Two-Stage Fine-Grained Trajectory Generation Constrained by Road Networks

Zewu Lv, **Zipei Fan**, **Zhiwen Zhang** and **Xuan Song**†
School of Artificial Intelligence, Jilin University
lvzw24@mails.jlu.edu.cn, fanzipei@jlu.edu.cn, zhiwen@locationmind.com, songxuan@jlu.edu.cn

Abstract

Trajectory generation is a pivotal technique for mitigating data sparsity, but existing methods struggle to simultaneously achieve strict road network alignment and capture realistic movement characteristics. To bridge this gap, we propose RNTrajGen, a two-stage fine-grained trajectory generation framework constrained by road networks. Specifically, we first develop a Road Network Knowledge-Enhanced Encoder (RNKEE) to provide semantically rich representations for trajectory generation. By leveraging graph attention networks, RNKEE encodes static prior knowledge of the road network while integrating self-supervised learning to extract latent movement patterns from real-world trajectories. Subsequently, RNTrajGen follows a two-stage generation strategy: it first generates a sequence of road segments to ensure strict topological alignment, and then infers the moving ratios of trajectory points along road segments to capture fine-grained movement characteristics. Extensive experiments on two real-world datasets demonstrate that RNTrajGen significantly outperforms state-of-the-art baselines, and the generated trajectories exhibit high utility in downstream prediction tasks. Our code is available at https://github.com/jkzh986/RNTrajGen.

1 Introduction

With the widespread adoption of GPS-enabled devices in Intelligent Transportation Systems (ITS), recording vehicle trajectories has become highly convenient. Mining this trajectory data plays a crucial role in supporting various downstream intelligent navigation applications, such as path prediction [Yang *et al.*, 2018; Wen *et al.*, 2021; Wen *et al.*, 2022]. However, due to the sensitivity of such data and associated privacy concerns [Hu *et al.*, 2024; Kapp *et al.*, 2023], publicly available trajectory datasets are typically limited in scale. To address these challenges, trajectory data generation has emerged as a significant research direction. Its goal is to produce trajectory data that closely aligns with real-world distributions, thereby resolving the difficulty of acquiring large-scale trajectory datasets.

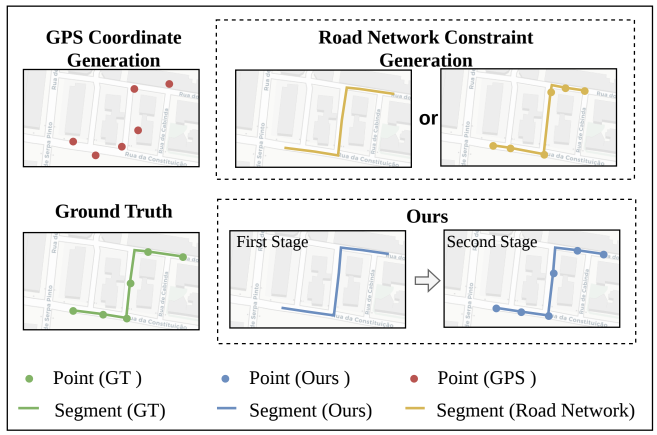


Figure 1: Comparison of trajectory generation models.

In recent years, trajectory generation methods have been extensively studied. However, existing approaches still face challenges in generating fine-grained trajectories that align with road networks while preserving realistic movement characteristics. Specifically, as illustrated in Fig. 1, some studies [Zhu *et al.*, 2023; Zhu *et al.*, 2024] attempt to represent trajectories using sequences of GPS points in a generated geographic coordinate system (i.e., latitude and longitude). However, trajectories generated by such methods often suffer from poor alignment with road networks. Other studies [Shi *et al.*, 2024; Rao *et al.*, 2025] generate trajectories directly on road networks, resolving the alignment issue. However, most methods remain limited to coarse-grained road segment-level trajectories. Although existing research [Wei *et al.*, 2024] attempts to generate fine-grained trajectories constrained by road networks, it still fails to accurately capture the movement characteristics of trajectory points along road segments. Furthermore, existing approaches [Rao *et al.*, 2025] typically employ data-driven graph representation techniques to learn road network representations, enabling trajectory generation over road networks. This approach neglects prior knowledge about the road network, such as static road segment attributes and road network topology, leading to semantically

†Corresponding author.

weak road segment embeddings. In summary, existing trajectory generation tasks face two core challenges: (i) How to ensure generated trajectories align with road networks while preserving realistic movement characteristics; and (ii) How to learn representations of road networks to provide semantically rich road segment embeddings for trajectory generation.

To address these challenges, we propose a two-stage fine-grained trajectory generation framework constrained by road networks, named RNTrajGen. It not only resolves the misalignment between trajectories and road networks but also generates fine-grained trajectories with realistic movement characteristics. Specifically, (i) To provide semantically rich road segment embeddings for trajectory generation, we design a road network knowledge-enhanced encoder. This encoder innovatively combines a graph attention network with self-supervised learning to capture rich road network features from both road network prior knowledge and the movement patterns of real trajectories. (ii) To ensure generated trajectories align with the road network while preserving realistic movement characteristics, we adopt a two-stage generation strategy. The first stage generates a sequence of road segments, ensuring the trajectories align with the road network. In the second stage, the obtained road segment sequence and its static attributes serve as semantic conditions to further infer the moving ratio of trajectory points along the road segments, thereby capturing the realistic movement characteristics of the trajectories. In summary, the main contributions of our work are as follows:

- We propose RNTrajGen, a framework for two-stage fine-grained trajectory generation constrained by road networks. This staged modeling approach effectively generates fine-grained trajectories that align with road networks while exhibiting realistic movement characteristics.
- We designed a Road Network Knowledge-Enhanced Encoder (RNKEE) that integrates graph attention networks with self-supervised learning. By leveraging both static prior knowledge from road networks and movement patterns from real-world trajectories, it provides semantically rich road segment embeddings for trajectory generation tasks.
- We conducted extensive experiments on two real-world trajectory datasets. Experimental results demonstrate that the proposed RNTrajGen consistently outperforms state-of-the-art methods across all trajectory quality metrics while exhibiting strong utility in downstream tasks.

2 Related Work

Trajectory generation techniques have been extensively studied for producing large-scale realistic movement data. Existing approaches can be broadly categorized into model-based and learning-based methods.

Model-based Methods. Early research primarily relied on predefined physical rules or probabilistic models to simulate human movement. For instance, [Kim *et al.*, 2006] extracted statistical features such as transition matrices and velocity distributions to construct generative models. While these approaches offer interpretability, they often oversimplify the inherent randomness and complexity of human mobility, limiting their practical utility.

Learning-Based Methods. With the advancement of Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), data-driven generative models have been integrated into trajectory generation. [Huang *et al.*, 2019] combined the VAE with a Seq2Seq model to learn the latent distribution of trajectories, while [Long *et al.*, 2023] employed a hierarchical VAE structure within a variational temporal point process framework to process user trajectory data. Additionally, grid-based or image-based GANs [Wang *et al.*, 2021; Yuan *et al.*, 2022; Ouyang *et al.*, 2018; Feng *et al.*, 2020] have been employed for trajectory generation. However, these methods trade off spatial resolution for computational complexity, only generating coarse-grained trajectory data. Recent research has begun applying diffusion models to trajectory generation, aiming to generate more fine-grained trajectories. [Zhu *et al.*, 2023] guided diffusion models to generate trajectories by incorporating trajectory attributes as additional constraints. [Zhu *et al.*, 2024] extracted road segment information traversed by trajectories as topological constraints to enhance fine-grained control over generated trajectories. [Wei *et al.*, 2024] attempted to jointly represent trajectory points using road segments and moving ratios, enabling end-to-end generation of trajectories constrained by road networks. [Guo *et al.*, 2025] introduced a coarse-to-fine cascaded framework, guiding GPS trajectory generation through road segment-level trajectory representations.

3 Problem Definition

Definition 1 (Road Network)**.** *A road network is modeled as a directed graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_{|V|}\}$ is the set of vertices representing road segments, and $E \subseteq V \times V$ is the set of directed edges representing the topological connectivity between adjacent road segments.*

Definition 2 (Road Network-Constrained Trajectory)**.** *A road network-constrained trajectory is defined as $\tau_{RN} = [q_1, q_2, \dots, q_n]$, where each trajectory point $q_i = (s_i, r_i)$ consists of a road segment s_i and a moving ratio r_i. Specifically, if q_i is the first point on s_i, then $r_i = \frac{d_i}{l_i}$, where d_i denotes the traveled distance from the start of s_i to q_i, and l_i is the length of s_i. Otherwise, $r_i = \frac{d_i}{l_i - d_{i-1}}$, where d_i denotes the distance traveled from q_{i-1} to q_i, and d_{i-1} is the traveled distance from the start of s_i to q_{i-1}.*

Definition 3 (Trajectory Generation)**.** *Given a real-world trajectory dataset, the objective of this problem is to train a deep learning model to generate a dataset that follows the distribution of the real trajectory data.*

4 Methodology

To address the dual challenges of road network alignment and realistic movement characteristics, we propose RNTrajGen, a two-stage fine-grained trajectory generation framework constrained by road networks. As shown in Fig. 2, this framework consists of a road segment representation learning module, a road segment sequence generation module, and a moving ratio inference module. The following subsections will introduce these modules in turn.

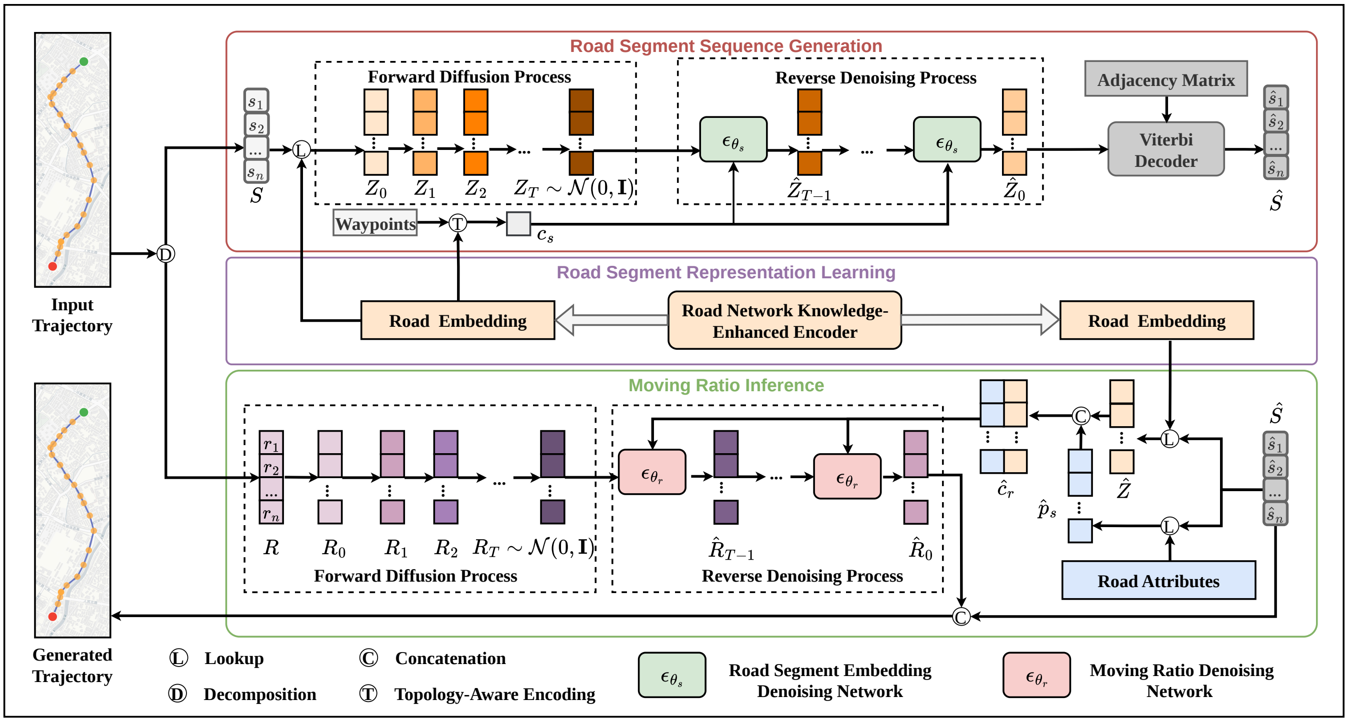


Figure 2: The framework of RNTrajGen, which consists of a road segment representation learning module, a road segment sequence generation module, and a moving ratio inference module.

4.1 Road Representation Learning

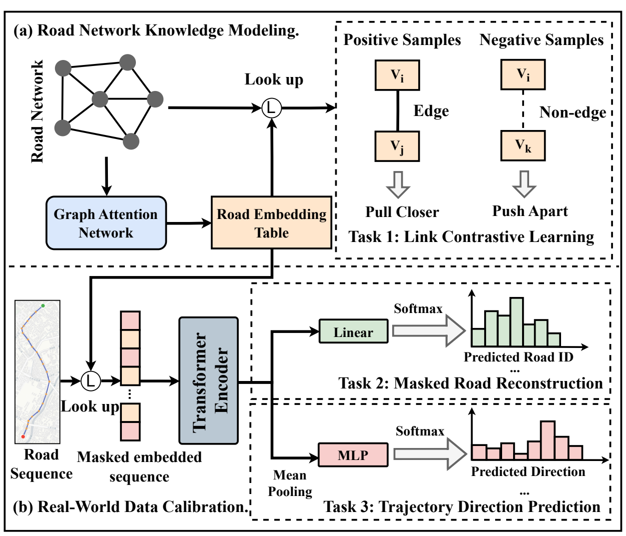


Figure 3: The pipeline of Road Network Knowledge-Enhanced Encoder.

Existing trajectory generation studies predominantly employ purely data-driven graph representation learning for encoding road segments, neglecting explicit prior knowledge such as road segment attributes (e.g., level, length and direction) and topological structure. To address this, we propose the Road Network Knowledge-Enhanced Encoder (RNKEE). As illustrated in Fig. 3, RNKEE first employs a GNN [Veličković *et al.*, 2018] to encode road segment attributes and topological constraints, establishing a prior baseline. Subsequently, it extracts movement patterns from trajectory sequences through a self-supervised learning framework [Devlin *et al.*, 2019]. This dual-drive mechanism enables the generated road embeddings to integrate both prior knowledge of road networks and real-world movement patterns.

Road Network Knowledge Modeling

In this stage, we first employ a graph attention network (GAT) to encode the static attributes of road segments, establishing a preliminary local spatial context. To further enhance the discriminative perception of embedded representations toward road network topology and mitigate the over-smoothing issue [Li *et al.*, 2018] in deep graph neural networks, we designed a link contrastive learning task [Hadsell *et al.*, 2006]. This task explicitly models the connectivity between road segments, forcing topologically related segment embeddings to cluster closer together in vector space while pushing disconnected segments apart. Specifically, we use the binary cross-entropy (BCE) loss function based on dot product similarity as the optimization objective:

$$\mathcal{L}_{\text{link}} = -\frac{1}{|\mathcal{S}|}\sum_{(i,j)\in\mathcal{S}}[y_{ij}\log(\sigma(e_i^\top e_j)) + (1-y_{ij})\log(1-\sigma(e_i^\top e_j))] \quad (1)$$

where $\mathcal{S}$ denotes the set of segment pairs containing positive and negative samples, and $y_{ij} \in \{0,1\}$ indicates whether a real physical connection exists between road segments.

Real-World Data Calibration

While road network knowledge provides a structural foundation, it often fails to capture real-world movement patterns (e.g., traffic flow). We thus introduce a self-supervised calibration framework via masked road segment reconstruction. Specifically, given a sequence $S = [s_1, \ldots, s_n]$ of discrete road segments, we first obtain its initial embedding sequence via the GAT-learned node embedding table, and then apply a BERT-style masking strategy on the embeddings (80% mask, 10% random, 10% unchanged) to produce a corrupted embedding sequence $\hat{\mathbf{S}} \in \mathbb{R}^{n \times d}$. This sequence is passed through a Transformer Encoder [Vaswani *et al.*, 2017] to compute hidden states $\mathbf{H} \in \mathbb{R}^{n \times d}$, and a linear layer finally predicts the actual segment IDs for all masked positions $\mathcal{M}$. The reconstruction loss is defined as:

$$\mathcal{L}_{\text{msr}} = -\frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \log P_{\theta_e}(s_i \mid \hat{\mathbf{S}}) \tag{2}$$

Here, s_i represents the actual road segment ID at position i, and θ_e denotes the set of learnable parameters of the model.

To further enhance spatial structural awareness, we incorporate trajectory direction prediction as an auxiliary task. We discretize geographic orientations into D equal intervals. A global representation $\bar{\mathbf{h}} \in \mathbb{R}^d$ is obtained via mean pooling over $\mathbf{H}$ and fed into an MLP to predict the probability distribution of the destination's relative orientation from the origin. The loss function $\mathcal{L}_{dir}$ is defined as:

$$\mathcal{L}_{\text{dir}} = -\sum_{k=1}^{D} y_k \log P_{\theta_e}(d = k \mid \hat{\mathbf{S}}) \tag{3}$$

Here, $y_k \in \{0, 1\}$ is the k-th component of the one-hot encoded target direction vector, and $P_{\theta_e}(d = k \mid \hat{\mathbf{S}})$ is the predicted probability that the trajectory direction d belongs to the k-th class.

Multi-Task Joint Optimization

To embed static knowledge of the road network while capturing real-world movement patterns, we jointly optimize three tasks: link contrastive learning, masked road segment reconstruction, and trajectory direction prediction. The final total loss function $\mathcal{L}_{\text{total}}$ is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{msr}} + \lambda_1 \mathcal{L}_{\text{dir}} + \lambda_2 \mathcal{L}_{\text{link}} \tag{4}$$

Where λ_1 and λ_2 are hyperparameters that adjust the contribution ratios of each task.

4.2 Road Sequence Generation

To train the diffusion model to learn road segment transition patterns within trajectories, we first derive road segment-level trajectories from the road network-constrained trajectory τ_{RN}, obtaining the discrete road segment sequence $S = [s_1, \ldots, s_n]$. Different from direct modeling in discrete space, we utilize pre-trained road segment embeddings to map the discrete sequence S into a continuous latent representation $\mathbf{Z}_0 \in \mathbb{R}^{n \times d}$. Following the standard diffusion paradigm, RNTrajGen recovers the clean representation $\mathbf{Z}_0$ from the noisy sequence $\mathbf{Z}_t$ by predicting the added noise $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times d}$ at each diffusion step t. The denoising network follows the Traj-UNet architecture [Zhu *et al.*, 2023], which comprises downsampling, intermediate, and upsampling stages. To effectively capture both local and global dependencies, each stage integrates 1D-CNN-based residual modules [He *et al.*, 2016] with Transformer Encoders, ensuring robust modeling of long-range trajectory patterns.

To ensure the generated results align with the real distribution, we propose a topology-aware conditional encoding method. In contrast to traditional GPS generation methods that employ grid IDs with granularity trade-offs, we refine the condition to the road segment level, inspired by MTrajRec [Ren *et al.*, 2021]. Specifically, for key points p_i such as origin and destination, we model their association with surrounding road segments using an exponential function:

$$f(d_{j,p_i}) = \begin{cases} \exp(-(d_{j,p_i})^2/\beta^2), & \text{if } d_{j,p_i} < \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Here, d_{j,p_i} denotes the geodesic distance between point p_i and its projection onto road segment j, while β is a hyperparameter related to the road network. Clearly, the function $f(d_{j,p_i})$ only considers road segments j whose distance to point p_i falls below the threshold δ. Subsequently, we utilize the pre-trained road segment embedding representation $\mathbf{e}_j \in \mathbb{R}^d$ to obtain the road segment-level representation $\mathbf{h}_i \in \mathbb{R}^d$ for point p_i through weighted summation:

$$\mathbf{h}_i = \sum_{j \in \mathcal{R}} f(d_{j,p_i}) \mathbf{e}_j \tag{6}$$

Here, $\mathcal{R}$ denotes the set of candidate road segments. Furthermore, to capture the temporal dependencies between trajectory point locations, we model the sequence of waypoints using a GRU. The final hidden state is utilized as a contextual condition vector $\mathbf{c}_s \in \mathbb{R}^d$ to guide the denoising network in the road segment generation task. The loss function for noise prediction is defined as follows

$$\mathcal{L}_{\text{n}} = \mathbb{E}_{\mathbf{Z}_0, \boldsymbol{\epsilon}, t} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_s} \left(\sqrt{\bar{\alpha}_t} \mathbf{Z}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t, \mathbf{c_s} \right) \right\|^2 \right] \tag{7}$$

Here, θ_s denotes the learnable parameters of the denoising network in the road segment generation task, and $\bar{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_i)$ represents the predefined cumulative noise coefficient. However, the noise prediction loss does not account for the structure of the road segment embeddings, leading to invalid recovered road segment representations. Therefore, we simultaneously compute the estimated road segment embedding $\tilde{\mathbf{Z}}_0 \in \mathbb{R}^{n \times d}$ and optimize the road segment embedding loss $\mathcal{L}_{\text{e}}$:

$$\tilde{\mathbf{Z}}_0 = (\mathbf{Z}_t - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_{\theta_s}(\mathbf{Z}_t, t, \mathbf{c}_s)) / \sqrt{\bar{\alpha}_t} \tag{8}$$

$$\mathcal{L}_{\text{e}} = \left\| \mathbf{Z}_0 - \tilde{\mathbf{Z}}_0 \right\|^2 \tag{9}$$

Thus, the total loss function for the sequence generation stage is defined as follows:

$$\mathcal{L}_{\text{segment}} = \mathcal{L}_{\text{n}} + \mathcal{L}_{\text{e}} \tag{10}$$

For the latent representation $\hat{\mathbf{Z}}_0 \in \mathbb{R}^{n \times d}$ obtained via the reverse denoising process of the diffusion model, directly mapping it back to discrete IDs via nearest-neighbor search cannot guarantee the topological connectivity of the generated road segment sequences. Here, we combine the road segment adjacency matrix with the Viterbi algorithm to search for globally optimal discrete sequences $\hat{S}$ that conform to the road network topology.

4.3 Moving Ratio Inference

As the second stage of RNTrajGen, we employ a Teacher Forcing strategy [Williams and Zipser, 1989] to train the diffusion model. Specifically, during training, we first decompose the real trajectory data τ_{RN} into a discrete road segment sequence $S = [s_1, \ldots, s_n]$ and a moving ratio sequence $\mathbf{R}_0 = [r_1, \ldots, r_n]$. Subsequently, using pre-trained road segment embeddings and static attributes (e.g., length, level), we map S to a semantically rich embedding sequence $\mathbf{Z} \in \mathbb{R}^{n \times d}$ and a static attribute sequence $\mathbf{p}_s \in \mathbb{R}^{n \times d_p}$. During the forward diffusion process, Gaussian noise $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times 1}$ is added to the original moving ratio sequence $\mathbf{R}_0 \in \mathbb{R}^{n \times 1}$ at time step t to generate the noisy sequence $\mathbf{R}_t$. During the reverse denoising process, the semantic feature embedding sequence $\mathbf{Z}$ is concatenated with the static attribute sequence $\mathbf{p}_s$ to form the conditional input $\mathbf{c}_r \in \mathbb{R}^{n \times (d+d_p)}$. We then optimize the denoising network parameters θ_r by minimizing the predicted noise loss, defined as follows:

$$\mathcal{L}_{\text{ratio}} = \mathbb{E}_{\mathbf{R}_0,\boldsymbol{\epsilon},t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_r}\left(\sqrt{\bar{\alpha}_t}\mathbf{R}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c}_r\right)\right\|^2\right] \tag{11}$$

Given that the inference of moving ratios exhibits local temporal correlations, the denoising network ϵ_{θ_r} adopts a UNet architecture [Ronneberger *et al.*, 2015] based on 1D CNNs. During inference, the model no longer relies on the real road segment sequence S, but instead uses the discrete sequence $\hat{S}$ generated in the first stage. $\hat{S}$ is first mapped into an embedding sequence $\hat{\mathbf{Z}}$, which is then concatenated with the corresponding static attribute sequence $\hat{\mathbf{p}}_s$ of road segments to construct the condition $\hat{\mathbf{c}}_r$. Subsequently, the denoising network ϵ_{θ_r} performs the reverse diffusion process conditioned on $\hat{\mathbf{c}}_r$. After T steps, it generates a moving ratio sequence $\hat{\mathbf{R}}_0$ from Gaussian noise. Finally, combining $\hat{S}$ with $\hat{\mathbf{R}}_0$ yields the fine-grained trajectory $\hat{\tau}_{RN} = [\hat{S}, \hat{\mathbf{R}}_0]$, which aligns with the road network topology and exhibits realistic movement patterns.

4.4 Training and Sampling

For training, the RNTrajGen model employs parallel training to optimize the diffusion denoising networks for both stages. For the road segment sequence generation stage, the denoising network $\boldsymbol{\epsilon}_{\theta_s}$ is optimized by minimizing the sum of estimated losses between noise and road segment embeddings, as defined in Eq. 10. For the moving ratio inference stage, the denoising network $\boldsymbol{\epsilon}_{\theta_r}$ is optimized by minimizing the estimated noise loss, as defined in Eq. 11. For sampling, the RNTrajGen model employs a two-stage cascaded generation approach. It first generates road segment sequences, then infers the moving ratio of trajectory points along these segments by using the sequences and their static attributes as semantic conditions.

5 Experiment

This section first details the experimental setup, including datasets, baseline methods, evaluation metrics, and implementation details. Subsequently, we conduct extensive experiments to investigate the following research questions:

- **RQ1 (Main Performance):** How does the proposed RNTrajGen perform compared with the state-of-the-art trajectory generation baselines?
- **RQ2 (Ablation Study):** What are the individual contributions of the key technical components in RNTrajGen to the overall performance?
- **RQ3 (Embedding Effectiveness):** Can the Road Network Knowledge-Enhanced Encoder provide semantically rich road segment embeddings?
- **RQ4 (Generation Fidelity):** Can RNTrajGen generate fine-grained trajectories aligned with the road network?
- **RQ5 (Downstream Task Performance):** How does RNTrajGen perform on downstream tasks?

5.1 Experimental Setups

Datasets. We evaluate the performance of RNTrajGen on two publicly available real-world taxi trajectory datasets: Porto and Chengdu. To incorporate road network constraints, we utilize the OSMnx tool to download the road network from OpenStreetMap. We then project the raw GPS trajectories onto the road network via a map matching algorithm.

Baselines. We compared RNTrajGen with several representative methods, including traditional approaches such as Node2Vec and advanced generative methods like SVAE, Diff-RNTraj, DiffTraj, and ControlTraj. Since SVAE, DiffTraj, and ControlTraj only produce GPS trajectories, we subsequently employed a map matching algorithm to project their GPS trajectories onto the road network in order to obtain trajectories constrained by the road network.

Evaluation Metrics. We employ the Johnson-Shannon Divergence (JSD) to assess the distributional similarity between generated trajectories and real trajectories: total distance (JSD-TD), step distance (JSD-SD), grid point frequency (JSD-GPD), and road segment frequency (JSD-RS). Additionally, we define Trajectory Adjacency Connectivity (TAC) and Trajectory Complete Connectivity (TCC) to quantify the proportion of adjacent points conforming to road network topology connectivity and the proportion of trajectories where all points meet topology connectivity, respectively. Lower JSD values and higher TAC or TCC values indicate superior generation quality.

5.2 Experimental Results

Main Performance(RQ1)

Table 1 summarizes the performance of baseline methods on real-world datasets. Results demonstrate that RNTrajGen comprehensively outperforms baseline methods on both the Porto and Chengdu datasets, achieving particularly significant improvements on the complex Porto dataset, highlighting its robustness to intricate topological structures. Additionally, we observe the following findings: (i) Learning-based methods outperform traditional approaches on distributed metrics, while Node2Vec maintains an advantage

Model	Porto						Chengdu					
	JSD-TD(↓)	JSD-SD(↓)	JSD-GPD(↓)	JSD-RS(↓)	TAC(↑)	TCC(↑)	JSD-TD(↓)	JSD-SD(↓)	JSD-GPD(↓)	JSD-RS(↓)	TAC(↑)	TCC(↑)
Node2vec	0.328315	0.067598	0.284082	0.099659	**100.00**	**100.00**	0.447927	0.144982	0.354370	0.088788	**100.00**	**100.00**
SVAE	0.327810	0.076996	0.189638	0.202988	56.10	0.03	0.010489	0.008478	0.202559	0.218843	89.51	15.02
Diff-RNTraj	0.043753	0.023707	0.036073	0.037146	88.93	14.12	0.078314	0.046964	0.026099	0.028477	88.70	20.44
DiffTraj	0.000952	0.003434	0.029100	0.030676	94.50	49.77	0.000337	0.002923	0.051434	0.061067	95.51	48.72
ControlTraj	0.000850	0.002376	0.011857	0.011033	98.05	73.21	0.000349	0.001874	0.015184	0.022753	98.29	74.47
RNTrajGen	**0.000448**	**0.001043**	**0.005352**	**0.008274**	**100.00**	**100.00**	**0.000285**	**0.001646**	**0.006246**	**0.006921**	**100.00**	**100.00**

Table 1: Performance comparison on two datasets. In each column, the best and second-best methods are marked with **boldface** and underline, respectively. ↓ indicates lower is better, while ↑ indicates the opposite.

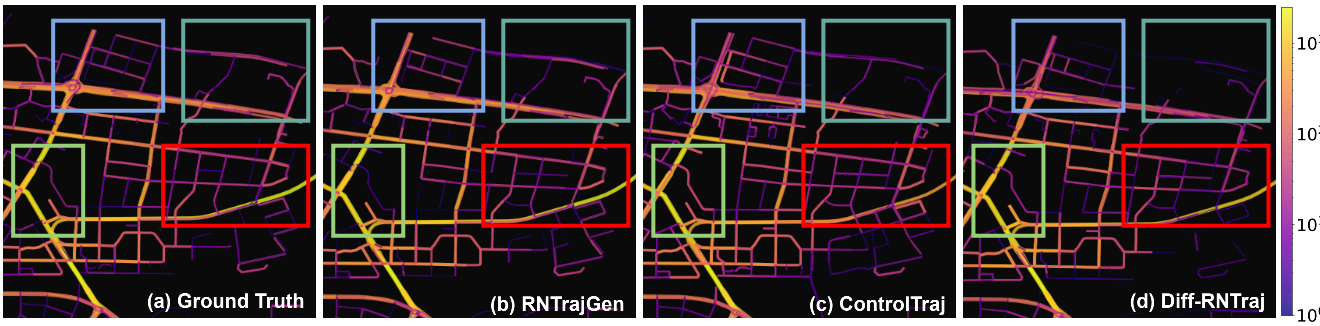


Figure 4: Visualization of road segment distribution in the Chengdu dataset, with rectangular boxes highlighting areas showing significant variation.

in topological connectivity through its random walk mechanism. Among learning-based methods, diffusion models outperform VAE due to their superior distributed learning capabilities. (ii) ControlTraj, an enhanced version of DiffTraj incorporating road network constraints, outperforms DiffTraj, confirming the critical role of road network constraints in trajectory generation.

Fig. 4 compares heatmaps of trajectory point distributions across road segments for various models in the Chengdu region. Results show that ControlTraj exhibits flow deviations on certain segments, indicating that its generated trajectories deviate from the road network during map matching. Diff-RNTraj performs worse than ControlTraj, probably due to its lack of conditional guidance. In contrast, RNTrajGen exhibits a distribution highly consistent with ground truth. This is attributable to the rich geosemantic embeddings provided by RNKEE and the precise constraints imposed on the generation process by its topology-aware guidance mechanism at the road segment level.

Ablation Study (RQ2)

The RNTrajGen framework comprises four key technical modules. To evaluate the independent contributions of each component, we constructed the following four variants:

- **w/o C**: This variant removes the topology-aware conditional guidance mechanism, degenerating into an unconditional trajectory generation model.
- **w/o RNKEE**: This variant replaces the Road Network Knowledge-Enhanced Encoder with the Node2Vec algorithm to learn road segment representations.
- **w/o Viterbi**: This variant employs a decoding method based on maximum cosine similarity instead of the Viterbi algorithm.
- **w/o Two Stage**: This variant abandons the two-stage generation strategy and adopts the generation approach of the Diff-RNTraj model, directly modeling the joint distribution of road segment embeddings and moving ratios within a single stage.

According to the results in Table 2, we draw the following conclusions: (i) Trajectories generated by variant sampling without the condition exhibit significant deviations across all distribution metrics, indicating that the topology-aware condition is crucial for accurately capturing the real data distribution. (ii) Compared to the variant using the Node2Vec algorithm, RNKEE achieves lower values of the road segment distribution metric (JSD-RS), reducing it by 16.28% and 25.50% on the Porto and Chengdu datasets, respectively. This demonstrates the advantage of RNKEE in capturing road network features, highlighting the limitations of purely data-driven approaches like Node2Vec. (iii) Trajectories generated via the variant decoding based on maximum cosine similarity exhibit a Trajectory Complete Connectivity (TCC) below 80%, significantly lower than RNTrajGen. This demonstrates the superiority of the Viterbi algorithm in ensuring topological connectivity of generated trajectories. (iv) The single-stage generation strategy exhibits significantly higher values in the distance distribution metric (JSD-SD) between adjacent trajectory points, with reductions of 84.24% on the Porto dataset and 70.82% on the Chengdu dataset achieved by the two-stage approach. However, in terms of the road segment distribution metric (JSD-RS), the single-stage approach performs slightly better than the two-stage modeling method, because the moving ratio provides additional information to assist in modeling road segment transitions. Nevertheless, the single-stage approach still fails to effectively capture the movement

Model	Porto						Chengdu					
	JSD-TD(↓)	JSD-SD(↓)	JSD-GPD(↓)	JSD-RS(↓)	TAC(↑)	TCC(↑)	JSD-TD(↓)	JSD-SD(↓)	JSD-GPD(↓)	JSD-RS(↓)	TAC(↑)	TCC(↑)
w/o C	0.005132	0.002813	0.020998	0.024127	**100.00**	**100.00**	0.002926	0.004787	0.021554	0.023837	**100.00**	**100.00**
w/o RNKEE	0.000616	0.001364	0.006030	0.009883	**100.00**	**100.00**	0.000614	0.003082	0.006778	0.009290	**100.00**	**100.00**
w/o Viterbi	0.000513	0.001361	0.005481	0.008524	98.28	72.39	0.000689	0.002327	0.006364	0.007296	98.60	78.97
w/o Two Stage	0.000563	0.006619	0.007441	**0.008230**	**100.00**	**100.00**	0.000798	0.005641	0.009261	**0.006465**	**100.00**	**100.00**
RNTrajGen	**0.000448**	**0.001043**	**0.005352**	0.008274	**100.00**	**100.00**	**0.000285**	**0.001646**	**0.006246**	0.006921	**100.00**	**100.00**

Table 2: Comparison of RNTrajGen with Variant Performance.

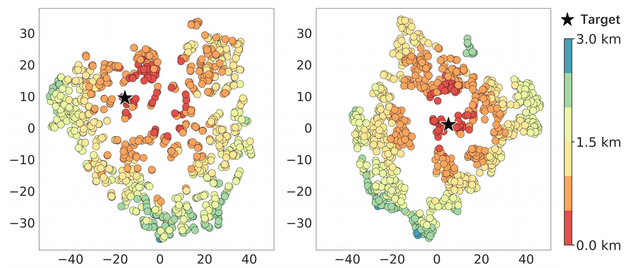


Figure 5: Visualization of the embedding of the Chengdu road segment with t-SNE. RNKEE (right) displays a structured concentric diffusion effect aligned with geographic proximity, whereas Node2Vec (left) lacks such spatial consistency.

characteristics of real trajectories.

Embedding Effectiveness (RQ3)

To verify whether RNKEE provides semantically rich embeddings, we evaluate the discriminability of road segments across varying geodesic distances in the latent space. Specifically, we sample a target road segment and its N-hop neighbors, binning them into 500m geographic intervals. The t-SNE visualization (Fig. 5) shows that, unlike Node2Vec, RNKEE embeddings exhibit a distinct concentric diffusion pattern, where road segments transition from center to periphery as their geodesic distance increases. This visual consistency demonstrates RNKEE's superior ability to capture spatial-semantic relationships.

Generation Fidelity (RQ4)

To visually assess the quality of the generated trajectories, we conducted trajectory generation experiments using RNTrajGen on the Chengdu dataset. As shown in Fig. 6, the generated results not only align well with the road network, but also exhibit movement characteristics similar to those of the ground-truth data. This further demonstrates RNTrajGen's strong capability to generate fine-grained trajectories that closely conform to the underlying road network.

Downstream Task Performance (RQ5)

To evaluate the utility of generated trajectories, we conduct a next-location prediction task using an LSTM model. We split the real dataset into training, validation, and test sets (6:2:2). By gradually replacing real training data with RNTrajGen-generated trajectories while keeping the total training size constant, we measure performance via Hits@n. As shown in Fig. 7, the model achieves robust results even with 0% real data. This result fully demonstrates the practical utility of trajectories generated by RNTrajGen for downstream tasks.

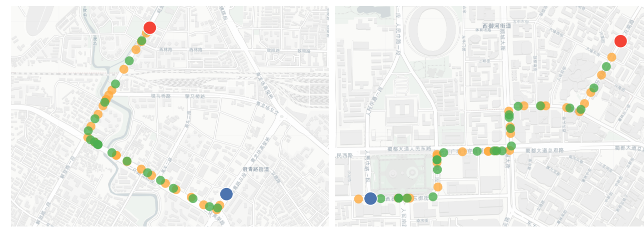
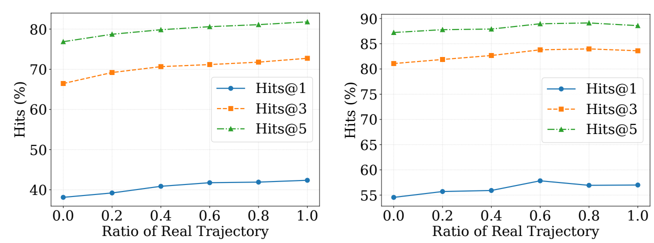


Figure 6: Visualization of condition-guided trajectory generation on the Chengdu dataset. Blue and red markers denote origins and destinations, respectively. Yellow and green points represent the generated and ground-truth intermediate points along road segments.



(a) Porto (b) Chengdu

Figure 7: The performance of next position prediction.

6 Conclusion

In this paper, we propose RNTrajGen, a two-stage fine-grained trajectory generation framework constrained by road networks, designed to generate fine-grained trajectories that align with road networks while exhibiting realistic characteristics. Specifically, we first design a Road Network Knowledge-Enhanced Encoder(RNKEE) to provide semantically rich segment representations for the trajectory generation task. Subsequently, a two-stage generation strategy is employed to produce fine-grained trajectories that are strictly aligned with the road network and exhibit realistic movement characteristics. Extensive experiments on real-world datasets demonstrate that RNTrajGen consistently outperforms state-of-the-art baseline models. Furthermore, ablation studies validate the independent contributions of each component within the model, with visual analysis further confirming that RNKEE can capture road network knowledge to provide semantically rich road segment embeddings. Finally, analysis of downstream tasks shows that the trajectories generated by RNTrajGen exhibit performance comparable to that of real-world trajectories.

Acknowledgements

This research was funded by the Science and Technology Development Program of Jilin Province (No. 20250203122SF).

References

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[Feng *et al.*, 2020] Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. Learning to simulate human mobility. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3426–3433, 2020.

[Guo *et al.*, 2025] Baoshen Guo, Zhiqing Hong, Junyi Li, Shenhao Wang, and Jinhua Zhao. Leveraging the spatial hierarchy: Coarse-to-fine trajectory generation via cascaded hybrid diffusion. *arXiv preprint arXiv:2507.13366*, 2025.

[Hadsell *et al.*, 2006] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hu *et al.*, 2024] Yujia Hu, Yuntao Du, Zhikun Zhang, Ziquan Fang, Lu Chen, Kai Zheng, and Yunjun Gao. Real-time trajectory synthesis with local differential privacy. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1685–1698. IEEE, 2024.

[Huang *et al.*, 2019] Dou Huang, Xuan Song, Zipei Fan, Renhe Jiang, Ryosuke Shibasaki, Yu Zhang, Haizhong Wang, and Yugo Kato. A variational autoencoder based generative model of urban human mobility. In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 425–430. IEEE, 2019.

[Kapp *et al.*, 2023] Alexandra Kapp, Julia Hansmeyer, and Helena Mihaljević. Generative models for synthetic urban mobility data: A systematic literature review. *ACM Computing Surveys*, 56(4):1–37, 2023.

[Kim *et al.*, 2006] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–13, 2006.

[Li *et al.*, 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[Long *et al.*, 2023] Qingyue Long, Huandong Wang, Tong Li, Lisi Huang, Kun Wang, Qiong Wu, Guangyu Li, Yanping Liang, Li Yu, and Yong Li. Practical synthetic human trajectories generation based on variational point processes. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4561–4571, 2023.

[Ouyang *et al.*, 2018] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. A non-parametric generative model for human trajectories. In *IJCAI*, volume 18, pages 3812–3817, 2018.

[Rao *et al.*, 2025] Xuan Rao, Shuo Shang, Renhe Jiang, Peng Han, and Lisi Chen. Seed: Bridging sequence and diffusion models for road trajectory generation. In *Proceedings of the ACM on Web Conference 2025*, pages 2007–2017, 2025.

[Ren *et al.*, 2021] Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1410–1419, 2021.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Shi *et al.*, 2024] Dingyuan Shi, Yongxin Tong, Zimu Zhou, Ke Xu, Zheng Wang, and Jieping Ye. Graph-constrained diffusion for end-to-end path planning. In *The Twelfth International Conference on Learning Representations*, 2024.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2021] Xingrui Wang, Xinyu Liu, Ziteng Lu, and Hanfang Yang. Large scale gps trajectory generation using map based on two stage gan. *Journal of Data Science*, 19(1):126–141, 2021.

[Wei *et al.*, 2024] Tonglong Wei, Youfang Lin, Shengnan Guo, Yan Lin, Yiheng Huang, Chenyang Xiang, Yuqing Bai, and Huaiyu Wan. Diff-rntraj: A structure-aware diffusion model for road network-constrained trajectory generation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Wen *et al.*, 2021] Haomin Wen, Youfang Lin, Fan Wu, Huaiyu Wan, Shengnan Guo, Lixia Wu, Chao Song, and Yinghui Xu. Package pick-up route prediction via modeling couriers' spatial-temporal behaviors. In *2021*

IEEE 37th International Conference on Data Engineering (ICDE), pages 2141–2146. IEEE, 2021.

[Wen *et al.*, 2022] Haomin Wen, Youfang Lin, Xiaowei Mao, Fan Wu, Yiji Zhao, Haochen Wang, Jianbin Zheng, Lixia Wu, Haoyuan Hu, and Huaiyu Wan. Graph2route: A dynamic spatial-temporal graph neural network for pick-up and delivery route prediction. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4143–4152, 2022.

[Williams and Zipser, 1989] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[Yang *et al.*, 2018] Zhou Yang, Heli Sun, Jianbin Huang, Zhongbin Sun, Hui Xiong, Shaojie Qiao, Ziyu Guan, and Xiaolin Jia. An efficient destination prediction approach based on future trajectory prediction and transition matrix optimization. *IEEE transactions on knowledge and data engineering*, 32(2):203–217, 2018.

[Yuan *et al.*, 2022] Yuan Yuan, Jingtao Ding, Huandong Wang, Depeng Jin, and Yong Li. Activity trajectory generation via modeling spatiotemporal dynamics. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4752–4762, 2022.

[Zhu *et al.*, 2023] Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. Difftraj: Generating gps trajectory with diffusion probabilistic model. *Advances in Neural Information Processing Systems*, 36:65168–65188, 2023.

[Zhu *et al.*, 2024] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4676–4687, 2024.