FedSTDiTraj: Federated Spatio-Temporal Diffusion Model for Privacy-Preserving Trajectory Generation

Anonymous submission

Abstract

The utilization of trajectory data is crucial in Intelligent Transportation Systems (ITS), propelling significant advancements in various traffic-related tasks. Since some privacy-preserving methods, e.g. K-anonymity and Differential Privacy, can deal with GPS trajectories that contain personal geolocation information in an anonymized way, they may lead to suboptimal performance in downstream tasks and data utility loss when altering original data with perturbations. To address these problems, we propose a privacy-preserving structure named FedSTDiTraj for trajectory generation. This model harnesses the robust generative capabilities of diffusion models along with a graph-based embedding module for traffic trajectories in different road segment topologies. Additionally, we extend our model to the Federated Learning (FL) structure and design an elaborate resampling strategy to reduce the model size and protect privacy from common privacy attacks during FL. Experiments on two real-world datasets demonstrate that FedSTDiTraj can effectively generate high-fidelity trajectories while preserving the original distributions. Furthermore, the generated results can be utilized in traffic-related downstream tasks, significantly outperforming other methods in terms of utility.

Introduction

With the burgeoning prevalence of GPS-embedded devices, along with advancements in data collection and data mining technologies, an immense volume of trajectory data has been generated (Wang et al. 2020). This data is extensively employed across various traffic-related tasks due to its rich spatial-temporal information, such as travel time prediction (Zhu et al. 2022), origin-destination prediction (Shi et al. 2020) and other location-based services (Chekol and Fufa 2022). These applications are pivotal in fostering the adaptive development of Intelligent Transportation Systems (ITS). While these applications can significantly enhance our daily lives, the collection of trajectory data engenders substantial privacy concerns, as it contains sensitive information about users' daily routines and specific geographic locations (Jin et al. 2023). This data is vulnerable to misuse and unauthorized access without proper protection, potentially resulting in privacy breaches. Thus, it is imperative to prioritize the safeguarding of users' privacy when utilizing such data in downstream tasks.

Recently, there has been an increasing interest in developing privacy-preserving methods for GPS trajectory data. K-Anonymity (Sweeney 2002) and Differential Privacy (DP) (Dwork and Lei 2009) have been utilized to protect privacy in trajectory data. However, these methods often alter trajectory data, such as adding noise and increasing virtual trajectories, potentially introducing bias and compromising the integrity of the entire dataset, so as to degrade traffic-related downstream tasks. Therefore, to strike a balance between privacy preservation and data utility, trajectory generation methods have been proposed to replace the original data with synthesized data for downstream analysis. With the rapid development of deep learning techniques, many data generation methods have been proposed to generate high-quality data that closely resembles the original, maintaining the feature space and data distribution, including Generative Adversarial Network (GAN) (Rao et al. 2020), Variational AutoEncoder (VAE) (Xia et al. 2018; Jiang et al. 2023) and Diffusion Model (Zhu et al. 2023), which all ensure high-quality data generation and maintain the feature space and data distribution. However, the generation results of these models are not always high-quality, leading to post-processes like trajectory correction or map-matching. Besides, the centralized training approach still faces the risk of privacy breaches.

To address these challenges, we propose a privacy-preserving system that utilizes federated learning and diffusion models to generate high-quality trajectory data with privacy protection. In particular, we design a spatial-temporal diffusion model named **S**patial-**T**emporal **Di**ffusion **T**ransformer for T**raj**ectory Generation (STDiTraj) for trajectory generation. STDiTraj generates high-quality trajectory data to replace the original dataset for downstream tasks. A pretrained graph-based embedding module is applied to capture spatial features of traffic trajectories in different road segment topologies. Meanwhile, to ensure privacy preservation, a **Fed**erated Learning (FL) structure is applied and the entire system, FedSTDiTraj, will be trained within a privacy-preserving framework. Besides, to reduce communication costs and prevent gradient attacks during federated learning, a well-designed resampling strategy together with information compensation is integrated with STDiTraj. Experiments verify the utility of our model in two real-world datasets. The results illustrate that STDi-

Traj can effectively generate high-fidelity trajectories while preserving the privacy of data owners. The contributions of this paper are summarized as follows:

- We propose a FedSTDiTraj framework for trajectory generation with the consideration of the privacy-preserving scheme. The synthesized trajectory dataset can replace the original one without privacy concerns while keeping the original data distribution for further downstream tasks.
- We design a graph attention embedding module to capture spatial features within the dataset based on different topological road network structures, further enhancing the quality of trajectory generation. What's more, a novel resampling strategy is designed to protect privacy during the FL process.
- We validate FedSTDiTraj on two real-world datasets, which show superior performance in generating high-quality trajectory data than other baselines. What's more, experiments for the utility of downstream tasks are conducted to enrich the evaluation of each method.

Related Work

Trajectory generation

As spatial-temporal data with rich traffic information, trajectory data plays a crucial role in traffic data mining, guiding traffic tasks such as traffic forecasting. Generally, trajectory generation is employed for data augmentation, and here we utilize it to synthesize trajectory data for privacy-free usage, thereby protecting the private information contained in the original dataset. GAN (Demetriou et al. 2020; Rao et al. 2020), VAE (Jiang et al. 2023; Liu et al. 2022a) and diffusion model (Zhu et al. 2023, 2024; Wei et al. 2024) are widely used in traffic data generation tasks. In addition to these generative methods, non-generative methods offer privacy-free solutions. Traditional privacy-preserving theories like K-Anonymity (Sweeney 2002; Terrovitis and Mamoulis 2008; Chen et al. 2018) and Differential Privacy (Dwork and Lei 2009; Sun et al. 2023; Zhang et al. 2022) are applied to trajectory data to synthesize privacy-free dataset for further usages in downstream tasks. However, these methods only address privacy issues and overlook the bias introduced to the original dataset, resulting in utility loss. Therefore, we focus on generative methods and build up a privacy-preserving structure around them.

Diffusion model

As a state-of-the-art data generation method, the diffusion model demonstrates its powerful ability for data generation. It was proposed and developed in recent years (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2020). The main processes of the diffusion model are the forward process and the reverse process. The forward process incrementally adds noise to the original data, while the reverse process learns to recover the original data from the perturbed data. Additionally, several advancements have been made to improve generation speed and quality. The Denoising Diffusion Implicit Model (DDIM) (Song, Meng, and Ermon 2020) improves sampling speed through a non-Markovian diffusion process. Learning the variances of the reverse diffusion process speeds up the forward process with negligible differences in sample quality (Nichol and Dhariwal 2021). Several works have also focused on spatial-temporal generation based on the diffusion model (Zhu et al. 2023, 2024; Wei et al. 2024; Peebles and Xie 2023). Compared to these studies, we extend the diffusion transformer to traffic trajectory data and achieve high-quality traffic trajectory generation by capturing spatial features of traffic trajectory with a graph-based embedding module and integrating spatial-temporal features. Besides, since we apply the data synthesis method for privacy-preserving, federated learning and several enhancements on it are designed for the robust system.

Trajectory privacy preserving

Recently, various approaches have been developed to address the challenge of preserving location privacy in ITS. In a centralized architecture, a centralized entity is typically employed to protect location privacy. One of the most widely used methods in this context is K-anonymity (Sweeney 2002), which ensures that each data point is indistinguishable from at least $k-1$ other data points, effectively preventing the identification of unique users through queries (Terrovitis and Mamoulis 2008; Dai and Hua 2015). On the other hand, non-centralized architectures include obfuscation-based methods (Dwork and Lei 2009) and cryptographic-based methods (Liu et al. 2022b). However, centralized approaches and cryptographic methods often require a trusted third party, which can be challenging to establish. Besides, obfuscation-based methods can be vulnerable to background knowledge attacks (de Mattos, Domingues, and Loureiro 2019), where attackers use additional information to de-anonymize users. Generally, cryptographic and clustering-based approaches are often unsuitable for handling sparse and large datasets. Nevertheless, previous methods for preserving privacy in trajectory data have often overlooked the importance of semantic information, leading to a loss of dataset utility. To address these limitations, a trajectory generation-based method is proposed using the diffusion model, which can maintain the original feature space while generating new, synthetic data that preserves user privacy. Furthermore, FL is applied to the training process, ensuring that users' data is trained locally and not exposed to other devices.

Methodology

Before we propose our model, some preliminaries are presented to describe the problem and provide the basic definitions. We propose a novel privacy-preserving framework for traffic trajectory generation, leveraging diffusion models and FL. The details of our model design are shown below.

Preliminaries

Definition 1 (GPS Trajectory) *Let $\mathcal{P}$ denote a sequence of consecutively sampled GPS points, i.e., $\mathcal{P} = \{p_1, \ldots, p_m\}$, where $p_i = [lat_i, lon_i, t_i]$, $i \in \{1, \ldots, m\}$, represents the* *latitude and longitude of the GPS point at the time step t_i, indicating the device's (or the user's) location at that time.*

Definition 2 (Traffic road network) *The traffic road network is the base of the traffic system, which is defined as a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. $\mathcal{V}$ contains the intersections between the road segments and $\mathcal{E}$ contains all the road segments. Each road segment $e \in \mathcal{E}$ has several attributes, like road length and road type. Here we focus on the start point, end point and road length of the road segment.*

Definition 3 (Road-based Trajectory) *To constrain GPS points on traffic road segments, let $\mathcal{L}$ denote a sequence of consecutively map-matched GPS points, i.e., $\mathcal{R} = \{r_1, \ldots, r_m\}$, where $r_i = [e_i, pos_i, t_i]$, $i \in \{1, \ldots, m\}$, where e_i represents the corresponding mapped road segment ID and pos_i represents the concrete position on the mapped road for GPS point p_i. t_i indicates the device's (or the user's) location at that time. e_i is a discrete integer ranging from 0 to the number of road segments in a given traffic road topology and $e_i \in [0, 1]$.*

Problem 1 (Trajectory Generation) *Given a set of GPS trajectories $\mathcal{T}$, generate a new set of GPS trajectories $\mathcal{T}'$ that has similar data distribution to $\mathcal{T}$ but does not reveal the user's identity in the original dataset $\mathcal{T}$. During the process, the privacy of the original data must be guaranteed.*

Diffusion probabilistic model

The diffusion probabilistic model, or diffusion model for short, can generate high-quality images, text, and audio data with powerful data modeling capabilities and flexibility to handle a wide range of tasks including image generation, super-resolution, image restoration and editing, as well as natural language processing and multimodal learning, demonstrating superior capabilities to other models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Song et al. 2020). The core idea of the diffusion model consists of two processes: the forward process and the reverse process. In the forward process, random noise is incrementally added to the original data. During the reverse process, the model learns to recover the original data from the noised data. We can formulate these processes as follows.

Forward process: Given the original data X_0, Gaussian noises are added on X_0 through T steps. The forward process can be defined as a Markov chain:

$$\begin{aligned} q(X_{1:T}|X_0) &= \prod_{t=1}^{T} q(X_t|X_{t-1}), \\ q(X_t|X_{t-1}) &= \mathcal{N}(X_t; \sqrt{1-\beta_t}X_{t-1}, \beta_t\mathbf{I}), \end{aligned} \tag{1}$$

where $\mathbf{I}$ is Identity Matrix and $\beta_t \in (0,1)_{t=1}^{T}$ is corresponding variance. Since backward propagation can not be applied directly on samples from Gaussian distribution, reparameterization trick is applied(Ho, Jain, and Abbeel 2020), which represents X_t as $X_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$ and $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$.

Reverse process: During the reverse process, the model learns to recover the original data from noised data. The noised data is defined as $X_t \sim \mathcal{N}(0, \mathbf{I})$. The reverse process can also be defined as a Markov chain:

$$\begin{aligned} p_\theta(X_{0:T}) &= p(X_T)\prod_{t=1}^{T} p_\theta(X_{t-1}|X_t), \\ p_\theta(X_{t-1}|X_t) &= \mathcal{N}(X_{t-1}; \mu_\theta(X_t, t), \sigma_\theta(X_t, t)^2\mathbf{I}), \end{aligned} \tag{2}$$

where $\mu_\theta(X_t, t)$ and $\sigma_\theta(X_t, t)$ are mean and variance by θ respectively. Introduced by paper(Ho, Jain, and Abbeel 2020), for any $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t (t > 1)$ and $\tilde{\beta}_1 = \beta_1$, μ_θ and σ_θ are parameterized as:

$$\begin{aligned} \mu_\theta(X_t, t) &= \frac{1}{\sqrt{\alpha_t}}(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t)), \\ \sigma_\theta(X_t, t) &= \sqrt{\tilde{\beta}_t}. \end{aligned} \tag{3}$$

Diffusion Transformer

Since the diffusion model was initially proposed for image generation, the Diffusion Transformer (DiT) (Peebles and Xie 2023) is proposed built upon the Vision Transformer (ViT) (Dosovitskiy et al. 2021). A given image is first encoded by an encoder from the pretrained AutoEncoder and then delivered into the diffusion model for subsequent training. Due to the heavy computational requirements, a pretrained encoder E is used to compress the images into latent space, reducing the dimension of the input data X while maintaining its representational capabilities. Encoded images will be "patchified" into sequences for ViT structure. The diffusion model is trained on $z = E(X)$ while E remains frozen during training. Images are generated by sampling z' from the diffusion model and decoded with the pretrained decoder D. The synthesized image is $X' = D(z')$.

Model design

In this section, we will introduce our model in several parts: graph-based embedding module, FL framework, well-designed resampling strategy with information compensation, and the whole structure for trajectory generation.

Graph-based embedding To enhance the quality of trajectory generation, we focus on reducing unrealistic and meaningless trajectories, especially those that are not constrained to the road network. Traffic trajectory is not merely a type of time-series data, it is also a type of spatial-temporal data, which requires the consideration of spatial features. Given the road network $\mathcal{G}$, we first apply map-matching to the original GPS points. Each GPS point p_i is then mapped onto the road, and we represent it as $\boldsymbol{r_i} = [e_i, \mathrm{pos}_i, t_i]$ for the model. However, directly using discrete variables in training is challenging, necessitating processes like embedding.

To achieve a more suitable representation of the road segments, we develop a Graph Attention Network (GAT) (Veličković et al. 2017) for road-based trajectory representation learning. As shown in Figure 3, the GAT aggregates a node's features with those of its neighboring nodes through the self-attention mechanism. The initial features of each node $\boldsymbol{h_i}$ are defined by several attributes associated with the road segment, such as road type, road length, and the degree

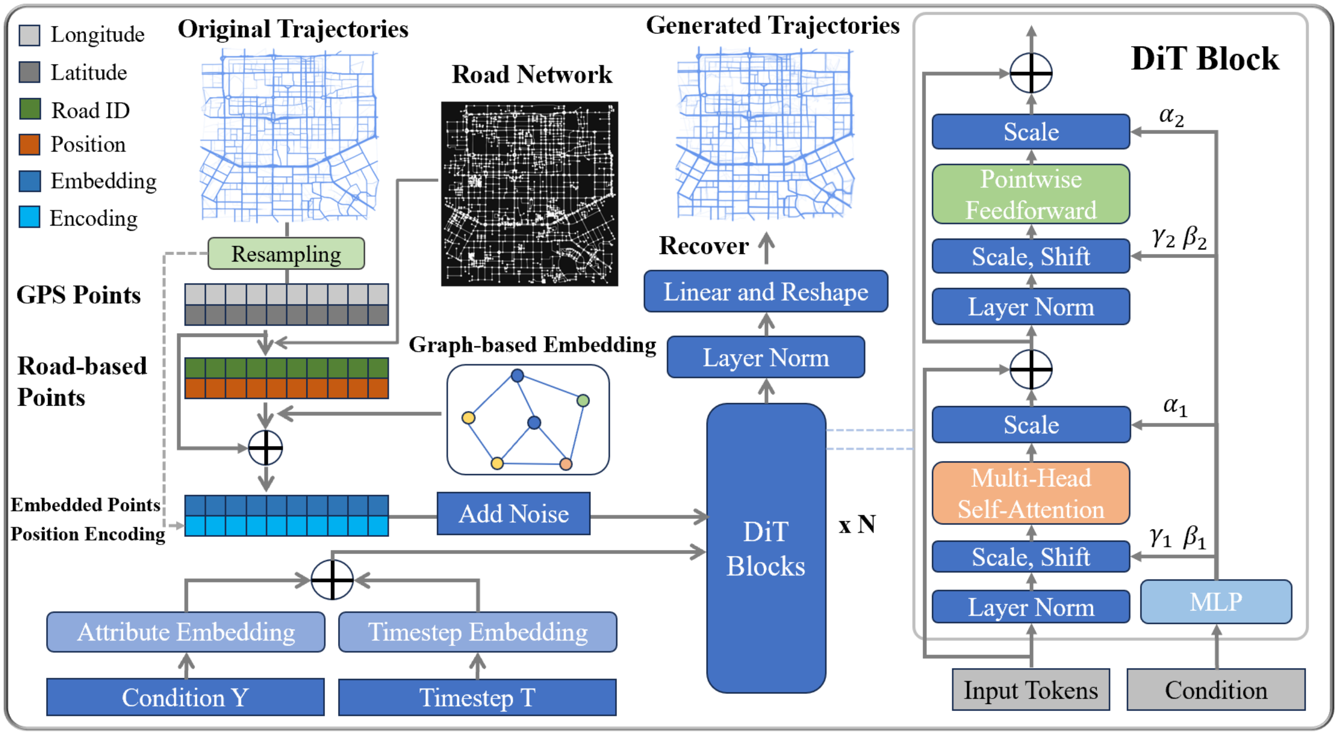

Figure 1: Network structure of STDiTraj.

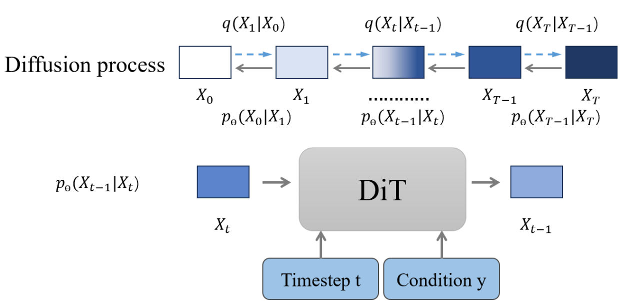

Figure 2: Workflow of the diffusion model.

of the road in the traffic topology graph. After processing with the GAT, the updated node features $\boldsymbol{h}_i'$ are obtained for trajectory embedding. The process can be illustrated as follows:

$$
\begin{aligned}
s_{ij} &= \text{LeakyReLU}(\mathbf{W_s}((\mathbf{W_i}h_i, \mathbf{W_j}h_j))), \\
\alpha_{ij} &= \frac{\exp(s_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(s_{ik})}, \\
h_i' &= \text{ELU}\left(\sum_{j\in\mathcal{N}_i}\mathbf{W_a}\alpha_{ij}h_j\right),
\end{aligned}
\tag{4}
$$

where $\mathbf{W_i}$, $\mathbf{W_j}$, $\mathbf{W_s}$ and $\mathbf{W_a}$ are learnable matrices, $\mathcal{N}_i$ is neighbor node set of node i. ELU is the Exponential Linear Unit activation function (Veličković et al. 2017).

To obtain a reliable pretrained embedding of road segments, we design a Masked Trajectory Recovery (MTR) task for pretraining. Inspired by Masked Language Models used for learning word representations (Devlin et al. 2018), we propose an MTR task to learn the representations of road segments by predicting masked locations within a trajectory. Given the time sensitivity inherent in trajectory representation, we also incorporate time embedding into the MTR task. This time embedding represents the minutes' index within a day (ranging from 0 to 288, with each index representing 5 minutes). Therefore, the input trajectory for the MTR task can be represented as follows:

$$
\begin{aligned}
\mathcal{R}' &= (GAT(h_i)\otimes\mathcal{R})\oplus pos_i, \\
M\mathcal{R}' &= (\mathcal{R}' + t_i + pe_i)\odot M,
\end{aligned}
\tag{5}
$$

where $M\mathcal{R}'$ is the masked trajectory, which is the input for pretraining, and $\boldsymbol{M}$ is a binary mask matrix with entity $\boldsymbol{M} \in \{0,1\}$. pos_i represents the detailed position of a trajectory point and is directly concatenated with road segments' embedding as $\mathcal{R}'$. Following this, a transformer-based model is employed to recover the masked locations within the trajectory. Through this recovery task, the model learns effective representations of the road segments.

Federated learning To ensure that personal data remains local, we apply the FL structure to uphold privacy-preserving model training. However, simply using FL is inadequate, as training on Non-Independent and Identically Distributed (Non-IID) data can result in performance degradation (Zhu et al. 2021). Drawing inspiration from prior

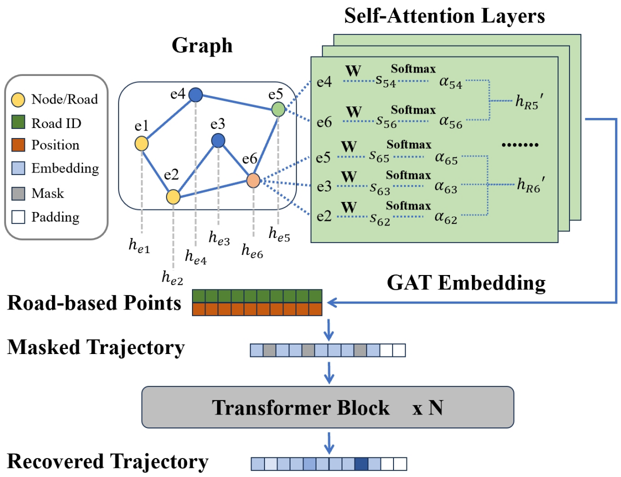

Figure 3: Graph-based embedding pretraining.

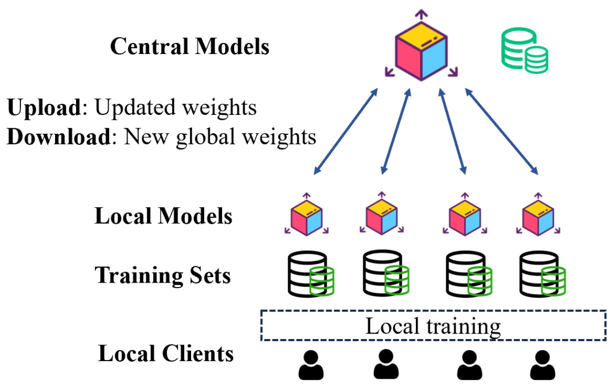

Figure 4: Federated learning structure.

work (Zhao et al. 2018), we design a data-sharing framework for FL to enhance training performance on Non-IID data. In this framework, a global trajectory set and $\boldsymbol{K}$ local trajectory sets are defined within the FL system to facilitate the data-sharing strategy, as illustrated in Figure 4.

We detail the definitions of our global and local sets. The global set comprises public trajectories that cannot be uniquely identified. A public trajectory is defined as a series of $\boldsymbol{r_i}$, where each $\boldsymbol{e_i}$ contains at least $\boldsymbol{g}$ trajectories at timestep $\boldsymbol{t_i}$, ensuring that the trajectory cannot be distinguished from at least $\boldsymbol{g}$ trajectories. In contrast, all other trajectories are classified as local trajectories. The parameter $\boldsymbol{g}$ serves as a hyperparameter that governs the privacy level of the entire system. A higher $\boldsymbol{g}$ value increases privacy but results in a smaller public trajectory set. Additionally, we exclusively use the public set for the road pretraining task.

Resampling strategy What's more, to decrease the model size which is crucial for reducing communication costs during federated learning, we propose a resampling strategy for input data. Considering the complexity of the transformer structure which is influenced by the maximum length of input sequences, we resample the trajectory to a fixed length $\boldsymbol{L}$ for trajectories of different lengths, which is implemented by linear interpolation.

Information compensation However, while resampling is beneficial, it introduces the risk of information loss, particularly for long-distance trajectories, where many location points may be omitted. Although the original trajectory length is provided to guide the generation model, it is insufficient for optimal performance. To address this, we propose an Information Compensation (IC) mechanism.

We reconstruct the positional encoding within the input sequence to ensure the model receives accurate positional information. Typically, the time difference between two trajectory points p_i and p_j will be changed when up-sampling or down-sampling, rendering the original positional encoding less effective in capturing the true relative temporal differences. Therefore, we design a new positional encoding tailored for resampled sequences. The updated positional encoding can be described as follows:

$$\begin{aligned} pos_L &= \frac{pos}{\text{scaling factor}}, \\ PE_{(pos_L, 2i)} &= \sin\left(\frac{pos_L}{10000^{2i/d_{\text{model}}}}\right), \\ PE_{(pos_L, 2i+1)} &= \cos\left(\frac{pos_L}{10000^{2i/d_{\text{model}}}}\right), \end{aligned} \tag{6}$$

where pos is the original position of the element in the sequence, i is the dimension index, and d_{model} is the dimension of the model. pos_L is new position lists under the fixed length $\boldsymbol{L}$. In this context, the scaling factor accounts for the degree of resampling applied, ensuring that the positional encoding continues to represent the correct time intervals between resampled trajectory points. This adjustment preserves the integrity of the temporal information despite changes in sequence length, thereby enhancing the model's ability to process the resampled data accurately.

Trajectory generation We detail the use of the Diffusion Transformer (DiT) for traffic trajectory generation, as illustrated in Figure 2. DiT has demonstrated superior performance in image generation tasks, particularly outperforming models with U-net structures. Due to its transformer-based architecture, DiT is particularly well-suited for handling time-series data generation, making it an excellent choice as the base model for synthesizing traffic trajectories.

Given a GPS trajectory set $\mathcal{D}_i$, we first resample it into a set $\mathcal{D}_i'$ with a fixed length $\boldsymbol{L}$ to standardize the input sequence length. To ensure that the spatial structure of the road network is effectively captured, we utilize a pretrained GAT for embedding road-based points. This embedding captures the spatial dependencies and relationships between the road segments, providing a rich context for trajectory generation. Road-based points are treated as auxiliary information that guides the generation process. Consequently, the input for DiT can be formulated as follows:

$$\boldsymbol{X_0} = \mathcal{P} + GAT(\mathcal{R}) + PE_L, \tag{7}$$

where $\mathcal{P}$ is GPS points, $GAT(\mathcal{L})$ is road-based points after graph-based embedding and PE_L is rejudged positional encoding under fixed length $\boldsymbol{L}$. The model takes it as input along with condition embeddings $\boldsymbol{Y}$ and timestep embeddings $\boldsymbol{T}$. Condition attributes, such as the total length of the trajectory, average speed, origin grid, and destination grid, guide the model in learning the desired trajectory generation patterns, ensuring that the generated trajectories align with the given spatial-temporal constraints.

Following the approach in DiT (Peebles and Xie 2023), diffusion models are trained on the reverse process defined by the distribution $p_\theta(X_{t-1}|X_t) = \mathcal{N}(\mu_\theta(X_t, t), \sigma_\theta(X_t, t))$. This is further extended using the variational lower bound of the log-likelihood of the initial trajectory X_0, expressed as:

$$\mathcal{L}(\theta) = -p(X_0|X_1) + \sum_t \mathcal{D}_{KL}(q^*(X_{t-1}|X_t, X_0) \| p_\theta(X_{t-1}|X_t)). \quad (8)$$

Here, q^* and p_θ are Gaussian distributions. By applying the reparameterization trick, the mean function μ_θ can be transformed into a noise prediction network ϵ_θ. The objective is to minimize the mean squared error (MSE) between the predicted noise $\epsilon_\theta(X_t)$ and the true noise ϵ_t sampled from a Gaussian distribution, as $\mathcal{L}_{mse} = \|\epsilon_t - \epsilon_\theta\left(X_t,\ t,\ Y\right)\|_2^2$ Additionally, the variance σ_θ can be trained using the full loss $\mathcal{L}$.

During sampling, we start with $X_T \sim \mathcal{N}(0, \mathbf{I})$ and recursively sample $X_{t-1} \sim p_\theta(X_{t-1}|X_t)$ using the reparameterization trick. This process generates high-quality, realistic traffic trajectories that are consistent with the underlying road network and temporal dynamics.

Experiments

We conduct several experiments on two real-world datasets to verify the performance of our proposed method. Due to space limitation, we provide detailed information in the **Appendix** and describe the basic results in this section.

Experimental settings

We conduct the experiments using PyTorch. The model is trained on four NVIDIA A100 40GB GPUs.

Datasets We evaluate our method against various baselines on two real-world datasets, which consist of daily taxi trajectories collected over a month in the cities of Chengdu and Xi'an.

Table 1: Statistics of the Real-world Trajectory Datasets.

Dataset	Chengdu	Xi'an
Trajectory Number	3 731 344	2 255 474
Average Time	13.51 min	16.11 min
Average Distance	3.56 km	3.49 km
Maximum Length	1878 Points	10 821 Points

Baselines The baselines we compare include both non-generative methods, such as Random Perturbation (RP) and Gaussian Perturbation (GP), and generative methods, including TrajGAN (Xi, Hanzhou, and Clio 2018), VAE (Xia et al. 2018), and diffusion models (Zhu et al. 2023). Detailed descriptions of these baselines are provided in **Appendix**.

Evaluation metrics We follow the methodology from previous work (Du et al. 2023) and use four evaluation metrics to assess the quality of generated trajectories across different models: **Density error**, **Trip error**, **Length error**, and **Pattern score**. These metrics are essential in evaluating trajectory generation. Specifically, the Jenson-Shannon divergence (JSD) is used to compare the distribution differences between the original and generated datasets. A lower JSD indicates higher similarity and, consequently, better generation quality. Density error measures the geographical distribution differences between the generated and original datasets. Trip error focuses on the geographical distribution between the origins and destinations of trajectories. Length error examines the distribution of trajectory lengths. Pattern score assesses the similarity of movement patterns. Detailed descriptions of these metrics are provided in the **Appendix**. We randomly sampled 9,000 trajectories from each model and calculated the metrics for them.

Hyperparameter Here we introduce several important hyperparameter settings. We discuss it further in **Appendix**. For FL, we set client number $\boldsymbol{K}$ as 100 to meet real life approximately. $\boldsymbol{g}$ is set as 5 to split the global set and local sets. The resampled length $\boldsymbol{L}$ is set as 200. The pretraining embedding dimension is set as 64.

Performance

As shown in Table 2, our method outperforms all other methods across two real-world datasets. These results demonstrate the effectiveness of combining a diffusion transformer structure with graph-based embedding and resampling with IC. Non-generative methods, such as RP and GP, directly perturb the original data without considering the underlying data distribution, resulting in lower-quality results. Generative methods like VAE and TrajGAN perform better but are still surpassed by diffusion-based methods. Among generative models, diffusion-based methods achieve the highest performance, with STDiTraj outperforming DiffTraj. The superior performance of STDiTraj is attributed to several factors shown as follows. The transformer structure effectively captures the temporal features of trajectories. Graph-based embedding provides essential traffic topology information, enhancing spatial dependency. IC mitigates the information loss associated with resampling, especially for trajectories of varying lengths. These components collectively contribute to our model's top performance. Additionally, the FL framework ensures successful model convergence with minimal performance loss. The global set guides the overall generation direction, while local sets supply detailed insights into the generation.

We also provide the visualization of the generated trajectories in Xi'an City. As shown in Figure 5, the generative methods can generate trajectories with a similar distribution as the original ones. Diffusion-based methods can generate trajectories with higher similarity. Compared to DiffTraj, our method reduces unrealistic and meaningless trajectories

Table 2: Performance comparison of different generative models.

Methods	Chengdu				Xi'an			
	Density (↓)	Trip (↓)	Length (↓)	Pattern (↑)	Density (↓)	Trip (↓)	Length (↓)	Pattern (↑)
RP	0.0690	0.0827	0.1647	0.3704	0.5198	0.4522	0.0634	0.2593
GP	0.0689	0.0833	0.1647	0.3519	0.6753	0.4458	0.0634	0.2963
VAE	0.0592	0.0793	0.1631	0.4814	0.0574	0.0468	0.0614	0.6667
TrajGAN	0.0445	0.0646	0.1574	0.5185	0.0520	0.0246	0.0591	0.7778
DiffTraj	0.0051	0.0134	0.0144	0.8519	0.0106	0.0190	0.0152	0.7678
FedSTDiTraj w/o *IC*	0.0020	0.0059	0.0083	0.8782	0.0055	0.0068	0.0129	0.8649
FedSTDiTraj w/o $\mathcal{G}$	0.0012	0.0034	0.0060	0.8825	0.0036	0.0049	0.0110	0.8674
FedSTDiTraj	**0.0009**	**0.0027**	**0.0051**	**0.8847**	**0.0030**	**0.0039**	**0.0101**	**0.8689**

Bold shows the best performance over all models. ↓: lower is better, ↑: higher is better.

(a) VAE (b) TrajGAN (c) DiffTraj (d) FedSTDiTraj (e) Real

Figure 5: Visualization of generated trajectory dataset in Xi'an City.

and constrains the trajectories on road segment topology, especially on sparse roads, further improving the generation quality. What's more, the visualization results illustrate the ability to understand spatial-temporal dynamics for FedSTDiTraj, leading to a more realistic generation.

Ablation study Ablation studies are conducted on the key components of our model to assess the importance of each part. Specifically, we evaluated FedSTDiTraj with the absence of either the graph-based embedding or the IC module. The results, presented in Table 2, underscore the significance of these modules since performance loss occurs due to the absence of these modules. Based on the diffusion transformer, IC provides the model with a comprehensive view of resampled trajectories, leading to more precise generation results. What's more, graph-based embedding is crucial for ensuring the model performs well in sparse areas by introducing road constraints.

Table 3: Utility test on traffic flow prediction task (Xi'an). Performance is shown as (original / generated).

Methods	ASTGCN	GWNet	DCRNN
RMSE	4.62 / 4.60	5.93 / 5.74	4.89 / 4.72
MAE	2.91 / 2.90	3.59 / 3.47	3.07 / 3.01

Utility test Our proposed framework aims to generate data that is not only privacy-preserving but also useful for downstream tasks, similar to the original data. To verify the utility, we chose the traffic flow prediction task (Jiang et al. 2021), an important task in Intelligent Transportation Systems (ITS). We trained prediction models, like ASTGCN (Guo et al. 2019), GWNet (Wu et al. 2019), and DCRNN (Li et al. 2017), using the original data and tested them on both the original and generated datasets. The results, shown in Tables 3, demonstrate that the generated data from FedSTDiTraj performs almost identically to the original data, thereby validating its utility in downstream tasks. Thus, the generated trajectories can replace the original ones for further traffic tasks in a privacy-free theme with the same utility.

Conclusion

In this work, we propose a privacy-preserving trajectory generation framework named FedSTDiTraj that leverages the diffusion model and federated learning. The framework incorporates the graph-based embedding module and the resampling strategy with the information compensation mechanism to enhance data generation quality. Extensive experiments validate the effectiveness of FedSTDiTraj, demonstrating both high-quality generation and the preservation of data utility in downstream tasks. For future work, we plan to extend this framework to other types of spatial-temporal data to address a wider range of tasks. This expansion will further validate the versatility and robustness of FedSTDiTraj in various applications.

References

Chekol, A. G.; and Fufa, M. S. 2022. A survey on next location prediction techniques, applications, and challenges. *EURASIP Journal on Wireless Communications and Networking*, 2022(1): 29.

Chen, Z.; Fu, Y.; Zhang, M.; Zhang, Z.; and Li, H. 2018. A Flexible Mix-Zone Selection Scheme Towards Trajectory Privacy Protection. In *Proc. IEEE TrustCom/BigDataSE*, 1180–1186.

Dai, J.; and Hua, L. 2015. A Method for the Trajectory Privacy Protection Based on the Segmented Fake Trajectory under Road Networks. In *2015 2nd International Conference on Information Science and Control Engineering*, 13–17.

de Mattos, E. P.; Domingues, A. C.; and Loureiro, A. A. 2019. Give Me Two Points and I'll Tell You Who You Are. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 1081–1087.

Demetriou, A.; Allsvåg, H.; Rahrovani, S.; and Chehreghani, M. H. 2020. Generation of Driving Scenario Trajectories with Generative Adversarial Networks. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Du, Y.; Hu, Y.; Zhang, Z.; Fang, Z.; Chen, L.; Zheng, B.; and Gao, Y. 2023. Ldptrace: Locally differentially private trajectory synthesis. *Proceedings of the VLDB Endowment*, 16(8): 1897–1909.

Dwork, C.; and Lei, J. 2009. Differential Privacy and Robust Statistics. In *Proc. ACM Symposium on Theory of Computing*, 371–380.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 922–929.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Jiang, R.; Yin, D.; Wang, Z.; Wang, Y.; Deng, J.; Liu, H.; Cai, Z.; Deng, J.; Song, X.; and Shibasaki, R. 2021. Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 4515–4525.

Jiang, Y.; Wu, Y.; Zhang, S.; and Yu, J. J. 2023. FedVAE: Trajectory privacy preserving based on Federated Variational AutoEncoder. In *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 1–7.

Jin, F.; Hua, W.; Francia, M.; Chao, P.; Orlowska, M. E.; and Zhou, X. 2023. A Survey and Experimental Study on Privacy-Preserving Trajectory Data Publishing. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5577–5596.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Liu, C.; Antypenko, R.; Sushko, I.; and Zakharchenko, O. 2022a. Intrusion Detection System After Data Augmentation Schemes Based on the VAE and CVAE. *IEEE Transactions on Reliability*, 71(2): 1000–1010.

Liu, M.; Cheng, L.; Gu, Y.; Wang, Y.; Liu, Q.; and O'Connor, N. E. 2022b. MPC-CSAS: Multi-Party Computation for Real-Time Privacy-Preserving Speed Advisory Systems. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 5887–5893.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Rao, J.; Gao, S.; Kang, Y.; and Huang, Q. 2020. LSTM-TrajGAN: A Deep Learning Approach to Trajectory Privacy Protection. In Janowicz, K.; and Verstegen, J. A., eds., *11th International Conference on Geographic Information Science (GIScience 2021) - Part I*, volume 177 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 12:1–12:17. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISBN 978-3-95977-166-5.

Shi, H.; Yao, Q.; Guo, Q.; Li, Y.; Zhang, L.; Ye, J.; Li, Y.; and Liu, Y. 2020. Predicting Origin-Destination Flow via Multi-Perspective Graph Convolutional Network. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 1818–1821.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sun, X.; Ye, Q.; Hu, H.; Wang, Y.; Huang, K.; Wo, T.; and Xu, J. 2023. Synthesizing Realistic Trajectory Data With Differential Privacy. *IEEE Transactions on Intelligent Transportation Systems*, 1–14.

Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05): 557–570.

Terrovitis, M.; and Mamoulis, N. 2008. Privacy Preservation in the Publication of Trajectories. In *The Ninth International Conference on Mobile Data Management*, 65–72.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, S.; Bao, Z.; Culpepper, J. S.; and Cong, G. 2020. A Survey on Trajectory Data Management, Analytics, and Learning. *arXiv preprint arXiv:2003.11547*.

Wei, T.; Lin, Y.; Guo, S.; Lin, Y.; Huang, Y.; Xiang, C.; Bai, Y.; Ya, M.; and Wan, H. 2024. Diff-rntraj: A structure-aware diffusion model for road network-constrained trajectory generation. *arXiv preprint arXiv:2402.07369*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*.

Xi, L.; Hanzhou, C.; and Clio, A. 2018. trajGANs: using generative adversarial networks for geo-privacy protection of trajectory data. *Vision paper*.

Xia, T.; Song, X.; Fan, Z.; Kanasugi, H.; Chen, Q.; Jiang, R.; and Shibasaki, R. 2018. DeepRailway: A Deep Learning System for Forecasting Railway Traffic. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 51–56.

Zhang, J.; Huang, Q.; Huang, Y.; Ding, Q.; and Tsai, P.-W. 2022. DP-TrajGAN: A privacy-aware trajectory generation model with differential privacy. *Future Generation Computer Systems*.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Zhu, H.; Xu, J.; Liu, S.; and Jin, Y. 2021. Federated learning on non-IID data: A survey. *Neurocomputing*, 465: 371–390.

Zhu, Y.; Ye, Y.; Liu, Y.; and James, J. 2022. Cross-Area Travel Time Uncertainty Estimation From Trajectory Data: A Federated Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, 23(12): 24966–24978.

Zhu, Y.; Ye, Y.; Zhang, S.; Zhao, X.; and Yu, J. 2023. Diff-Traj: Generating GPS Trajectory with Diffusion Probabilistic Model. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 65168–65188. Curran Associates, Inc.

Zhu, Y.; Yu, J. J.; Zhao, X.; Liu, Q.; Ye, Y.; Chen, W.; Zhang, Z.; Wei, X.; and Liang, Y. 2024. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. *arXiv preprint arXiv:2404.15380*.

Appendix

A Experiments details

In this section, we introduce all settings in experiments, including baseline description and hyperparameter settings.

A.1 Baseline

We introduce the baseline and our method in detail.

- **Random Perturbation (RP)**: Random noise is added to the original GPS points. The noise scale is $[-0.01, 0.01]$ to control the perturbed distance.
- **Gaussian Perturbation (GP)**: Similar to RP, Gaussian noise with mean 0 and variance 0.005 is added on the original GPS points.
- **VAE** (Xia et al. 2018): A VAE with four convolutional layers and two linear layers is built up for trajectory generation. The trajectories are encoded and decoded by VAE. The well-trained decoder is used for generation.
- **TrajGAN** (Xi, Hanzhou, and Clio 2018): A GAN with four convolutional layers and two linear layers is built up for trajectory generation. Adversarial training is applied and the generator tries to generate fake trajectories from random noise that are similar to the real ones. The discriminator tries to distinguish them from real samples. The well-trained generator is used for generation.
- **DiffTraj** (Zhu et al. 2023): A diffusion model with convolutional layers and U-net structure is designed to generate high-quality traffic trajectories.
- **FedSTDiTraj w/o *IC***: FedSTDiTraj without IC. It is trained to verify the influence of IC.
- **FedSTDiTraj w/o $\mathcal{G}$**: FedSTDiTraj without graph-based embedding. It is trained directly on GPS points and aims to verify the influence of road-based constraints.
- **STDiTraj**: The basic model of FedSTDiTraj. It is trained in a centralized theme without FL.

A.2 Hyperparameters

For FedSTDiTraj, we provide the pretraining settings in Table 4 and the hyperparameters of FedSTDiTraj in Table 5. Also, the reference range of hyperparameters based on the experience is given.

Table 4: Hyperparameters for Graph-based embedding.

Hyperparameter	Setting value	Refer range
Attention layers number	5	$2 \sim 10$
Transformer block number	10	$2 \sim 15$
Mask ratio	0.4	$0 \sim 0.6$
Heads number	8	≥ 1
Embedding dimension	64	$32 \sim 128$

Table 5: Hyperparameters for FedSTDiTraj.

Parameter	Setting value	Refer range
Diffusion Steps	500	$300 \sim 500$
Skip steps	5	$1 \sim 10$
Hidden dimension	128	≥ 64
β (linear schedule)	$0.0001 \sim 0.05$	–
Batch size	128	≥ 64
Resampled Length	200	$120 \sim 400$
Client number	100	$5 \sim 500$
g-Anonymity	5	$1 \sim 10$

A.3 Implementation

We trained the model on two taxi trajectory datasets from Chengdu and Xi'an cities[1]. We remove trajectories with lengths less than 120 and resample the left trajectories into fixed lengths of 200. The client number is set as 100 to ensure the acceptable distributed training theme.

The condition guidance for generation is defined as a list of attributes from the original trajectory, including departure time, total distance, travel time, total length, average distance, average speed, origin grid, and destination grid. These attributes guide the model to generate the desired one.

A.4 Metrics

For evaluation, we apply grid-based statistics and divide the city into grids of 16x16 size for distribution counting.

- **Density error**: Density error evaluate the geographic distribution between the original dataset $\mathcal{D}$ and the generated dataset $\mathcal{D}'$. The evaluation is conducted based on discrete grids which are divided on the city.
- **Trip error**: Trip error also evaluates the geo-distribution on the grid map. It focuses on the origin grids and destination grids of the trajectories, which also verify the guidance of the condition variables.
- **Length error**: Length error measures the distribution of trajectory distances. It is calculated by computing the distribution difference of geo-distances between consecutive points in original trajectories and generated trajectories.
- **Pattern score**: Pattern score aims to find top-n grids that occur most frequently. A higher pattern score means more of the same top-n in the generated dataset as the original one and higher distribution similarity. n is set to 25.

B Supplementary experiments

In this section, we introduce further experiments and discussion on diffusion-based models and discover their efficiency.

B.1 Additional performance experiment

Due to space limitations, we also test our model on the Porto dataset [2]. Since Chengdu and Xi'an may be hard to access,

[1]https://outreach.didichuxing.com/

[2]https://www.kaggle.com/datasets/crailtap/taxi-trajectory/

Table 6: Performance comparison of different settings.

Methods	Chengdu				Xi'an			
	Density (↓)	Trip (↓)	Length (↓)	Pattern (↑)	Density (↓)	Trip (↓)	Length (↓)	Pattern (↑)
FedSTDiTraj	0.0009	0.0027	0.0051	0.8847	0.0030	0.0039	0.0101	0.8689
STDiTraj w/o *IC*	0.0013	0.0047	0.0076	0.8794	0.0037	0.0059	0.0108	0.8660
STDiTraj w/o $\mathcal{G}$	0.0008	0.0029	0.0044	0.8869	0.0023	0.0044	0.0091	0.8702
STDiTraj	**0.0006**	**0.0022**	**0.0044**	**0.8869**	**0.0014**	**0.0031**	**0.0069**	**0.8717**

Bold shows the best performance over all models. ↓: lower is better, ↑: higher is better.

Table 7: Performance comparison of different generative models on Porto dataset.

Methods	Porto			
	Density (↓)	Trip (↓)	Length (↓)	Pattern (↑)
RP	0.1944	0.2571	0.0528	0.3036
GP	0.1872	0.2603	0.0527	0.3037
VAE	0.0527	0.0311	0.0570	0.5106
TrajGAN	0.0436	0.0261	0.0483	0.6667
DiffTraj	0.0072	0.0119	0.0215	0.7940
FedSTDiTraj	**0.0015**	**0.0032**	**0.0087**	**0.8711**

Bold shows the best performance over all models. ↓: lower is better, ↑: higher is better.

we conducted additional experiments on the Porto dataset and the results also show the superior ability of our model as illustrated in Table 7.

B.2 Utility results

According to Table 8, we also test the utility of generated trajectories in Chengdu. The code for the utility test is publicly available[3].

Table 8: Utility test on traffic flow prediction task (Chengdu). Performance is shown as (original / generated).

Methods	ASTGCN	DCRNN	GWNet
RMSE	6.53 / 6.52	8.37 / 8.20	7.29 / 7.18
MAE	3.75 / 3.58	4.97 / 4.95	4.70 / 4.66

B.3 Case study

Here we select the generated trajectories and discover the generation quality of them. Generally, we focus on the guidance accuracy for diffusion-based methods and samples 9,000 trajectories in Xi'an city.

Evaluation To evaluate the generation quality from attributes guidance, we focus on **Total distance**, **Average speed**, **Origin grid**, and **Destination grid** to check whether the generated trajectories satisfy these attributes. We use accuracy for **Origin grid** and **Destination grid** and mean absolute error for **Total distance** and **Average speed** to check the generation quality.

As shown in Table 9, our method achieves better performance as it generates trajectories more similar to the given attributes. Higher accuracy shows more accurate generation according to the origin and destination grids. The lower mean absolute error means more precise context in the generated trajectories. It not only accounts for better performance in main experiments but also shows the superior ability of reliable and controllable generation.

Table 9: Generation case study.↓: lower is better, ↑: higher is better.

Methods	**DiffTraj**	**FedSTDiTraj**
Origin grid (↑)	91.59%	94.24%
Destination grid (↑)	91.41%	94.69%
Total distance (↓)	$0.1301\,\mathrm{km}$	$0.0826\,\mathrm{km}$
Average speed (↓)	$0.2179\,\mathrm{m/s}$	$0.1384\,\mathrm{m/s}$

B.4 Ablation study

We also further explore the centralized training theme on STDiTraj and the performance is shown in Table 6. The performance illustrates that FL still influences the final results of the model. Setting global and local sets reduces the influence of the Non-IID problem but can not eliminate it.

[3]https://github.com/deepkashiwa20/DL-Traff-Graph