to learn. Current studies also suggest that scattering locations based on biases may help preserve privacy.

This paper also explores how the spatial embedding dimension affects the metrics in the TUL task. As shown in Table IV, embedding location information into a high-dimensional vector (e.g., 128) can improve the TUL metric score and vice versa. This makes sense because vectors in high-dimensional space are usually able to extract and embed more information than vectors in low-dimensional space. However, it also involves a trade-off between localization accuracy and computational effort due to physical equipment limitations.

In general, specific trajectory analysis tasks may rely on different types of trajectory data (e.g., POI-based or road network-based) or different requirements (e.g., road extraction requires the location of each trajectory point to be precise), which makes it challenging to design generic privacy-preserving methods. However, a method can be evaluated by some specific criteria to determine its application scenarios and even design a method to cover as many scenarios as possible based on this consideration. Inspired by evaluation frameworks involving privacy, analytics, and uncertainty, this paper investigates the relationship between privacy-preserving effectiveness and utility. Considering this relationship will help to select and design a suitable trajectory privacy-preserving method for a specific scenario to protect privacy while still retaining some similarity as a good choice for spatio-temporal modeling or analysis. As an end-to-end deep learning model, ST-TrajGAN can monitor and quantify this relationship during training and help find the best-balanced parameter settings. For example, as training proceeds, the TUL accuracy (Top-5 accuracy) increases while the Average Hausdorff Distance decreases (Fig.8). Based on this relationship, careful selection of model weights for different periods can ensures that the synthetic trajectory retains a certain degree of spatio-temporal characteristics while maintaining a low TUL accuracy when



1. Trade-off between the utility of privacy preservation (expressed by TUL Top-5 accuracy) and spatial feature retention (expressed by average Hausdorff distance)

needed, thus balancing the privacy-preserving effect and utility of the synthetic trajectory.

# VIII. Conclusion

In summary, we propose a new ST-TrajGAN method, which is an end-to-end deep learning model for generating privacy-preserving synthetic trajectory data. Because the Transformer decoder outperforms most RNNs, such as LSTM and GRU, in modeling long-term dependencies, it is more suitable for trajectory data generation and publishing. In this paper, a loss metric function TrajLoss is designed to measure trajectory similarity loss for model training and optimization, which can also solve the privacy leakage problem of the training dataset. The model is evaluated on a trajectory-user linking task on a real-world semantic trajectory dataset. Compared with other common location privacy-preserving models, the model proposed in this paper can better protect users from re-identification attacks, in addition to retaining the basic spatio-temporal and thematic features of real trajectory data. ST-TrajGAN better balances the effectiveness of trajectory privacy protection and the practicality of spatial and temporal analysis, and provides a new idea for GeoAI-based privacy protection.

Future work will focus on improving the loss function for measuring trajectory similarity, extending the framework of this paper to a global range of trajectory datasets and generating custom variable-length synthetic trajectory data. In addition, we will further explore potential privacy attacks and defense strategies, and evaluate the effectiveness and usefulness of the model in this paper for privacy preservation in other trajectory data mining and analysis tasks.

References

1. Drakonakis K, Ilia P, Ioannidis S, et al. Please forget where I was last summer: The privacy risks of public location (meta) data[J]. arXiv preprint arXiv:1901.00897, 2019.
2. Shokri R, Theodorakopoulos G, Papadimitratos P, et al. Hiding in the mobile crowd: Locationprivacy through collaboration[J]. IEEE transactions on dependable and secure computing, 2013, 11(3): 266-279.
3. Liu X, Chen H, Andris C. trajGANs: Using generative adversarial networks for geo-privacy protection of trajectory data (Vision paper) [C]//Location privacy and security workshop. 2018: 1-7.
4. Gursoy M E, Liu L, Truex S, et al. Utility-aware synthesis of differentially private and attack-resilient location traces[C]//Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018: 196-211.
5. Ghane S, Kulik L, Ramamohanarao K. TGM: A generative mechanism for publishing trajectories with differential privacy[J]. IEEE Internet of Things Journal, 2019, 7(4): 2611-2621.
6. Rao J, Gao S, Kang Y, et al. LSTM-TrajGAN: A deep learning approach to trajectory privacy protection[J]. arXiv preprint arXiv:2006.10521, 2020.
7. De Montjoye Y A, Hidalgo C A, Verleysen M, et al. Unique in the crowd: The privacy bounds of human mobility[J]. Scientific reports, 2013, 3(1): 1-5.
8. L. Sweeney, k-anonymity: A model for protecting privacy, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (05) (2002) 557–570.
9. A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, ldiversity: Privacy beyond k-anonymity, ACM Trans. Knowl Discov. Data1 (1) (2007) 3–es.
10. N. Li, T. Li, S. Venkatasubramanian, t-closeness: Privacy beyond kanonymity and l-diversity, in: 2007 IEEE 23rd International Conference on Data Engineering, IEEE, 2006, pp. 106–115.
11. C. Dwork, Differential privacy: A survey of results, in: International Conference on Theory and Applications of Models of Computation, Springer, 2008, pp. 1–19.
12. C. Dwork, G.N. Rothblum, S. Vadhan, Boosting and differential privacy, in: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, IEEE, 2010, pp. 51–60.
13. X. Liu, H. Chen, C. Andris, trajGANs: Using generative adversarial networks for geo-privacy protection of trajectory data (Vision paper), in: Location Privacy and Security Workshop, 2018, pp. 1–7.
14. J. Rao, S. Gao, Y. Kang, Q. Huang, LSTM-TrajGAN: A deep learning approach to trajectory privacy protection, 2020, arXiv preprint arXiv:2006.10521.
15. S. Choi, J. Kim, H. Yeo, Trajgail: Generating urban vehicle trajectories using generative adversarial imitation learning, Transp. Res. C 128 (2021)103091.
16. X. Chen, J. Xu, R. Zhou, W. Chen, J. Fang, C. Liu, TrajVAE: A variational AutoEncoder model for trajectory generation, Neurocomputing 428 (2021)332–339.
17. L. Sweeney, k-anonymity: A model for protecting privacy, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (05) (2002) 557–570.
18. Tu Z, Zhao K, Xu F, et al. Protecting trajectory from semantic attack considering k-anonymity, l-diversity, and t-closeness[J]. IEEE Transactions on Network and Service Management, 2018, 16(1): 264-278.
19. S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, J. Li, Privacy preservation in location-based services: A novel metric and attack model, IEEE Trans. Mob. Comput. 20 (10) (2020) 3006–3019.
20. Zhang S, Li X, Tan Z, et al. A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services[J]. Future Generation Computer Systems, 2019, 94: 40-50.
21. Mao X, Zhang S, Choo K K R, et al. A trajectory privacy-preserving scheme based on a dual-K mechanism for continuous location-based services[J]. Information Sciences, 2020, 527: 406-419.
22. C. Dwork, Differential privacy: A survey of results, in: International Conference on Theory and Applications of Models of Computation, Springer, 2008, pp. 1–19.
23. T. Cunningham, G. Cormode, H. Ferhatosmanoglu, D. Srivastava, Real-world trajectory sharing with local differential privacy, 2021, arXiv preprint arXiv:2108.02084.
24. M.E. Andrés, N.E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Geoindistinguishability: Differential privacy for location-based systems, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 901–914.
25. Yuan S, Pi D, Zhao X, et al. Differential privacy trajectory data protection scheme based on R-tree[J]. Expert Systems with Applications, 2021, 182: 115215.
26. Li M, Zhu L, Zhang Z, et al. Achieving differential privacy of trajectory data publishing in participatory sensing[J]. Information Sciences, 2017, 400: 1-13.
27. F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, D. Jin, Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1241–1250.
28. K. Ouyang, R. Shokri, D.S. Rosenblum, W. Yang, A non-parametric generative model for human trajectories, in: IJCAI, Vol. 18, 2018, pp. 3812–3817.
29. J. Rao, S. Gao, Y. Kang, Q. Huang, LSTM-TrajGAN: A deep learning approach to trajectory privacy protection, 2020, arXiv preprint arXiv:2006.10521.
30. Wu, Z., Li, G., Shen, S., Lian, X., Chen, E., & Xu, G. "Constructing dummy query sequences to protect location privacy and query privacy in location-based services." World Wide Web 24.1 (2021): 25-49.
31. J. Tang, H. Zhu, R. Lu, X. Lin, H. Li and F. Wang, "DLP: Achieve customizable location privacy with deceptive dummy techniques in lbs applications." IEEE Internet of Things Journal 9.9 (2021): 6969-6984.
32. Pan, X., Nie, S., Hu, H., Yu, P., & Guo, J. "Reverse nearest neighbor search in semantic trajectories for Location based Services." IEEE Transactions on Services Computing (2020).
33. N. Nisha, I. Natgunanathan and Y. Xiang, "An Enhanced Location Scattering Based Privacy Protection Scheme," in IEEE Access, vol. 10, pp. 21250-21263, 2022.
34. Zhao P, Li J, Zeng F, et al. ILLIA: Enabling $ k $-anonymity-based privacy preserving against location injection attacks in continuous LBS queries[J]. IEEE Internet of Things Journal, 2018, 5(2): 1033-1042.
35. Tu Z, Zhao K, Xu F, et al. Protecting trajectory from semantic attack considering ${k} $-anonymity, ${l} $-diversity, and ${t} $-closeness[J]. IEEE Transactions on Network and Service Management, 2018, 16(1): 264-278.
36. Tan, Z., Wang, C., Yan, C., Zhou, M., & Jiang, C. "Protecting privacy of location-based services in road networks." IEEE Transactions on Intelligent Transportation Systems 22.10 (2020): 6435-6448.

1. ZINAN DING and XUEBIN MA are with the Department of Inner Mongolia Laboratory of Wireless Networking and Mobile Computing, Inner Mongolia University, Hohhot, 010000, China (e-mail: 32009139@mail.imu.edu.cn; csmaxuebin@imu.edu.cn) [↑](#footnote-ref-1)