Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

SemTraj: Semantic-controllable diffusion model for high-fidelity trajectory data generation

Yuchen Jiang [a], Guanhua Chen [b], Shiyao Zhang [c], Liang Han [d], James Jianqiao Yu [d,*]

[a] Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, 518055, China
[b] Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, 518055, China
[c] School of Advanced Engineering, Great Bay University, Dongguan, 523000, China
[d] Computer Science and Technology, Harbin Institute of Technology, Shenzhen, 518055, China

ARTICLE INFO

Keywords:
Trajectory generation
Diffusion transformer
Conditional generation

ABSTRACT

Trajectory data consists of sequential GPS points recording geographical movement behaviors, constituting fundamental resources for urban analytics. However, publicly accessible trajectory datasets remain scarce or privacy-encumbered. Prevailing generative methodologies often fail to adhere to trajectory-specific constraints or necessitate granular segment-level annotations to guide the generative process. Herein, we present SemTraj, a novel diffusion-Transformer framework engineered to synthesize voluminous, high-fidelity, and diverse trajectories conforming to trajectory semantic constraints such as origin and destination. Central to SemTraj is a Trajectory Denoising Transformer that integrates a novel Semantic-Adaptive Layer Normalization for fine-grained conditioning, alongside an adaptive resampling strategy that adjusts position encoding while preserving temporal fidelity and optimizing computational efficiency. We rigorously evaluate SemTraj on three diverse real-world datasets, demonstrating its superior fidelity and controllability compared to existing methods.

1. Introduction

Trajectories, timestamped sequences of GPS points tracing entity movements, have become indispensable for urban analytics and facilitate diverse applications ranging from travel-time estimation (Zhu et al., 2022) and origin-destination (OD) flow analysis (Shi et al., 2020) to public-safety operations like criminal path reconstruction and shared-bicycle deployment strategy (Chekol & Fufa, 2022). To develop and thoroughly evaluate models that capture the full range of spatiotemporal dynamics, researchers need large, high-quality trajectory datasets that reflect diverse mobility patterns. However, creating such datasets is often challenging due to strict privacy regulations, high data collection costs, and restrictions on data sharing (Jiang et al., 2021a,b). As a result of this data scarcity, many studies have focused on trajectory modeling and synthetic data generation. Yet real-world trajectories vary widely due to differences in individual behavior and contextual factors such as road network layouts and traffic conditions. This variability makes accurate modeling and realistic data synthesis particularly difficult (Chen et al., 2022; Li et al., 2021).

Trajectory synthesis has emerged as a promising approach, using generative frameworks (e.g., GANs Rao et al., 2020; Xi et al., 2018, VAEs Xia et al., 2018, diffusion models Wei et al., 2024; Zhu et al., 2023) to generate synthetic trajectories that closely resemble real-world distributions. Attribute-driven trajectory generation focuses on using specific, domain-relevant attributes, such as origin-destination pairs and departure times, to produce trajectories that match these attributes. However, most existing methods rely heavily on detailed input specifications, including ordered road segment identifiers, velocity profiles, or full waypoint sequences, to guide the generation process (Feng et al., 2020; Wang et al., 2021). In practice, such detailed information is rarely available in advance. For instance, criminal investigators often only have access to OD pairs when analyzing suspect movements, and bike-sharing services typically base deployment decisions based on OD demand patterns rather than precise routing. As a result, there remains a significant gap between the trajectory-defining attributes required by prior models and the limited but realistic data available for real-world applications.

In this paper, we present SemTraj, a trajectory generation model based on diffusion that relaxes the requirement for excessive trajectory attributes. It enables high-fidelity generation by conditioning solely on OD pairs and departure times, as shown in Fig. 1. To introduce conditional guidance effectively, we propose a semantic-adaptive layer normalization mechanism. By unleashing the dependency on prior road-segment information, SemTraj can generate realistic trajectories even when intermediate waypoints or speed profiles are unavailable.

* Corresponding author.
E-mail addresses: 12232418@mail.sustech.edu.cn (Y. Jiang), chengh3@sustech.edu.cn (G. Chen), zhangshiyao@gbu.edu.cn (S. Zhang), han.liang@hit.edu.cn (L. Han), jqyu@hit.edu.cn (J.J. Yu).

https://doi.org/10.1016/j.eswa.2026.131661
Received 30 August 2025; Received in revised form 19 January 2026; Accepted 10 February 2026
Available online 15 February 2026

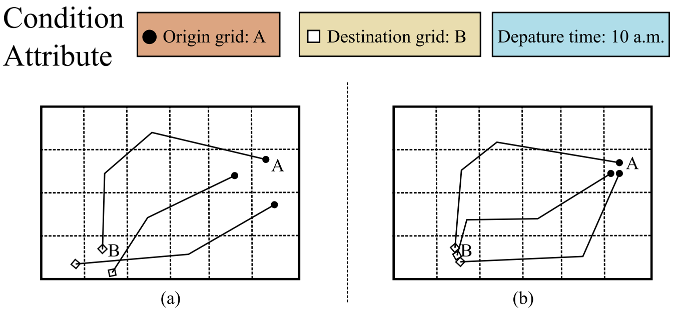


Fig. 1. (a) Generate trajectories but do not match the semantic attributes. (b) High-fidelity and semantic-controllable generation.

Additionally, an adaptive resampling module further enhances performance by dynamically adjusting sequence position encodings. This allows for faster training and inference while preserving the fidelity of the generated trajectories.

Our main contributions are as follows:

- We propose SemTraj that synthesizes trajectories based on given semantic attributes (origin-destination and departure time). The generated data follows the distribution of real-world trajectories without requiring road segment information and other commonly used prior but practically unavailable information, like average speed and total distance.
- We develop an adaptive resampling strategy to align position encoding with resampled trajectories, reducing information loss and accelerating generation. During inference, a lightweight length generator predicts trajectory length from semantic attributes to guide positional encoding and restore temporal resolution after resampling.
- We empirically validate SemTraj utilizing three real-world datasets. The results demonstrate superior performance in generating high-fidelity trajectory data with respect to the baseline state of the art.

2. Related work

In this section, we provide a brief overview of previous research on generative models and conditional generation techniques, with their potential applications for mobility analysis. Our discussion emphasizes diffusion models, owing to their outstanding performance when compared to other contemporary generative modeling paradigms.

2.1. Diffusion model

As a cutting-edge data generation technique, diffusion models have demonstrated strong capabilities in producing high-quality data (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2020b). The diffusion model operates through two main processes: the forward process, where noise is gradually added to the original data, and the reverse process, where the model learns to recover the original data from the noisy input. Several advancements have been introduced to improve both the speed and quality of generation. For instance, the Denoising Diffusion Implicit Model (DDIM) (Song et al., 2020a) accelerates sampling using a non-Markovian process. Learning variance schedules in the reverse process has also been shown to speed up generation with minimal loss in sample quality (Nichol & Dhariwal, 2021). Diffusion models have also been applied to spatiotemporal data. For example, DiffTraj (Zhu et al., 2023) employs the diffusion model combined with a U-Net architecture to generate trajectory data. ControlTraj (Zhu et al., 2024) extends this approach by adding topology constraints to improve data quality, while Diff-RNTraj (Wei et al., 2024) introduces a pretraining module focused on road segment representations. More recently, Transformer-based diffusion frameworks have leveraged self-attention mechanisms to model long-range dependencies, enabling scalable and high-fidelity data synthesis across a variety of domains (Peebles & Xie, 2023).

2.2. Conditional generation

Conditional data generation has attracted increasing attention due to its ability to incorporate external information and produce controlled outputs. Two common approaches are Conditional Generative Adversarial Networks (CGANs) (Mirza, 2014) and Conditional Variational Autoencoders (CVAEs) (Lee et al., 2017). These models extend their original forms (GANs and VAEs) by accepting auxiliary inputs (e.g., class labels or attributes) to guide the generation process toward desired outcomes. Conditional generation is particularly useful for addressing class imbalance and data scarcity by generating more samples for underrepresented categories. For example, Das et al. (2022) combines a conditional normalizing flow with a downstream classifier, using feedback from the classifier to improve the quality of synthetic data related to pandemics. In conditional diffusion models, guidance is typically introduced by either appending attribute embeddings to intermediate representations (Zhu et al., 2023) or integrating attribute tokens via cross-attention layers (Peebles & Xie, 2023; Rombach et al., 2022). These techniques improve both controllability and the fidelity of the generated samples (Cao et al., 2024), allowing the model to effectively follow the semantic attributes and generate the intended outputs.

2.3. Applications in mobility analysis

The generation of high-fidelity, controllable trajectory data is not an isolated goal but is deeply motivated by and contributes to several key areas in mobility research. In Mobile Crowd Sensing and Computing (MCSC), which leverages participatory sensing and mobile social data for large-scale, cross-space sensing (Guo et al., 2015), generating realistic mobility traces is vital for simulating human-in-the-loop systems and testing novel paradigms. Similarly, research in spatial context inference directly utilizes mobility patterns to enrich applications. For instance, clustering WiFi access point visitation patterns can extract spatial context to significantly improve the performance of bandit-based recommendation systems (Gutowski et al., 2019), while check-in trajectories from Location-Based Social Networks (LBSNS) can be modeled to discover local geographic topics for personalized recommendations (Long et al., 2012). Our semantically controllable generation provides a valuable tool for creating simulated mobility data to stress-test such context-aware algorithms under diverse and specific conditions.

Furthermore, trajectory generation is fundamentally connected to large-scale human mobility prediction. A core task in this domain is learning powerful representations from trajectory data (Gutowski et al., 2017), often achieved through methods like trajectory embeddings to identify and model human mobility patterns (Gao et al., 2017). The field of pedestrian trajectory prediction, critical for autonomous systems and robotics, continuously evolves with new models that require extensive, diverse data for training and validation, as noted in recent comprehensive surveys (Zaier et al., 2025). By producing variable-length, semantically anchored trajectories, our work offers a scalable data synthesis foundation that can support the development and benchmarking of next-generation prediction and embedding models across these interconnected areas.

3. Preliminaries

In this section, we define the problem, introduce core concepts, and review the diffusion probabilistic model.

3.1. Problem definition

Definition 1 (GPS Trajectory)**.** Formally, a GPS trajectory $\mathcal{P}$ is defined as a temporally ordered sequence of GPS points $\mathcal{P} = \{p_1, \dots, p_m\}$. Each point $p_i \in \mathcal{P}$ is represented as a triplet $p_i = \langle \text{lon}_i, \text{lat}_i, t_i \rangle$, where lon_i and lat_i denote the longitude and latitude, respectively, at time step t_i. The triplet p_i thus encapsulates the spatial coordinates and the timestamp for the ith point in the trajectory.

Definition 2 (Semantic Attribute)**.** Semantic attribute c is a compact representation of contextual information used to steer trajectory synthesis. We specify $c = (O, D, t_{\text{dep}})$, where O is the origin location (start point), D is the destination location (end point), and t_{dep} is the departure time. During conditional generation, c is embedded into the model to guide the sampling process such that generated trajectories begin at O, terminate at D, and reflect temporal patterns associated with t_{dep}.

Problem 1 (Trajectory Generation)**.** Given a dataset of real-world GPS trajectories $\mathcal{T} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n\}$, where each $\mathcal{P}_i = \{p_1^i, \dots, p_m^i\}$ constitutes an individual trajectory, and the corresponding semantic attribute set of N trajectories $\mathbf{C} = \{c_k\}_N$. The objective is to train a generative model $\mathbf{G}$ which is tasked with generating a set of synthetic trajectories $\tilde{\mathcal{T}} = \mathbf{G}(\mathbf{C})$. The generated trajectories are required to exhibit a data distribution statistically similar to that of $\mathcal{T}$. Moreover, the generation process must be effectively controlled by the provided semantic attribute set $\mathbf{C}$.

3.2. Diffusion probabilistic model

In practice, the training process of the diffusion model can be summarized as learning the Gaussian noise ϵ_θ and minimizing the mean squared error (MSE) between the predicted noise $\epsilon_\theta(X_t)$ and the true noise ϵ_t, which is sampled from a Gaussian distribution:

$$\min_\theta \mathcal{L}(\theta) = \min_\theta \mathrm{E}_{t, X_0 \sim q} \left\| \epsilon_t - \epsilon_\theta\left(X_t,\ t\right) \right\|_2^2, \tag{1}$$

where ϵ_t is the true noise for time step t.

Diffusion models have emerged as powerful generative frameworks across domains such as image, text, and audio synthesis (Ho et al., 2020; Saharia et al., 2022; Song et al., 2020a). Compared to GANs and VAEs, they offer greater training stability and generation quality.

A diffusion model defines a forward process that gradually adds noise to data, and a reverse process that learns to denoise and reconstruct the original sample.

3.2.1. Forward process

Given original data denoted as X_0, the forward diffusion process is characterized by the progressive addition of Gaussian noise over T discrete steps. This process is formally structured as a Markov chain, defined as:

$$q(X_{1:T}|X_0) = \prod_{t=1}^{T} q(X_t|X_{t-1}), \tag{2}$$

$$q(X_t|X_{t-1}) = \mathcal{N}(X_t; \sqrt{1-\beta_t} X_{t-1}, \beta_t \mathbf{I}), \tag{3}$$

where $\mathbf{I}$ represents the identity matrix and $\beta_t \in (0,1)_{t=1}^T$ is a sequence of variances. To enable differentiability and facilitate gradient-based optimization, the reparameterization trick is typically employed (Ho et al., 2020). This technique allows for the expression of X_t in terms of X_0 and a noise term, specifically, $X_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$ and $\bar{\alpha}_t = \prod_{i=1}^t (1-\beta_i)$.

3.2.2. Reverse process

In the reverse process, the generative model is trained to approximate the reverse Markov chain and reconstruct the original data distribution from pure noise. The initial state of the reverse process is given by sampling from a standard Gaussian distribution, $X_t \sim \mathcal{N}(0, \mathbf{I})$. Analogous to the forward process, the reverse process is also formulated as a Markov chain:

$$p_\theta(X_{0:T}) = p(X_T) \prod_{t=1}^{T} p_\theta(X_{t-1}|X_t), \tag{4}$$

$$p_\theta(X_{t-1}|X_t) = \mathcal{N}(X_{t-1}; \mu_\theta(X_t, t), \sigma_\theta(X_t, t)^2 \mathbf{I}), \tag{5}$$

where $\mu_\theta(X_t, t)$ and $\sigma_\theta(X_t, t)$ are the mean and variance functions, respectively, parameterized by θ. Following the reparameterization technique (Ho et al., 2020), for any $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ $(t > 1)$ and $\tilde{\beta}_1 = \beta_1$, the parameters μ_θ and σ_θ are specified as:

$$\mu_\theta(X_t, t) = \frac{1}{\sqrt{\alpha_t}}(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t)), \tag{6}$$

$$\sigma_\theta(X_t, t) = \sqrt{\tilde{\beta}_t}. \tag{7}$$

4. SemTraj framework

In this section, we introduce the proposed model, SemTraj, for generating high-fidelity and semantic-controllable trajectories. We propose an efficient framework for traffic trajectory generation. The following details outline the design of SemTraj.

4.1. SemTraj overview

The SemTraj framework, as illustrated in Fig. 2, is designed to generate high-fidelity, semantically-controllable trajectories by processing raw inputs through a structured pipeline. Specifically, SemTraj aims to learn a conditional generative model $p_\theta(\mathbf{X}_0 \mid c)$, where $\mathbf{X}_0 \in \mathbb{R}^{L\times 2}$ denotes the target trajectory and c represents the semantic condition. During training, raw trajectories are first adaptively resampled and perturbed with random noise, conditioned on semantic attributes such as origin, destination, and departure time. These noisy sequences are then iteratively refined by a Trajectory Denoising Transformer, which learns a conditional denoising function $\mathcal{D}_\theta(\mathbf{X}_t, c, t)$ to recover the underlying clean trajectory distribution. In parallel, a conditional VAE-based length generator $p_\phi(L \mid c)$ is trained to model the distribution of trajectory lengths based on semantic inputs. While this component does not influence training-time resampling, it plays a key role during inference: since generation starts from random noise, the predicted length $\hat{L}$ is used to determine position encodings and resolution via adaptive resampling. By combining iterative denoising with semantic conditioning, SemTraj produces realistic trajectories that adhere to both spatial and temporal constraints, making it well-suited for downstream applications.

4.2. TDFormer block

SemTraj is designed as a denoising framework that aims to develop a neural network capable of accurately estimating and removing the noise component $\epsilon_\theta(X_t, t)$ at each diffusion timestep t, transforming pure noise into meaningful trajectories. To achieve this, we introduce the Trajectory Denoising Transformer (TDFormer) block, a novel architecture that combines Semantic-Adaptive Layer Normalization (SALN) with Multi-Head Self-Attention (MHSA). This design enables precise modeling of trajectory-specific attributes and captures the underlying spatiotemporal patterns. The SALN module introduces conditional guidance by dynamically adjusting normalization parameters based on the trajectory's semantic context. Meanwhile, the MHSA mechanism effectively captures internal dependencies within the trajectory sequence, allowing the model to learn meaningful spatial and temporal relationships. Specifically, MHSA focuses on modeling pairwise interactions among trajectory points to capture fine-grained spatial and temporal correlations, while SALN modulates feature statistics at the layer level, allowing global semantic conditions to consistently influence the denoising process across all layers. TDFormer parameterizes the conditional denoising operator by jointly capturing local geometric variations and long-range trajectory dependencies, which are both essential for reconstructing movement patterns from noise.

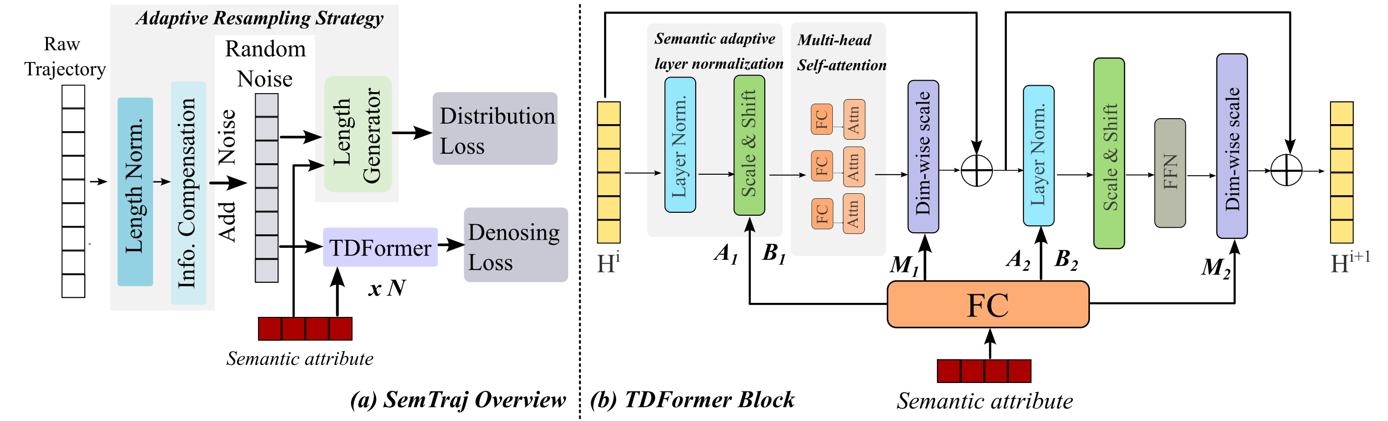


Fig. 2. Network structure of SemTraj. (a) SemTraj Overview. (b) TDFormer Block.

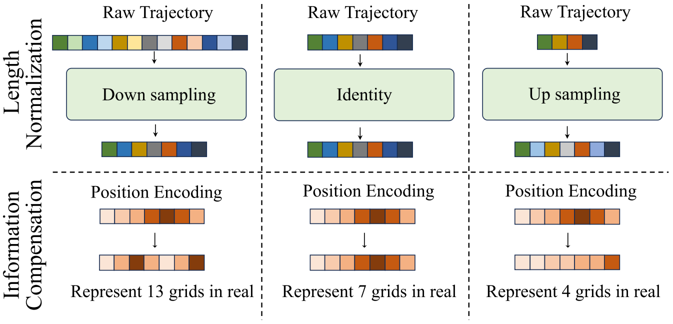


Fig. 3. Adaptive resampling Strategy. (a) Length Normalization. (b) Information Compensation.

4.2.1. Semantic adaptive layer normalization

The SALN module operates through a scale-and-shift mechanism. It uses learnable scale ($\boldsymbol{A}$) and shift ($\boldsymbol{B}$) parameters, which are conditioned on both the diffusion timestep t and the semantic attribute c. In addition, we introduce dimension-wise scaling parameters ($\boldsymbol{M}$), which are applied just before the residual connections in each TDFormer block. This mechanism effectively works as a semantic-aware gating strategy, allowing the network to adaptively regulate the contribution of attention and feed-forward transformations under different semantic and temporal conditions. These parameters control the influence of each module's output, enabling dynamic adjustment of its contribution to the final representation. Moreover, zero-initialization of the final batch normalization scale factor $\boldsymbol{a}$ in each TDFormer block is employed to expedite large-scale training procedures, particularly in supervised learning scenarios. Zero-initialization strategy ensures that each TDFormer block initially behaves as an identity mapping, which stabilizes early-stage diffusion training where noise levels are high and prevents premature overfitting to semantic signals.

The SALN is defined as follows:

$$\mathrm{SALN}(\boldsymbol{H}^i) = A^i \odot \frac{\boldsymbol{H}^i - \mathrm{E}[\boldsymbol{H}^i]}{\sqrt{\mathrm{Var}(\boldsymbol{H}^i) + \epsilon}} + B^i \tag{8}$$

$$\boldsymbol{A}^i = \boldsymbol{W}_a^i \cdot \boldsymbol{c}, \quad \boldsymbol{B}^i = \boldsymbol{W}_b^i \cdot \boldsymbol{c}, \quad \boldsymbol{M}^i = \boldsymbol{W}_m^i \cdot \boldsymbol{c}, \tag{9}$$

where $\boldsymbol{H}^i$ is the input of ith TDFormer block. $\mathrm{E}[\boldsymbol{H}^i]$ and $\mathrm{Var}(\boldsymbol{H}^i)$ represent the mean and variance calculated per channel respectively. ϵ is a small constant used for training stability. $\boldsymbol{M}^i$ is dimension-wise scale factor, $\boldsymbol{A}^i$ and $\boldsymbol{B}^i$ are scale and shift factors respectively. The computational flow of these parameter interactions is formally described by the equations below:

$$\boldsymbol{h}^{i+1} = \boldsymbol{M}_1^i \cdot \mathrm{MHSA}(\mathrm{SALN}(\boldsymbol{H}^i)) + \boldsymbol{H}^i, \tag{10}$$

$$\boldsymbol{H}^{i+1} = \boldsymbol{M}_2^i \cdot \mathrm{FFN}(\mathrm{SALN}(\boldsymbol{h}^{i+1})) + \boldsymbol{h}^{i+1}, \tag{11}$$

where $\boldsymbol{H}^{i+1}$ is the output of ith TDFormer block and FFN denotes Feed Forwad Network, as illustrated in Fig. 2.

This layer-wise processing enables the model to progressively refine its trajectory representations, at each step capturing richer spatio-temporal dependencies while integrating the external semantic attribute. The refinement is particularly well-suited to the diffusion setting, as early layers focus on recovering coarse global structure, while deeper layers refine local route variations under semantic guidance.

4.3. Adaptive resampling strategy

As depicted in Fig. 3, the adaptive resampling module comprises three key components: length normalization, information compensation, and length generator. Adaptive resampling establishes an explicit mapping between variable-length real trajectories and fixed-length latent representations, enabling the efficient diffusion process while preserving temporal fidelity.

4.3.1. Length normalization

Real-world trajectories vary in length, but Transformer models require fixed-length inputs for computational efficiency. We resample each map-matched trajectory $\hat{\mathcal{P}}$ to a fixed length L via linear interpolation as previous studies do Zhu et al. (2023, 2024):

$$\hat{\mathcal{P}}' = \mathrm{AR}(\hat{\mathcal{P}}, L), \tag{12}$$

where $\hat{\mathcal{P}}'$ is the resampled trajectory. The length L is a predefined hyperparameter, typically set based on dataset statistics (e.g., the average or maximum trajectory length) during preprocessing. The normalization maps the original continuous trajectory domain onto a normalized space of fixed length L, thereby standardizing the input resolution for subsequent attention-based modeling.

4.3.2. Information compensation

While adaptive resampling effectively reduces computational overhead, it inherently introduces the potential for information loss, especially in long-distance trajectories where intermediate points are omitted. To mitigate this, we employ an Information Compensation (IC) mechanism by updating positional encodings. We adjust the positional encoding to reflect the resampled trajectory and maintain temporal accuracy. The updated positional encoding is calculated as:

$$PE_{(pos_L,2i)} = \sin\left(\frac{L}{l_0} \cdot \frac{pos}{10000^{2i/d_{\mathrm{model}}}}\right), \tag{13}$$

$$PE_{(pos_L,2i+1)} = \cos\left(\frac{L}{l_0} \cdot \frac{pos}{10000^{2i/d_{\mathrm{model}}}}\right), \tag{14}$$

where l_0 is the original trajectory length, pos is the original position, and $pos_L = pos * l_0/L$ is the adjusted position for the fixed length L. The scaling factor $\frac{l_0}{L}$ explicitly restores the relative temporal intervals of the original trajectory within the fixed-length representation, ensuring positional encodings accurately represent time intervals in the resampled trajectory. IC preserves temporal information and improves model processing of resampled data. Besides, IC is critical for attention layers, as it allows attention weights to reflect true temporal proximity between trajectory points, even after length normalization.

4.3.3. *Length generator*

To enable adaptive resampling during inference where ground-truth trajectory lengths are unavailable, we introduce a lightweight conditional variational autoencoder (CVAE)-based Length Generator. Formally, the length generator models the conditional distribution $p(l_0 \mid O, D, t_{\text{dep}})$, enabling length-aware trajectory synthesis without access to ground-truth durations at inference time. Given the semantic attribute (O, D, t_{dep}), it predicts the real trajectory length l_0 to guide both resampling and positional encoding. The CVAE consists of a 4-layer MLP encoder and decoder. During training, it learns the conditional distribution of real-world trajectory lengths. Although not used for resampling during training (as actual lengths are known), the predicted l_0 is crucial at inference time: (1) it defines the resolution for resampling positional encodings from random noise inputs, and (2) it is used in post-processing to restore the model's fixed-length output (of length l_0) to a realistic temporal length. This setup balances the need for variable-length trajectory generation with efficient training, made possible by fixing the generated sequence length to l_0.

Overall, the adaptive resampling strategy decouples sequence length modeling from spatial trajectory generation, enabling scalable diffusion training while preserving realistic temporal structure.

4.4. *Trajectory generation*

Based on TDFormer, trajectory-defining attributes c, and the adaptive resampling strategy, SemTraj is outlined with the following training and generating processes:

4.4.1. *Training*

The goal of training is to predict the noise at a given diffusion time step t and semantic attribute c. The optimization objective is to minimize the mean square error between real noise and predicted noise, represented as:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \mathrm{E}_{c,t,X_0 \sim q} \left\| \epsilon_t - \epsilon_\theta\left(X_t,\ t,\ c\right) \right\|_2^2, \tag{15}$$

4.4.2. *Generation*

During sampling, we start with $X_T \sim \mathcal{N}(0, \mathbf{I})$, a Gaussian noise initialization. The trajectory is then recursively sampled via the learned reverse process: $X_{t-1} \sim p_\theta(X_{t-1}|X_t)$, modeling the transition from X_t to X_{t-1}. The reparameterization trick ensures that the model can generate samples by backpropagating through the noise addition process.

4.4.3. *Noise scheduling*

We adopt the linear noise schedule β_t which increases uniformly from β_1 to β_T. Its deterministic and monotonic nature provides a stable and predictable denoising trajectory. The key to balancing global consistency and local variation lies not solely in the schedule but in the capacity of TDFormer and the SALN. During the early reverse steps, the input is dominated by noise, and the model must rely heavily on the strong semantic guidance from SALN to predict the overall direction from origin to destination, establishing global consistency. As denoising progresses and noise decreases, the input signal becomes clearer, allowing the TDFormer's self-attention mechanism to focus on recovering finer local variations and the nuanced correlations between consecutive points, all while remaining anchored by the persistent semantic conditioning. Thus, the linear schedule provides a structured noise reduction path, and our network architecture is explicitly designed to leverage different stages of this path to capture both macro and micro patterns.

Table 1
Statistics of the real-world trajectory datasets.

Dataset	Chengdu	Xi'an	Porto
Trajectory Number	3 731 344	2 255 474	1 414 164
Average Time	13.51 min	16.11 min	12.19 min
Average Distance	3.56 km	3.49 km	3.96 km
Sampling Interval	3 s	3 s	15 s

5. Experiments

In this section, we first describe the experimental settings, including datasets, baselines, evaluation metrics, and hyperparameters. Then, we conduct comprehensive experiments on real-world datasets to evaluate the performance of the proposed method, SemTraj, and address the following research questions:

- **RQ1**: Does the trajectory data generated by SemTraj demonstrate superior fidelity while maintaining the data distribution compared to state-of-the-art methods?
- **RQ2**: Can SemTraj be effectively controlled by the given semantic attributes?
- **RQ3**: How does SemTraj ensure geographically feasible trajectory?
- **RQ4**: How does the adaptive resampling strategy influence the balance between computational efficiency and temporal fidelity in trajectory generation?
- **RQ5**: How does each module in SemTraj contribute to the overall generation performance?
- **RQ6**: How does SemTraj perform with model scaling and dataset scaling?

5.1. *Experimental settings*

5.1.1. *Datasets*

We evaluate SemTraj with various state-of-the-art baselines on three real-world datasets, which consist of daily taxi trajectories collected over a month in the cities of Chengdu, Xi'an,[1] and Porto,[2] encapsulating diverse urban mobility dynamics. Table 1 presents the statistical overview of these datasets. All data is de-identified before use.

5.1.2. *Baselines*

We compare the proposed SemTraj with state-of-the-art generative methods described as follows:

- **CVAE** (Ding et al., 2020): A conditional variational autoencoder consisting of four convolutional layers and two linear layers, trained with semantic attributes. The decoder generates new trajectory samples based on learned representations.
- **CGAN** (Mirza, 2014): A conditional GAN architecture with four convolutional and two linear layers. It uses semantic attributes during training, where the generator produces synthetic samples and the discriminator distinguishes real from fake. The trained generator is used for trajectory generation.
- **DiffWave** (Kong et al., 2020): A Wavenet-based diffusion model that uses 16 residual blocks with bi-directional dilated convolutions and 1D CNNs. It generates sequences using sigmoid and tanh activation functions.

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

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

Table 2
Performance comparison of different generative models.

Dataset	Metrics	CVAE	CGAN	DiffWave	MDM	DiffTraj	Diff-RNTraj	ControlTraj	SemTraj
Chengdu	Density (↓)	0.0583	0.0442	0.0136	0.0046	0.0051	0.0371	0.0031	**0.0023**
	Length (↓)	0.1630	0.1566	0.0311	0.0125	0.0144	0.1078	0.0097	**0.0061**
	Pattern (↑)	0.5001	0.5219	0.7590	0.8437	0.8519	0.6094	0.8547	**0.8770**
	OD Acc. (↑)	47.11%	49.83%	–	92.81%	92.60%	–	93.76%	**94.83%**
Xi'an	Density (↓)	0.0569	0.0516	0.0208	0.0087	0.0106	0.0093	0.0070	**0.0035**
	Length (↓)	0.0607	0.0582	0.0313	0.0142	0.0152	0.0191	0.0134	**0.0083**
	Pattern (↑)	0.6790	0.7815	0.5920	0.7730	0.7678	0.7940	0.8330	**0.8611**
	OD Acc. (↑)	55.46%	56.89%	–	91.60%	91.50%	–	93.42%	**94.24%**
Porto	Density (↓)	0.0525	0.0435	0.0096	0.0060	0.0072	0.0052	0.0049	**0.0037**
	Length (↓)	0.0560	0.0479	0.0243	0.0156	0.0215	0.0156	0.0144	**0.0118**
	Pattern (↑)	0.5194	0.6774	0.8150	0.8259	0.7940	0.8220	0.8319	**0.8519**
	OD Acc. (↑)	50.03%	51.12%	–	91.93%	90.25%	–	92.49%	**94.07%**

Bold shows the best performance, and underline shows the second-best. ↓: lower is better, ↑: higher is better.

- **MDM** (Tevet et al., 2023): Motion Diffusion Model (MDM) is a Transformer-based diffusion denoising model that replaces convolutional backbones with a temporal Transformer encoder. It can be naturally adapted from human motion to traffic trajectory generation.
- **DiffTraj** (Zhu et al., 2023): A diffusion model employing a U-Net architecture with convolutional layers to generate high-quality trajectories. Semantic attributes are used during both training and inference to guide the generation process.
- **Diff-RNTraj** (Wei et al., 2024): A diffusion model that incorporates a pre-trained Road-Constraint-Trajectory module to convert GPS points into road network-constrained representations, ensuring that generated trajectories follow real-world road structures.
- **ControlTraj** (Zhu et al., 2024): This model uses a pretrained Masked Autoencoder (MAE) to encode road segment topology, enhancing trajectory fidelity. The encoded information is used throughout both training and inference.

5.1.3. Evaluation metrics

To evaluate how well the generated trajectories match the distribution of real trajectories, we conduct a rigorous assessment of their statistical similarity. We employ Jensen-Shannon Divergence (JSD) as a primary measure of trajectory quality. JSD quantifies the difference between the distributions of real and synthetic data, with lower values indicating closer alignment to the real distribution.

We employ the following metrics, which are widely used in previous studies, to evaluate performance.

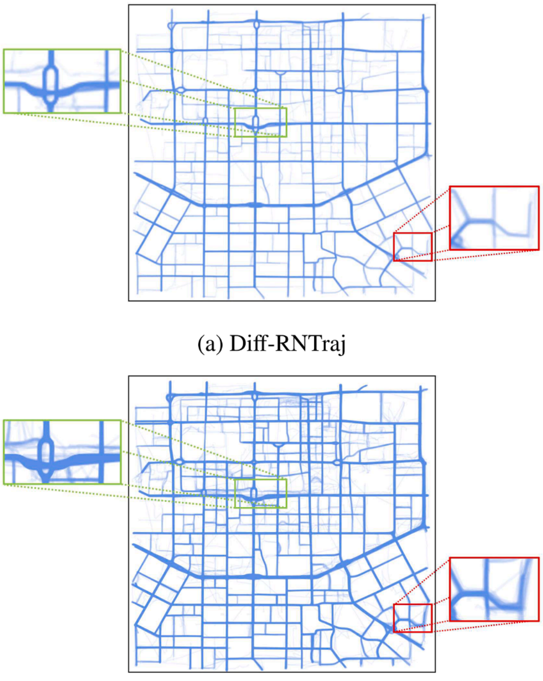
- **Density error**: Measures the geographic distribution difference between the original dataset $\mathcal{D}$ and the generated dataset $\mathcal{D}'$, computed point by point along each trajectory, shortened as "Density".
- **Length error**: Evaluates the difference in trajectory lengths by comparing the distribution of geographic distances between consecutive points in the real and generated trajectories, shortened as "Length".
- **Pattern score**: Captures how well the most frequently visited regions are preserved. Specifically, it computes the overlap between the top-n most visited grids in the generated and real datasets (with n set to 25). A higher score indicates better alignment in spatial usage patterns, shortened as "Pattern".
- **OD accuracy**: Assesses the controllability of the generation process by measuring how accurately the generated trajectories reflect the intended origin and destination. This metric is only applied to models that support conditional generation. As such, DiffWave and Diff-RNTraj, which are unconditional models, are excluded from this evaluation for fairness, shortened as "OD Acc.".

5.2. Overall performance (RQ1)

SemTraj outperforms all baseline methods across the three evaluated datasets, as shown in Table 2. In Chengdu, it reduces density and length errors by 25.81% and 37.11%, respectively, and improves the pattern score by 15.35%. In Porto, it achieves 24.49% and 18.06% reductions in density and length errors, along with an 11.90% gain in pattern score. These results underscore the effectiveness of the TDFormer block, which surpasses the U-Net architecture used in prior models. While Diff-RNTraj imposes road network constraints, it struggles in scenarios with high road segment complexity. Notably, we include MDM, a Transformer-based diffusion model adapted from human motion generation. While its performance surpasses that of U-Net-based models (e.g., DiffTraj), benefiting from the global receptive field of self-attention for capturing long-range dependencies, it still falls short of SemTraj and ControlTraj. It indicates that the naive Transformer backbone is insufficient for trajectory generation. The lack of a tailored mechanism for fine-grained spatiotemporal conditioning limits its ability to strictly adhere to semantic constraints and geographical realism.

The performance differences across datasets highlight the impact of road network complexity. In cities like Chengdu, with dense and intricate road layouts, SemTraj's ability to model complex, high-dimensional spatial patterns proves particularly valuable. In contrast, for simpler road networks such as those in Porto, the improvements are still present but relatively smaller. The superior performance of SemTraj, particularly its low length error and high pattern score, empirically validates the effectiveness of the proposed architecture. The Multi-Head Self-Attention mechanism inherently models long-range spatial relationships, allowing the model to reason about the overall route structure between the origin and destination. Concurrently, the Semantic-Adaptive Layer Normalization, conditioned on the trajectory factors, ensures that the temporal dynamics (e.g., travel speed variation by time of day) are involved in the generative process at every denoising step. This dual mechanism enables an effective learning of the data manifold compared to convolutional U-Net baselines, which have a more localized receptive field and less flexible conditioning mechanisms.

As shown in Fig. 4, we compare three diffusion-based methods for trajectory generation. All produce realistic patterns in high-density regions (green boxes). Among them, SemTraj demonstrates the highest generation quality. Compared to Diff-RNTraj and ControlTraj, it significantly reduces the occurrence of unrealistic or implausible trajectories, especially in low-density or sparse areas (highlighted in red boxes). It also exhibits stronger alignment with the underlying road network, even without explicitly relying on road-segment information. While Diff-RNTraj produces road-constrained trajectories by design, its output distribution diverges notably from the original data, resulting in lower overall fidelity. In contrast, the visualizations clearly show that Sem-



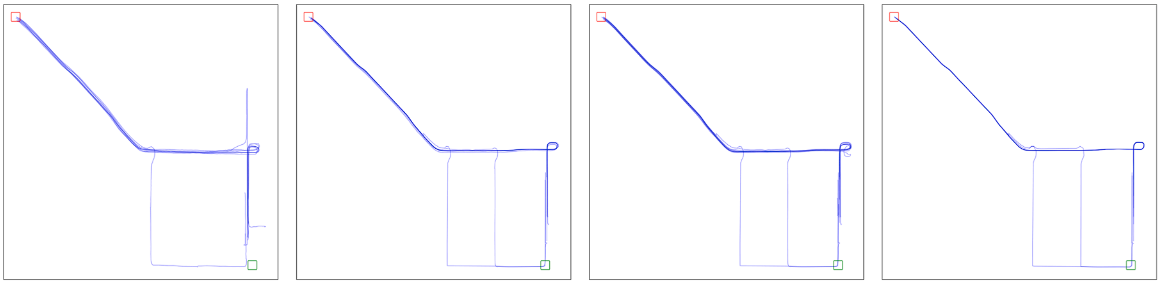
(a) Diff-RNTraj

(b) ControlTraj

(c) SemTraj

(d) Real

Fig. 4. Visualization of the generated trajectory dataset in Xi’an City.



(a) DiffTraj

(b) ControlTraj

(c) SemTraj

(d) Real

Fig. 5. Controllable case study in Xi’an City.

Table 3
Controllable generation study in Xi’an City.

Methods	CVAE	CGAN	DiffTraj	ControlTraj	SemTraj
Origin grid accuracy (↑)	55.30%	55.76%	91.59%	93.22%	**94.24%**
Destination grid accuracy(↑)	55.62%	58.03%	91.41%	93.62%	**94.69%**
Total distance error (↓)	1227 m	1203 m	130.1 m	86.10 m	**82.60 m**
Average speed error (↓)	3.480 m/s	3.368 m/s	0.2179 m/s	0.1590 m/s	**0.1384 m/s**
Fidelity error (↓)	844 m	809 m	356 m	130 m	**107 m**

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

Traj captures the spatial-temporal dynamics more effectively, leading to more realistic and semantically consistent trajectory generation.

Overall, these results demonstrate that SemTraj is especially effective in complex urban environments, yet remains consistently strong across all settings. Its capacity to capture essential trajectory patterns makes it a robust solution for trajectory data generation.

5.3. Controllable generation (RQ2)

The results in Table 2, along with the visualization in Fig. 5, validate SemTraj’s superior ability to follow semantic constraints. We specifically examine OD pairs between grid 7 (upper-left red box) and grid 100 (lower-right green box). The figure shows that DiffTraj fails to reproduce the true trajectory distribution. While ControlTraj maintains high fidelity, it depends on additional road-segment information, limiting its applicability in scenarios where such prior knowledge is unavailable. SemTraj’s advantage lies in its SALN module, which introduces learnable scale-and-shift parameters derived from the origin and destination embeddings into each denoising step. This mechanism consistently guides the diffusion process toward the intended OD pair. In contrast, CVAE and CGAN do not incorporate semantic conditioning beyond their initial input layers. As a result, their OD accuracy hovers around 50%,

Table 4
Efficiency study on SemTraj.

Resampling length	Chengdu					Xi'an				
	100	200	300	400	500	100	200	300	400	500
Length (↓)	0.0129	0.0073	0.0061	0.0059	0.0058	0.0138	0.0099	0.0083	0.0082	0.0081
Inference time	98	105	113	143	189	105	110	121	156	217

↓: lower is better. Inference time is represented as seconds/batch.

Table 5
Ablation study on SemTraj.

Metrics	Xi'an	
	Density (↓)	Pattern (↑)
SemTraj w/o AR	0.0044	0.8540
SemTraj w/o c	0.0103	0.8191
SemTraj-Cross Attention	0.0071	0.8322
SemTraj-In-Context	0.0056	0.8460
SemTraj-SALN	0.0038	0.8573
SemTraj	**0.0035**	**0.8611**

Bold shows the best performance and underline shows the second-best. ↓: lower is better, ↑: higher is better.

highlighting their limited ability to enforce semantic constraints during generation.

We also focus on the accuracy of guidance for these methods by sampling 9000 trajectories in Xi'an City with five condition-driven methods. To evaluate how well the generated trajectories align with the specified attributes, we focus on the origin grid, destination grid, total distance, and average speed to determine if the generated trajectories meet these conditions. Besides, we also examine the fidelity of the generated trajectories. The fidelity error measures the average distance difference between the generated trajectory data and the map-matched generation results, which use a post-processing technique for ground truth. Geo-distances between raw points and map-matched points serve as the criterion for this calculation. The results in Table 3 further validate the superior controllability of SemTraj in traffic trajectory generation. SemTraj achieves the highest accuracy for both the origin grid accuracy and the destination grid accuracy and the lowest errors for both total distance error and average speed error, surpassing all other models. This indicates that SemTraj can effectively adhere to conditional constraints, generating trajectories that start and end at the desired locations with remarkable precision. The results also show the reason for the superiority of our model in terms of generation quality. In contrast, traditional models like CVAE and CGAN perform significantly worse, demonstrating their limitations in understanding condition guidance. Notably, while DiffTraj and ControlTraj exhibit competitive performance in certain metrics, they fall short of SemTraj's overall balance. TDFormer with adaTLN shows its superior performance in aligning generation with condition guidance. This underscores SemTraj's robustness and its advantage in generating high-fidelity, controllable trajectories without compromising realism.

5.4. *Geographical feasibility (RQ3)*

A critical aspect of trajectory generation is adherence to real-world road networks and urban boundaries. Although SemTraj does not explicitly ingest road-segment data, our visual results in Fig. 5 demonstrate its strong implicit geographical plausibility. Particularly in the sparse regions highlighted by red boxes, SemTraj generates trajectories that closely follow the underlying road network structure and maintain realistic spatial distributions, avoiding the implausible shortcuts or cross-block artifacts seen in some baseline outputs. This emergent property stems from two key factors: high-fidelity distribution learning and architectural design for context preservation. First, as quantitatively validated in Table 2, diffusion-based models, including SemTraj, significantly outperform GAN and VAE counterparts in metrics like density and pattern error. This superior capacity to model the true data distribution means that when trained on clean, real-world trajectories, which inherently obey geographical constraints, the model internalizes these constraints. Second, TDFormer, empowered by SALN and adaptive resampling, effectively captures the complex spatio-temporal dependencies present in real mobility data. The semantic conditioning on origin and destination further anchors the global path in a realistic urban context. Therefore, by learning from and faithfully reconstructing the data manifold of real trajectories, SemTraj naturally generates geographically feasible paths.

5.5. *Model efficiency (RQ4)*

We evaluate different resampling lengths of the Adaptive Resampling module to examine the balance between computational efficiency and temporal fidelity. The Chengdu and Xi'an datasets are selected for the assessment since both share the same sampling interval of 3 s, ensuring consistent temporal patterns after resampling. We test SemTraj using resampling lengths of 100, 200, 300, 400, 500. The results are presented in Table 4. The Length Error reflects how well the synthesized trajectories preserve real-world temporal scales and movement patterns, serving as our primary indicator of temporal fidelity. Inference Time, measured in seconds per batch of 512 samples, captures the computational cost in a practical deployment setting. As shown in Table 4, increasing the resampling length reduces length error but increases inference time, which is expected given the quadratic complexity of the Transformer's self-attention mechanism with respect to sequence length.

While extending the length from 100 to 300 yields significant fidelity gains, further increases to 400 or 500 bring only marginal error reduction at a higher computational cost. Hence, a resampling length of 300 is identified as the optimal operating point, offering an excellent balance where substantial temporal fidelity is achieved before the computational cost rises sharply. This optimal value of 300 also aligns with the average original trajectory lengths in the datasets (270 for Chengdu, 322 for Xi'an). This correlation further supports our empirical finding, suggesting that setting the fixed resampling length close to the statistical average of the data provides an effective prior for balancing information preservation with model efficiency.

5.6. *Ablation study (RQ5)*

We perform ablation studies to evaluate the impact of key components in SemTraj, as summarized in Table 5. Removing the semantic condition leads to the most severe performance degradation, with density error increasing by over 190% and pattern score dropping significantly. It confirms that the OD and time attributes are indispensable for guiding the generative process toward spatially and temporally coherent outcomes. Similarly, disabling the adaptive resampling module results in a measurable increase in density error, highlighting its role in preserving temporal fidelity and optimizing the trade-off between sequence length and computational efficiency during training.

We further compare TDFormer with alternative designs based on Transformer, such as Cross Attention (Zhu et al., 2024) and In-Context

Table 6
Details of SemTraj models. We set model configurations for the Small (S), Medium (M), and Large (L) variants.

Model	Layers N	Hidden size d	Heads	GFlops	Training time	Inference time
SemTraj-S	10	64	8	0.2	343	102
SemTraj-M	18	128	16	1.4	390	115
SemTraj-L	28	128	16	2.2	501	146

Training and Inference time is represented as seconds/batch.

Table 7
Dataset scaling study on SemTraj.

Dataset ratio	Xi'an		
	Density (↓)	Length (↓)	Pattern (↑)
25%	0.0240	0.0311	0.7035
50%	0.0082	0.0096	0.8027
75%	0.0046	0.0091	0.8580
100%	0.0035	0.0083	0.8611

↓: lower is better, ↑: higher is better.

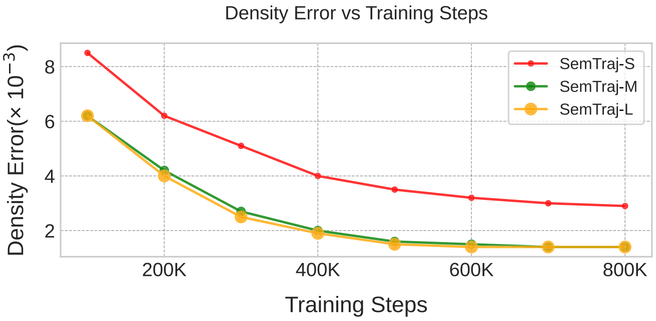


Fig. 6. Model scaling study on SemTraj.

Conditioning (Zhu et al., 2023). TDFormer achieves lower density errors and higher pattern scores, demonstrating its superior ability to integrate semantic attributes effectively. It demonstrates that simply applying these mechanisms to a DiT backbone is suboptimal for trajectory generation. SALN's layer-wise, feature-wise modulation offers a more precise and stable way for the denoising process across all network depths, leading to superior integration of semantic constraints. Finally, removing SALN with zero-initialization leads to moderate performance degradation, indicating that zero-initialization plays an important role in stabilizing optimization and enhancing the model's ability to capture trajectory patterns.

5.7. Model scaling and dataset scaling (RQ6)

The analysis of dataset scaling and model scaling provides valuable insights into how data volume and model capacity influence performance. In real-world scenarios, trajectory data is often limited, making it essential to assess model robustness under varying data conditions. For model scaling, we define three variants of SemTraj: SemTraj-S, SemTraj-M, and SemTraj-L, based on different depths and sizes of the model, characterized by the number of layers (N) and the hidden size (d) in each layer, as shown in Table 6. GFlops, Training and Inference time are also recorded for practical measure.

As shown in Table 7, increasing the dataset size from 25% to 100% yields significant improvements across all evaluation metrics, underscoring the importance of data volume in enhancing model effectiveness. Notably, even when trained on only 75% of the data, SemTraj is still able to generate trajectories that closely resemble real-world patterns, demonstrating strong generalization in low-data settings, which is a highly practical advantage.

As illustrated in Fig. 6, larger models, particularly SemTraj-M, consistently outperform smaller ones throughout training, as evidenced by lower density errors. This indicates that increasing model capacity enhances the learning of complex trajectory patterns. However, performance gains diminish beyond a certain point. SemTraj-L shows minimal improvement over SemTraj-M, suggesting that excessively large models offer diminishing returns, and scaling should be approached with consideration for efficiency.

6. Conclusion

In summary, SemTraj integrates a diffusion-Transformer backbone with semantic-adaptive layer normalization and a conditional VAE-based length generator to enable highly controllable, variable-length trajectory synthesis based on OD and departure time attributes. Through adaptive positional encoding and dynamic modulation of feature statistics using semantic information, SemTraj achieves strong controllability while maintaining efficient and stable training. Its unified architecture supports end-to-end generation of trajectories that accurately reflect both spatial endpoints and realistic temporal patterns. Experimental results on three real-world datasets show that SemTraj consistently outperforms both traditional and state-of-the-art diffusion-based models in OD accuracy, distributional alignment, and pattern fidelity. These improvements are largely attributed to semantic adaptive layer normalization and adaptive resampling strategies.

While SemTraj advances controllable trajectory generation, our work has some limitations. First, the model learns geographical feasibility implicitly from data without explicit road network encoding. Future work could explicitly integrate graph neural networks to incorporate road topology, further ensuring strict adherence. We plan to incorporate contrastive learning to further improve trajectory fidelity and conditional controllability. Additionally, extending the framework to handle multimodal conditions and few-shot scenarios would enhance its applicability in complex, data-scarce real-world settings. Second, the current evaluation metrics may not capture all nuances of physical realism. Developing more fine-grained metrics for kinematic plausibility or road adherence would be valuable. Third, the framework prioritizes high fidelity over formal privacy guarantees. Exploring techniques like differential privacy within the diffusion process presents a significant challenge for achieving both utility and privacy.

CRediT authorship contribution statement

Yuchen Jiang: Conceptualization, Investigation, Methodology, Writing – review & editing; **Guanhua Chen:** Writing – review & editing; **Shiyao Zhang:** Writing – review & editing; **Liang Han:** Writing – review & editing; **James Jianqiao Yu:** Writing – review & editing, Supervision.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant Nos. 62506097, U25A20530, U25B2045, by Shenzhen Basic Research Program Natural Science Foundation under Grant No. JCYJ20250604145542055, and in part by Guangdong Regional Joint Fund for Basic and Applied Basic Research Fund (No. 2024A1515110203).

References

Cao, P., Zhou, F., Song, Q., & Yang, L. (2024). Controllable generation with text-to-image diffusion models: A survey. arXiv preprint arXiv:2403.04279.

Chekol, A. G., & 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, X., Zhang, H., Zhao, F., Hu, Y., Tan, C., & Yang, J. (2022). Intention-aware vehicle trajectory prediction based on spatial-temporal dynamic attention network for internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23* (10), 19471–19483.

Das, H. P., Tran, R., Singh, J., Yue, X., Tison, G., Sangiovanni-Vincentelli, A., & Spanos, C. J. (2022). Conditional synthetic data generation for robust machine learning applications with limited pandemic data. *Proceedings of the AAAI Conference on Artificial Intelligence* (11792–11800). (*36*).

Ding, W., Xu, M., & Zhao, D. (2020). CMTS: A conditional multiple trajectory synthesizer for generating safety-critical driving scenarios. In *2020 IEEE international conference on robotics and automation (ICRA)* (pp. 4314–4321). https://doi.org/10.1109/ICRA40945.2020.9197145

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

Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X., & Zhang, F. (2017). Identifying human mobility via trajectory embeddings. In *Ijcai* (pp. 1689–1695). (*vol. 17*).

Guo, B., Wang, Z., Yu, Z., Wang, Y., Yen, N. Y., Huang, R., & Zhou, X. (2015). Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Computing Surveys (CSUR)*, *48* (1), 1–31.

Gutowski, N., Amghar, T., Camp, O., & Hammoudi, S. (2017). A framework for context-aware service recommendation for mobile users: A focus on mobility in smart cities. *From data to decision*, (pp. 1–17).

Gutowski, N., Camp, O., Chhel, F., Amghar, T., & Albers, P. (2019). Improving bandit-based recommendations with spatial context reasoning: An online evaluation. In *2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI)* (pp. 1366–1373). IEEE.

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

Jiang, B., Li, J., Yue, G., & Song, H. (2021a). Differential privacy for industrial internet of things: Opportunities, applications, and challenges. *IEEE Internet of Things Journal*, *8* (13), 10430–10451.

Jiang, H., Li, J., Zhao, P., Zeng, F., Xiao, Z., & Iyengar, A. (2021b). Location privacy-preserving mechanisms in location-based services: A comprehensive survey. *ACM Computing Surveys (CSUR)*, *54* (1), 1–36.

Kong, Z., Ping, W., Huang, J., Zhao, K., & Catanzaro, B. (2020). DiffWave: A versatile diffusion model for audio synthesis. In *International conference on learning representations*.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H. S., & Chandraker, M. (2017). Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 336–345).

Li, M., Tong, P., Li, M., Jin, Z., Huang, J., & Hua, X.-S. (2021). Traffic flow prediction with vehicle trajectories. *Proceedings of the AAAI Conference on Artificial Intelligence* (294–302). (*35*).

Long, X., Jin, L., & Joshi, J. (2012). Exploring trajectory-driven local geographic topics in foursquare. In *Proceedings of the 2012 ACM conference on ubiquitous computing* (pp. 927–934).

Mirza, M. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

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

Peebles, W., & Xie, S. (2023). Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4195–4205).

Rao, J., Gao, S., Kang, Y., & Huang, Q. (2020). LSTM-TrajGAN a deep learning approach to trajectory privacy protection. In K. Janowicz, & J. A. Verstegen (Eds.), *11th international conference on geographic information science (GIScience 2021) - Part I* (pp. 12:1–12:17). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik (*vol. 177*). Leibniz International Proceedings in Informatics (LIPIcs). https://doi.org/10.4230/LIPIcs.GIScience.2021.I.12

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684–10695).

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., & Norouzi, M. (2022). Palette: Image-to-image diffusion models. In *ACM siggraph 2022 conference proceedings* (pp. 1–10).

Shi, H., Yao, Q., Guo, Q., Li, Y., Zhang, L., Ye, J., Li, Y., & Liu, Y. (2020). Predicting origin-destination flow via multi-perspective graph convolutional network. In *2020 IEEE 36th international conference on data engineering (ICDE)* (pp. 1818–1821). https://doi.org/10.1109/ICDE48307.2020.00178

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (pp. 2256–2265). PMLR.

Song, J., Meng, C., & Ermon, S. (2020a). Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502.

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

Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-or, D., & Bermano, A. H. (2023). Human motion diffusion model. In *The eleventh international conference on learning representations*.

Wang, X., Liu, X., Lu, Z., & Yang, H. (2021). Large scale GPS trajectory generation using map based on two stage GAN. *Journal of Data Science*, *19* (1), 126–141.

Wei, T., Lin, Y., Guo, S., Lin, Y., Huang, Y., Xiang, C., Bai, Y., & Wan, H. (2024). Diff-RNTraj: A structure-aware diffusion model for road network-constrained trajectory generation. *IEEE Transactions on Knowledge and Data Engineering*, (pp. 1–15). https://doi.org/10.1109/TKDE.2024.3460051

Xi, L., Hanzhou, C., & Clio, A. (2018). TrajGANs: Using generative adversarial networks for geo-privacy protection of trajectory data (Vision paper). In *Location Privacy and Security Workshop*, .

Xia, T., Song, X., Fan, Z., Kanasugi, H., Chen, Q., Jiang, R., & Shibasaki, R. (2018). DeepRailway: A deep learning system for forecasting railway traffic. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)* (pp. 51–56). https://doi.org/10.1109/MIPR.2018.00017

Zaier, M., Wannous, H., Drira, H., & Boonaert, J. (2025). Pedestrian trajectory prediction: A literature review and current trends. *Neural Computing and Applications*, , *37* (35-36) 28869–28906.

Zhu, Y., Ye, Y., Liu, Y., & Yu, J. J. Q. (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., & Yu, J. (2023). Difftraj: Generating GPS trajectory with diffusion probabilistic model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 65168–65188). Curran Associates, Inc. (*vol. 36*).

Zhu, Y., Yu, J. J., Zhao, X., Liu, Q., Ye, Y., Chen, W., Zhang, Z., Wei, X., & Liang, Y. (2024). ControlTraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 4676–4687).