symmetry

MDPI

*Article*

# Sem-RoadDiff: Road-Aware Diffusion Model with Semantic Guidance for Trajectory Generation

**Yonghua Zhu, Jingxian Cheng *, Juan Zhao and Xiangyu Song**

School of Data Science and Artificial Intelligence, Chang'an University, Xi'an 710018, China; 2024140008@chd.edu.cn (Y.Z.); juan.zhao@chd.edu.cn (J.Z.); xiangyu.song@chd.edu.cn (X.S.)
* Correspondence: chengjx3@126.com

**Abstract**

Trajectory data is valuable for applications such as urban planning, but its public availability is often limited by privacy concerns and data collection costs. While recent diffusion models have shown promise in generative tasks, existing methods rarely integrate personalized conditioning with road network constraints. As a result, they struggle to simultaneously achieve personalized mobility modeling and high road-network spatial validity, resulting in limited trajectory quality. In this paper, we propose Sem-RoadDiff, a symmetry-aware dual-guided diffusion model for personalized and road network-constrained trajectory generation. Specifically, our model incorporates two key components. First, we design a semantic preference guidance mechanism to encode user history into a preference-weighted user embedding using a temperature-scaled softmax. This enables the model to capture user-level mobility patterns without directly using raw trip-level records as generation conditions. Second, we introduce a road-aware mechanism to improve overall spatial validity, employing a spatial validity loss derived from the User Mobility Transition Graph. From a symmetry perspective, Sem-RoadDiff aims to preserve distributional symmetry between real and generated trajectories while respecting the inherent asymmetry of directed road-network transitions. Extensive experiments on the Geolife and Porto datasets demonstrate that our approach improves trajectory distributional fidelity compared with the evaluated baselines and improves road-segment connectivity over the diffusion-based baseline.

**Keywords:** trajectory generation; diffusion models; road network constraints; spatial-temporal data

## 1. Introduction

Trajectory data play an important role in a wide range of applications, including urban planning [1–3], route recommendation [4–6], and traffic management [7]. However, due to privacy concerns [8–10] and high collection costs [10,11], real-world trajectory data are difficult to obtain, resulting in limited public datasets. To address these problems, trajectory generation has emerged as a promising solution for supporting downstream applications without directly releasing real-world mobility records.

In the beginning, research mainly focused on rule-based methods that rely on predefined rules to simulate movement. For example, TimeGeo [12] and EPR [13] models use statistical laws to simulate human mobility. Similarly, simulation software such as SUMO [14] generates vehicle trajectories based on traffic rules. Yet, these methods fail to capture personalized semantics and long-term dependencies. To overcome these limitations, deep learning methods have gained attention. These methods effectively capture

Check for updates

Academic Editor: Shi Cheng

Received: 30 April 2026
Revised: 1 June 2026
Accepted: 11 June 2026
Published: 15 June 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

complex trajectory distributions. Generally, these approaches are divided into three major categories: Variational Autoencoders (VAEs) [15–17], Generative Adversarial Networks (GANs) [18–22], and diffusion models [23–25]. VAEs often generate topologically distorted paths, which fail to reflect real road networks [16]. Similarly, GANs are unstable during training and suffer from mode collapse, leading to poor diversity [21]. Recently, diffusion models have become popular for their superior performance.

Diffusion models operate in two main steps: a forward process that progressively adds noise to the data and a reverse process that removes this noise step by step [26]. However, existing diffusion-based trajectory generation methods still have some limitations. For example, conditional generation methods such as DiffTraj [23] guide trajectory generation by using trajectory-level conditions. However, these approaches rely on specific historical trajectory, making it difficult to capture user-level mobility patterns beyond individual trips. In contrast, road-constrained methods such as Diff-RNTraj [25] integrate road network information, effectively improving the spatial validity of generated trajectories. However, they lack user-level semantic modeling, which often leads to insufficient behavioral diversity and limited personalization. Overall, existing methods typically rely on a single source of guidance—either trajectory-level conditioning or road network constraints. From a symmetry perspective, this challenge can be understood as preserving two complementary properties of mobility data. First, generated trajectories should maintain distributional symmetry with real trajectories, ensuring that macroscopic characteristics such as travel distance, spatial density, and road usage remain statistically consistent. Second, road-network mobility is inherently asymmetric, as transitions between road segments are not necessarily reversible because of directed roads, traffic rules, and user-specific movement preferences. Accordingly, an effective trajectory generation model should preserve distributional symmetry between real and generated trajectories while respecting the asymmetric road-network transitions inherent in road-network-constrained mobility. As a result, it is difficult to achieve both personalized mobility modeling and road-network consistency generation within a unified framework. This trade-off is a fundamental challenge in high-quality trajectory generation. Existing diffusion-based trajectory generators have not yet fully addressed the joint modeling of explicit user-history preferences and road-network feasibility. Personalized trajectory generation requires capturing user-level mobility habits from historical road-segment visitation patterns, whereas spatial validity requires generated trajectories to follow feasible road-segment transitions. This leaves a key research gap: how to design a diffusion framework that jointly incorporates user-history-based semantic guidance and road-aware constraints during trajectory generation.

To address this challenge, we introduce Sem-RoadDiff, a dual-guided diffusion model that combines semantic preference guidance with road-aware constraints. On the semantic side, we summarize each user's historical behavior into a preference-weighted embedding and inject it into the denoising process to guide user-level mobility generation.

On the constraint side, we facilitate topologically valid transitions via a spatial validity loss and a post-generation alignment strategy to penalize invalid road transitions and ensure overall spatial validity. Consequently, Sem-RoadDiff generates road-constrained trajectories that are both personalized and spatially valid. Unlike DiffTraj-like methods, which mainly rely on trip-level conditions, Sem-RoadDiff constructs user-level semantic preference embeddings from historical road-segment visitation patterns. In contrast to Diff-RNTraj-like methods, which emphasize road-network constraints but provide limited user-level semantic conditioning, Sem-RoadDiff integrates semantic preference guidance and road-aware constraints within a unified diffusion generation process.

The principal contributions of this work are summarized as follows:

- We propose Sem-RoadDiff, a symmetry-aware dual-guided diffusion model for personalized and road network-constrained trajectory generation. The proposed model integrates semantic preference guidance with road-aware constraints to improve both behavioral specificity and spatial validity by preserving distributional symmetry between real and generated trajectories and respecting asymmetric road-network transitions.
- We design a dual-guidance mechanism for trajectory generation. On the semantic side, we encode user history into a preference-weighted embedding and inject it into the denoising process to guide user-level mobility modelling without using raw trip-level conditions. On the constraint side, we facilitate topologically valid transitions via a spatial validity loss and a post-generation alignment strategy to penalize invalid road transitions and ensure overall spatial validity.
- We conduct extensive experiments on the Geolife and Porto datasets. The results show that Sem-RoadDiff consistently outperforms the strong baseline, achieving an average improvement of 12.83% across JSD metrics, while further enhancing the road segment connectivity by an absolute average of 1.91%.

## 2. Related Work

### *2.1. Trajectory Generation*

Trajectory generation aims to generate realistic trajectories that preserve the statistical patterns of real-world movement. These synthetic trajectories support applications such as traffic simulation, demand forecasting, and privacy-preserving data publishing [27–30]. Early approaches relied on rule-based and statistical modeling principles. For example, TimeGeo [12] infers anchor points from population-level spatiotemporal patterns. Similarly, EPR variants [13] model exploration behavior through distance decay. However, these rule-based techniques have clear limitations. They struggle to capture personalized mobility patterns and long-term behavioral dependencies. Consequently, research has transitioned to data-driven generative models. Deep generative models directly learn trajectory distributions from data. VAE-based methods encode trajectories into latent spaces but often produce blurry reconstructions that distort spatial topology. This limitation persists even in more complex architectures. For instance, MTG [15] employs a bi-directional encoder but still produces overly averaged outputs. Similarly, VOLUNTEER [17] decouples travel and dwell time yet remains limited by the mean-seeking property of the VAE objective. In contrast, GAN-based approaches [18–22] achieve sharper samples through adversarial training. However, GANs face the persistent challenge of mode collapse. The generator often ignores rare patterns to fool the discriminator, leading to low diversity and poor long-term consistency.

The limitations of VAEs and GANs have motivated the recent adoption of diffusion models. These models are known for superior sample quality and training stability [31]. DiffTraj [23] uses diffusion to generate trajectories conditioned on trip attributes such as origin-destination pairs. To incorporate structural priors, Diff-RNTraj [25] integrates road-network constraints to ensure spatial validity. However, a key challenge remains. Diff-RNTraj avoids detailed instance-level conditioning and instead emphasizes road-constrained generation, which improves spatial validity but limits controllability and personalization. Currently, many conditional approaches improve personalization by relying on relatively fine-grained attributes, which increases dependence on detailed trajectory-related information. This limitation highlights the open challenge of designing effective conditioning strategies that support personalization without relying on fine-grained raw trip-level information.

### 2.2. Diffusion Probabilistic Model

Diffusion models are a powerful class of generative models. Sohl-Dickstein et al. first introduced the framework [32], and Ho et al. significantly advanced it with Denoising Diffusion Probabilistic Models (DDPM) [26]. The framework consists of two sequential Markov processes. First, the forward process gradually adds Gaussian noise to the data until its distribution approaches a Gaussian distribution. Second, the reverse process iteratively removes this noise to reconstruct the original distribution [33,34].

Recent research has focused on improving sample quality and efficiency. For instance, Song et al. [34] proposed non-Markovian processes to reduce sampling steps. Similarly, Nichol et al. [35] learned variances in the reverse process to further accelerate sampling. Dhariwal et al. [36] also optimized the network architecture to achieve superior generation quality.

Diffusion models have succeeded in various fields, such as computer vision [37,38], NLP [39,40], and traffic forecasting [41]. Generally, these models are designed for either continuous data or discrete data. However, spatiotemporal trajectories are more complex because they involve both continuous movement patterns and discrete road-network structure. Existing diffusion-based methods only partially address these characteristics [23,42,43]. For example, while Diff-RNTraj [25] improves spatial validity through road network constraints, it provides limited support for personalized conditioning. Therefore, achieving controllable, realistic, and road network-constrained trajectory generation remains an open challenge.

## 3. Preliminaries

In this section, we formally define the key concepts used in our framework and formulate the problem of personalized road network-constrained trajectory generation.

**Definition 1** (Road Network). *A road network is defined as a directed graph $G = (V, E)$, where:*

- *$V$ is the set of intersections (nodes), each associated with geographic coordinates $(lon, lat)$;*
- *$E$ is the set of road segments (edges), where each segment $e \in E$ has:*
  - *$e.start$: start intersection;*
  - *$e.end$: end intersection;*
  - *$e.length$: length of the segment (in meters).*

**Definition 2** (Road Network-Constrained Trajectory, i.e., RoadTraj). *A RoadTraj of length $L$ is a sequence $\tau = \langle q_1, q_2, \ldots, q_L \rangle$, where each point $q_i = (e_i, r_i, t_i)$ is defined by:*

- *$e_i \in E$: The road segment where $q_i$ is located;*
- *$r_i \in [0, 1]$: The moving ratio, representing the fractional distance traveled along $e_i$;*
- *$t_i$: Timestamp.*

*The moving ratio $r_i$ is computed as:*

$$r_i = \begin{cases} \frac{dis(e_i.start, q_i)}{e_i.length}, & \text{if } e_i \neq e_{i-1} \text{ or } i = 1, \\ \frac{dis(q_{i-1}, q_i)}{e_i.length - dis(e_i.start, q_{i-1})}, & \text{otherwise}, \end{cases} \tag{1}$$

*where $dis(a, b)$ calculates the network distance between $a$ and $b$.*

**Definition 3** (User Trajectory Data). *For $n$ users, the dataset is denoted as $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$, where each user $u$ has:*

$$\mathcal{T}_u = \{\tau_1, \ldots, \tau_{m_u}\}. \tag{2}$$

*Each trajectory $\tau = [p_1, \ldots, p_l]$ consists of points $p = (loc, t)$ with:*

- $loc = (lon, lat)$: *Geographic coordinates;*
- $t$: *Timestamp.*

**Definition 4** (Personalized Road Network-Constrained Trajectory Generation)**.** *Given a road network G and dataset $\mathcal{T}$, the goal is to learn a generative model $\mathcal{M}_\theta$ that generates synthetic trajectories $\hat{\tau} \sim \mathcal{M}_\theta(\mathcal{T}_u, G)$. The generated road-constrained trajectories should preserve user-level mobility semantics while improving road-network spatial validity. This work focuses on the spatial, road-constrained component of trajectories, namely the road-segment sequence and the moving ratio along each segment. Timestamps are used for preprocessing and trajectory filtering, but the current generative model does not explicitly generate temporal intervals, travel speeds, or acceleration. Therefore, the lack of temporal dynamics is acknowledged as an important limitation and left for future extension. From a symmetry perspective, this task involves both distributional symmetry and transition asymmetry. At the distributional level, generated trajectories should remain statistically consistent with real trajectories in terms of travel distance, segment distance, spatial density, and road usage. At the structural level, road-network-constrained mobility is inherently asymmetric because the road network is directed and the transition from road segment $r_i$ to $r_j$ may not be equivalent to the reverse transition from $r_j$ to $r_i$. Therefore, Sem-RoadDiff aims to preserve distributional symmetry between real and generated trajectories while respecting the asymmetric road-network transitions encoded by the User Mobility Transition Graph.*

## 4. Methodology

### *4.1. Trajectory Embedding Module*

We convert the hybrid RoadTraj representation into a continuous space to enable the diffusion model to generate road-constrained trajectories. The main challenge is to learn embeddings for discrete road segments that accurately reflect real-world travel patterns. To learn effective representations, we must model the relationships between segments. A natural choice is to use the road-network structure. However, physical connections often differ from actual driving patterns. For example, connected roads may be unreachable due to traffic rules like one-way streets. Moreover, transition frequencies between road segments are highly uneven. For example, transitions between major roads occur much more frequently than transitions to minor roads. Therefore, we define segment relationships based on actual user behavior, rather than just the network structure.

We construct a structured graph representation from the raw trajectory data. We call this the User Mobility Transition Graph (UMTG), denoted as $\mathcal{G}_M = (\mathcal{R}, \mathcal{E}_M)$. In this graph, nodes ($\mathcal{R}$) represent road segments, and edges ($\mathcal{E}_M$) represent movement between them. We add an edge $e_{ij}$ when a user moves directly from segment $r_i$ to $r_j$ in a trajectory. The weight $w_{ij}$ counts how often this transition occurs across the entire dataset:

$$w_{ij} = \sum_{\tau \in \mathcal{T}} \sum_{k=1}^{|\tau|-1} \mathbb{1}\{\tau[k].e = r_i \wedge \tau[k+1].e = r_j\}, \tag{3}$$

here, $\tau[k].e$ is the segment at step $k$, $|\tau|$ is the total length of the trajectory, and $\mathbb{1}\{\cdot\}$ is the indicator function that equals 1 if the condition is true and 0 otherwise.

The UMTG is constructed as a directed weighted graph because transitions between road segments are direction-sensitive. The edge weight $w_{ij}$ denotes the pooled number of observed transitions from segment $r_i$ to segment $r_j$ across all users and training trajectories. Pooled transition counts, rather than user-specific transition graphs, are used to obtain a denser and more stable transition structure for modeling empirical mobility-transition feasibility. In the implementation, edge weights are used for Node2Vec sampling to learn road-segment embeddings, whereas the spatial validity loss uses only the binary

existence of an edge, i.e., whether $w_{ij} > 0$. Self-transitions from a road segment to itself are retained when consecutive trajectory points remain on the same road segment because they correspond to valid continuous movement along that segment. Rare transitions are also retained after preprocessing, and no additional frequency threshold is applied to remove low-count but valid transitions.

This construction naturally captures the asymmetric road-network transitions, where the transition probability or frequency from $r_i$ to $r_j$ can differ from that from $r_j$ to $r_i$.

By capturing real-world mobility patterns, this graph facilitates the learning of a segment embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{R}| \times D}$ via Node2Vec. These embeddings encode both the structure and the usage of road segments.

To vectorize a trajectory $\tau$, we process each trajectory point $q_t = (e_t, r_t)$ by retrieving its corresponding segment embedding $\mathbf{x}_t^e$ from the matrix $\mathbf{E}$. Simultaneously, we normalize the continuous moving ratio $r_t$ to a range of $[-1, 1]$, denoted as $r'_t$. Finally, we concatenate these embedding vectors and normalized ratios to construct the unified RoadTraj representation $\mathbf{X} \in \mathbb{R}^{T \times (D+1)}$:

$$\mathbf{X} = [\mathbf{x}_1^e \oplus r'_1, \mathbf{x}_2^e \oplus r'_2, \ldots, \mathbf{x}_T^e \oplus r'_T]. \tag{4}$$

### *4.2. Conditional Information Encoding*

We design a conditional encoder to facilitate personalized road network-constrained trajectory generation. This module captures how specific users use road segments. Specifically, it distills historical data into a compact user preference embedding to guide the diffusion model.

#### 4.2.1. User Road Segment Preference Statistics

We analyze a user's historical trajectories $\mathcal{T}_u = \{\tau_1, \ldots, \tau_M\}$. First, we calculate the visitation frequency $f(e_j)$ for each road segment $e_j$:

$$f(e_j) = \sum_{i=1}^{M} \mathbb{I}(e_j \in \tau_i), \tag{5}$$

here, $\mathbb{I}(\cdot)$ equals 1 if segment $e_j$ appears in trajectory $\tau_i$, and 0 otherwise. This gives us the complete frequency distribution $\mathcal{F}_u$.

To filter noise, we keep only the top-$K$ most visited segments. We sort the segments by frequency and define the preferred set $\mathcal{E}_u^{\text{top-}K} = \{e^{(1)}, \ldots, e^{(K)}\}$, empirically setting $K = 100$ to balance coverage and computational efficiency. Top-K visitation frequency is used as a lightweight and interpretable proxy for long-term spatial mobility preference. In urban mobility, frequently visited road segments often correspond to stable activity regions, habitual commuting corridors, or user-specific travel areas. Top-K selection therefore reduces the influence of occasional or noisy segments and focuses the condition vector on the most representative parts of a user's mobility history. This frequency-based representation does not explicitly model temporal order, time-of-day patterns, destination intent, or trip context; sequence-aware and context-aware preference modeling will be explored in future work.

#### 4.2.2. Preference-Weighted Condition Embedding

To construct the user preference vector, we employ a weighted aggregation strategy. We start by transforming the raw visitation frequencies of the top-$K$ segments into normalized weights. This is achieved via a temperature-scaled Softmax function:

$$w_j = \frac{\exp(f(e^{(j)})/\tau)}{\sum_{k=1}^{K} \exp(f(e^{(k)})/\tau)}. \tag{6}$$

In this calculation, we set the temperature parameter $\tau = 0.5$. This specific value sharpens the distribution, ensuring that the model focuses primarily on the most frequently visited segments while still acknowledging less common ones.

Using these normalized weights, we generate the final preference embedding $\mathbf{c}$ by computing a weighted sum of the corresponding segment vectors:

$$\mathbf{c} = \sum_{j=1}^{K} w_j \cdot \mathbf{E}[e^{(j)}], \tag{7}$$

here, $\mathbf{E}[e^{(j)}]$ represents the pre-trained embedding for a specific road segment. The resulting vector $\mathbf{c}$ effectively serves as a compact summary of the user's spatial habits, highlighting their preferred routes in the continuous space.

### *4.3. Conditional Diffusion Model*

Building upon the vectorized RoadTraj representation and user preference encoding, we now present our conditional diffusion model for road network-constrained trajectory generation. The model learns a conditional distribution $p_\theta(\mathbf{X}|\mathbf{c})$ that approximates the true distribution of trajectories $q(\mathbf{X}|\mathbf{c})$ given user preference $\mathbf{c}$.

#### 4.3.1. Conditional Forward Process

The forward diffusion process gradually adds Gaussian noise to a clean trajectory, transforming it into pure noise over $N$ steps. This process remains largely similar to the unconditional case, as it does not depend on the condition during noise addition.

Given an original vectorized RoadTraj $\mathbf{X}_0 \in \mathbb{R}^{T\times(D+1)}$ and its associated user preference $\mathbf{c} \in \mathbb{R}^{d_c}$, the forward process is defined as a Markov chain:

$$q(\mathbf{X}_N|\mathbf{X}_0) = \prod_{n=1}^{N} q(\mathbf{X}_n|\mathbf{X}_{n-1}), \tag{8}$$

where each transition follows a Gaussian distribution:

$$q(\mathbf{X}_n|\mathbf{X}_{n-1}) = \mathcal{N}(\mathbf{X}_n; \sqrt{1-\beta_n}\mathbf{X}_{n-1}, \beta_n\mathbf{I}), \tag{9}$$

here, $\beta_n \in (0,1)$ is a pre-defined noise schedule that controls the amount of noise added at each step. The values $\{\beta_1, \beta_2, \ldots, \beta_N\}$ gradually increase from a small value (e.g., $\beta_1 = 0.0001$) to a larger value (e.g., $\beta_N = 0.02$).

By leveraging the properties of the Gaussian diffusion process, we can directly sample $\mathbf{X}_n$ from $\mathbf{X}_0$ without iterating through all intermediate steps:

$$q(\mathbf{X}_n|\mathbf{X}_0) = \mathcal{N}(\mathbf{X}_n; \sqrt{\bar{\alpha}_n}\mathbf{X}_0, (1-\bar{\alpha}_n)\mathbf{I}), \tag{10}$$

where $\alpha_n = 1 - \beta_n$ and $\bar{\alpha}_n = \prod_{i=1}^{n} \alpha_i$.

Using the reparameterization trick, this sampling can be efficiently implemented as:

$$\mathbf{X}_n = \sqrt{\bar{\alpha}_n}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_n}\boldsymbol{\epsilon}, \tag{11}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is noise sampled from a standard Gaussian distribution.

After $N$ diffusion steps with a properly designed noise schedule, the terminal distribution of the perturbed trajectory approaches a standard Gaussian:

$$\mathbf{X}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{12}$$

This property is crucial as it allows us to start the generation process from pure noise.

#### 4.3.2. Conditional Reverse Process

The reverse process aims to recover a clean trajectory from noise by iteratively denoising over $N$ steps. Unlike the forward process, the reverse process is conditioned on the user preference vector $\mathbf{c}$, enabling personalized road network-constrained trajectory.

Each step of the reverse process is modelled as a conditional Gaussian distribution:

$$p_\theta(\mathbf{X}_{n-1} \mid \mathbf{X}_n, \mathbf{c}) = \mathcal{N}(\mathbf{X}_{n-1}; \boldsymbol{\mu}_\theta(\mathbf{X}_n, n, \mathbf{c}), \boldsymbol{\Sigma}_\theta(\mathbf{X}_n, n, \mathbf{c})), \tag{13}$$

where $\boldsymbol{\mu}_\theta(\cdot)$ and $\boldsymbol{\Sigma}_\theta(\cdot)$ are the mean and covariance predicted by a neural network with parameters $\theta$.

Following the DDPM framework [26], we parameterize the model to predict the noise $\boldsymbol{\epsilon}_\theta$ rather than directly predicting the mean $\boldsymbol{\mu}_\theta$. The mean is then derived as:

$$\boldsymbol{\mu}_\theta(\mathbf{X}_n, n, \mathbf{c}) = \frac{1}{\sqrt{\alpha_n}}\left(\mathbf{X}_n - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\boldsymbol{\epsilon}_\theta(\mathbf{X}_n, n, \mathbf{c})\right), \tag{14}$$

with the variance fixed to:

$$\boldsymbol{\Sigma}_\theta(\mathbf{X}_n, n, \mathbf{c}) = \frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha}_n}\beta_n \mathbf{I}. \tag{15}$$

The conditional denoising network $\boldsymbol{\epsilon}_\theta(\mathbf{X}_n, n, \mathbf{c})$ estimates the noise based on three inputs: the current noisy RoadTraj representation $\mathbf{X}_n$, the diffusion step $n$, and the user preference vector $\mathbf{c}$.

The sampling procedure begins with pure noise $\mathbf{X}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively applies the denoising update:

$$\mathbf{X}_{n-1} = \frac{1}{\sqrt{\alpha_n}}\left(\mathbf{X}_n - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\boldsymbol{\epsilon}_\theta(\mathbf{X}_n, n, \mathbf{c})\right) + \sigma_n \mathbf{z}, \tag{16}$$

where $\sigma_n = \sqrt{\frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha}_n}\beta_n}$, and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $n > 1$ while $\mathbf{z} = \mathbf{0}$ for $n = 1$.

The complete conditional generation process is defined by:

$$p_\theta(\mathbf{X}_0 | \mathbf{X}_N, \mathbf{c}) = \prod_{n=N}^{1} p_\theta(\mathbf{X}_{n-1} | \mathbf{X}_n, \mathbf{c}). \tag{17}$$

This formulation enables generation of valid road-constrained trajectories $\hat{\mathbf{X}}_0$ from random noise conditioned on user preferences $\mathbf{c}$.

#### 4.3.3. Conditional Denoising Architecture

The conditional denoising architecture $\boldsymbol{\epsilon}_\theta(\mathbf{X}_n, n, \mathbf{c})$ accepts three inputs: the corrupted vectorized RoadTraj $\mathbf{X}_n$, the diffusion timestep $n$, and the user preference embedding $\mathbf{c}$. We adopt a Residual Dilated Convolution architecture enhanced with a unified conditioning mechanism.

As illustrated in Figure 1, we first construct a unified conditional embedding to guide the road network-constrained trajectory generation. The diffusion timestep $n$ is mapped to a high-dimensional representation using a sinusoidal encoding strategy. The step encoding $\mathrm{SE}[n] \in \mathbb{R}^F$ is defined as:

$$\mathrm{SE}[n] = [\sin(\omega_0 \cdot n), \ldots, \sin(\omega_{\frac{F}{2}-1} \cdot n), \cos(\omega_0 \cdot n), \ldots, \cos(\omega_{\frac{F}{2}-1} \cdot n)], \tag{18}$$

where $\{\omega_i\}$ constitutes a geometric progression of frequencies. This encoding is processed through a two-layer MLP with SiLU nonlinearity to obtain the time embedding $\mathbf{e}_t$:

$$\mathbf{e}_t = \mathrm{FC}_2(\mathrm{SiLU}(\mathrm{FC}_1(\mathrm{SE}[n]))) \in \mathbb{R}^F, \tag{19}$$

In parallel, the user preference embedding $\mathbf{c} \in \mathbb{R}^{d_c}$ is projected into the same latent space via a separate MLP:

$$\mathbf{e}_c = \mathrm{FC}_2(\mathrm{SiLU}(\mathrm{FC}_1(\mathbf{c}))) \in \mathbb{R}^F, \tag{20}$$

We then fuse these embeddings by concatenation. A subsequent projection layer produces the unified conditional embedding:

$$\mathbf{e}_{\mathrm{cond}} = \mathrm{MLP}_{\mathrm{fusion}}([\mathbf{e}_t; \mathbf{e}_c]) \in \mathbb{R}^F, \tag{21}$$

This unified embedding encapsulates both the diffusion-step state and the personalized user preference signal.

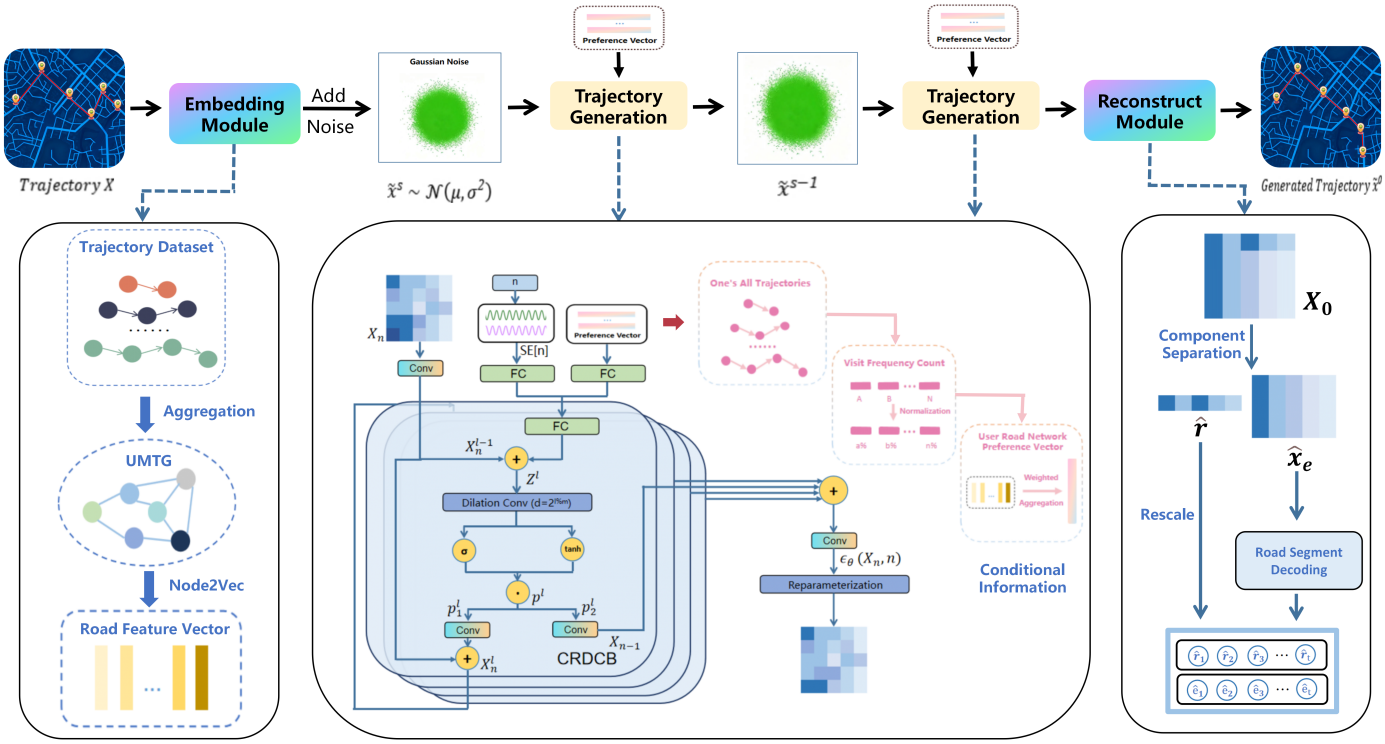



**Figure 1.** The architecture of Sem-RoadDiff, which consists of a RoadTraj vectorization module, a conditional diffusion module with semantic guidance, and a RoadTraj decoder module.

The backbone network consists of a stack of $L$ Conditional Residual Dilated Convolution Blocks. To ensure strong conditional guidance, we inject the unified conditional embedding at two stages. First, at the network entry, the noisy trajectory $\mathbf{X}_n$ is projected and combined with the conditional signal:

$$\mathbf{H}_0 = \mathrm{Conv1D}(\mathbf{X}_n) + \mathrm{Linear}(\mathbf{e}_{\mathrm{cond}}) \in \mathbb{R}^{T \times C}, \tag{22}$$

where the addition involves broadcasting the signal across the temporal dimension.

Subsequently, the features are processed through the residual blocks. In each $l$-th block, the unified conditional embedding is re-injected. Let $\mathbf{H}_{l-1}$ denote the block input. The conditional embedding is projected and added to the hidden states:

$$\mathbf{Z}^l = \mathbf{H}_{l-1} + \mathrm{Proj}^l(\mathbf{e}_{\mathrm{cond}}) \in \mathbb{R}^{T\times C}, \tag{23}$$

The modulated features $\mathbf{Z}^l$ pass through a dilated convolution with dilation factor $d = 2^{l \bmod m}$:

$$\mathbf{H}^l_{\mathrm{dilated}} = \mathrm{DilatedConv1D}(\mathbf{Z}^l, d) \in \mathbb{R}^{T\times 2C}, \tag{24}$$

The output is split into two parts for the gated activation mechanism:

$$\mathbf{H}^l_{\mathrm{gate}}, \mathbf{H}^l_{\mathrm{filter}} = \mathrm{split}(\mathbf{H}^l_{\mathrm{dilated}}) \in \mathbb{R}^{T\times C}, \tag{25}$$

$$\mathbf{P}^l = \sigma(\mathbf{H}^l_{\mathrm{gate}}) \odot \tanh(\mathbf{H}^l_{\mathrm{filter}}) \in \mathbb{R}^{T\times C}, \tag{26}$$

The activated features are then processed to generate residual and skip connections:

$$\mathbf{R}^l = \mathrm{Conv1D}(\mathbf{P}^l), \quad \mathbf{S}^l = \mathrm{Conv1D}(\mathbf{P}^l), \tag{27}$$

The output of the $l$-th block is computed as:

$$\mathbf{H}_l = \mathbf{H}_{l-1} + \mathbf{R}^l, \tag{28}$$

Finally, we aggregate the skip connections from all blocks. The sum is normalized by $\sqrt{L}$ and projected via a final $1 \times 1$ convolution layer to yield the predicted noise:

$$\boldsymbol{\epsilon}_\theta(\mathbf{X}_n, n, \mathbf{c}) = \mathrm{Conv}_{1\times 1}\left(\frac{1}{\sqrt{L}} \sum_{l=1}^{L} \mathbf{S}^l\right). \tag{29}$$

### *4.4. RoadTraj Reconstruction Module*

We design a reconstruction module to transform the generated vectorized RoadTraj $\hat{\mathbf{X}}_0 \in \mathbb{R}^{T\times(D+1)}$ back to its hybrid format. This format consists of discrete road segments $\hat{e}$ and continuous moving ratios $\hat{r}$. First, the module decomposes $\hat{\mathbf{X}}_0$ along the channel dimension to extract the road segment embeddings and the moving ratio components.

We separate the generated representation into two parts:

$$\begin{aligned} \hat{\mathbf{X}}_e &= \hat{\mathbf{X}}_0[:, 0:D] \in \mathbb{R}^{T\times D}, \\ \hat{r} &= \frac{\hat{\mathbf{X}}_0[:, D+1] + 1}{2} \in \mathbb{R}^{T\times 1}, \end{aligned} \tag{30}$$

where $\hat{\mathbf{X}}_e$ is the generated road segment embedding, and $\hat{r}$ is the moving ratio. We use the inverse transformation $(\cdot + 1)/2$ to rescale the moving ratio from $[-1, 1]$ back to the valid range $[0, 1]$.

To identify the discrete road segment $\hat{e}$, we use the pre-trained embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{R}|\times D}$. Specifically, we compute the cosine similarity between the generated embeddings $\hat{\mathbf{X}}_e$ and the matrix $\mathbf{E}$:

$$\mathbf{S} = \frac{\hat{\mathbf{X}}_e \cdot \mathbf{E}^\top}{\|\hat{\mathbf{X}}_e\|_2 \cdot \|\mathbf{E}\|_2} \in \mathbb{R}^{T\times|\mathcal{R}|}, \tag{31}$$

where $\mathbf{S}[t, i]$ is the similarity score between the $t$-th trajectory point and the $i$-th road segment. The UMTG is stored as a sparse adjacency dictionary, whose construction requires one scan over all map-matched trajectories and is linear in the total number of observed

trajectory transitions. Its memory cost is linear in the number of observed UMTG edges. For reconstruction, Equation (31) has complexity $O(BT|R|D)$ for batch size $B$, sequence length $T$, road-segment number $|R|$, and embedding dimension $D$, and is implemented by GPU-based batched matrix multiplication. Since $|R|$ is 19,906 for Geolife and 10,870 for Porto, this decoding step is manageable in our experiments. For larger road networks, approximate nearest-neighbor search or transition-based candidate pruning can be used to improve efficiency. Then, we select the road segment with the highest similarity score for each point:

$$\hat{e}_t = \arg \max_{i \in \{1,\ldots,|\mathcal{R}|\}} \mathbf{S}[t, i], \tag{32}$$

This yields the complete road segment sequence $\{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_T\}$. Finally, we map the segments and ratios to the road network to determine the exact GPS coordinates. This completes the RoadTraj generation process. Next, we describe the training methodology for the model parameters $\theta$.

### *4.5. Model Training*

We design a loss function to train the parameters $\theta$. This function optimizes three aspects: noise estimation, structural quality, and spatial validity.

First, we use a noise prediction loss. This measures the difference between the actual noise and the prediction. Given a clean sample $\mathbf{X}_0$, we add Gaussian noise $\boldsymbol{\epsilon}$ at a random timestep $n$ to get the noisy sample $\mathbf{X}_n$. The loss is:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}\Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{X}_n, n, \mathbf{c})\|^2\Big], \tag{33}$$

here, $\mathbf{c}$ represents the user preference. This loss trains the model to estimate noise based on user conditions.

However, noise prediction alone implies limited quality. Therefore, we add a structural loss to measure the reconstruction error. We derive the predicted clean sample $\hat{\mathbf{X}}_0$ from the noisy observation. The structural loss is:

$$\mathcal{L}_{\text{struct}} = \mathbb{E}\Big[\|\mathbf{X}_0 - \hat{\mathbf{X}}_0\|^2\Big], \tag{34}$$

where $\hat{\mathbf{X}}_0 = (\mathbf{X}_n - \sqrt{1-\bar{\alpha}_n}\boldsymbol{\epsilon}_\theta(\mathbf{X}_n, n, \mathbf{c}))/\sqrt{\bar{\alpha}_n}$. This ensures the generated trajectories preserve the structure of real data.

We must also ensure spatial validity. Consecutive segments must connect in the real road network. To enforce this, we use a spatial validity loss. During training, we reconstruct the segment sequence $\{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_T\}$ from $\hat{\mathbf{X}}_0$. Then, we use the UMTG $G_M = (R, E_M)$ to check the connectivity:

$$\mathcal{L}_{\text{spatial}} = \sum_{t=1}^{T-1} (1 - \mathbb{I}\{(\hat{e}_t, \hat{e}_{t+1}) \in E_M\}), \tag{35}$$

here, $\mathbb{I}\{(\hat{e}_t, \hat{e}_{t+1}) \in E_M\}$ is 1 if an edge exists between $\hat{e}_t$ and $\hat{e}_{t+1}$, and 0 otherwise. This penalizes disconnected transitions without requiring post-processing. This loss therefore helps preserve the asymmetric road-network transitions of real road-network mobility, rather than enforcing only undirected or geometrically symmetric connectivity. The total loss combines these three components equally:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{noise}} + \mathcal{L}_{\text{struct}} + \mathcal{L}_{\text{spatial}}. \tag{36}$$

## 5. Experiments

### *5.1. Datasets*

We evaluate our model on two real-world benchmarks:Geolife and Porto. Geolife captures human mobility in Beijing, while Porto contains taxi trajectories from Portugal. We align GPS records with road networks from OpenStreetMap. We use the TrackIt package to project coordinates onto road segments using an HMM-based algorithm [44].

After map matching, we preprocess the data to ensure quality. First, we remove noisy data, such as invalid coordinates and stationary points moving less than 10 m. Second, we filter trajectories by duration. We keep trips between 5–180 min for Geolife and 5–30 min for Porto. Finally, we remove users with fewer than five trajectories. We remap road segment IDs to a contiguous index for efficiency. Table 1 summarizes the dataset statistics.

**Table 1.** Dataset description.

| Metric | Beijing (Geolife) | Porto (Taxi) |
|---|---|---|
| # Users | 118 | 441 |
| # Trajectories | 10,800 | 1,391,772 |
| # Road Segments | 19,906 | 10,870 |
| # GPS Points | 5,873,981 | 99,233,790 |
| Avg. Points/Traj. | 544 | 71.3 |
| Avg. Traj./User | 91.5 | 3156.0 |
| Spatial Coverage | 39.4–41.0° N<br>115.7–117.4° E | 41.0–41.3° N<br>8.7–8.5° W |

### *5.2. Evaluation Metrics*

Our objective is to synthesize trajectory datasets that preserve the statistical properties of real-world data for downstream trajectory mining applications.

Following the common distribution-matching protocol in synthetic trajectory generation, the evaluation compares generated trajectories with the empirical distribution of each benchmark dataset. This setting assesses dataset-level distributional fidelity and spatial validity, rather than forecasting unseen future trips or evaluating cold-start generalization to unseen users.

We evaluate the quality of generated trajectories by measuring the distributional similarity between synthetic and real datasets from a macroscopic perspective. Following prior work, we employ the Jensen-Shannon Divergence (JSD) as our primary similarity measure, which quantifies the divergence between two probability distributions $p$ and $q$ as:

$$\mathrm{JSD}(p,q) = \frac{1}{2}\mathrm{KL}\left(p \,\|\, \frac{p+q}{2}\right) + \frac{1}{2}\mathrm{KL}\left(q \,\|\, \frac{p+q}{2}\right), \tag{37}$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence. Lower JSD values indicate higher distributional similarity. We compute JSD across the following trajectory characteristics:

- Total Distance (JSD-TD): Compares the distribution of total trajectory lengths. This ensures the generated travel distances represent reality;
- Segment Distance (JSD-SD): Checks the distance between consecutive points. This captures the granularity and local movement patterns;
- Grid Point Density (JSD-GPD): Evaluates the spatial spread of points. We divide the map into 50 m $\times$ 50 m grids to compare point density in each cell;
- Road Segment (JSD-RS): Measures road usage distribution. It validates whether the model learns realistic road patterns.

We also use the Road Segment Connectivity (RSC) metric to check spatial validity. RSC measures if consecutive points are connected by valid roads. To define connectivity,

we use the User Mobility Transition Graph. This graph captures real human movement patterns. For a generated RoadTraj $\hat{\tau} = \langle \hat{q}_1, \ldots, \hat{q}_T \rangle$, RSC is computed as:

$$\text{RSC} = \frac{\sum_{t=1}^{T-1} \mathbb{1}\{G_M(\hat{q}_t.e, \hat{q}_{t+1}.e)\}}{T-1} \times 100\%, \tag{38}$$

where $\mathbb{1}\{\cdot\}$ is an indicator function that equals 1 if consecutive road segments $\hat{q}_t.e$ and $\hat{q}_{t+1}.e$ are connected in $G_M$, and 0 otherwise. Higher RSC values indicate greater spatial validity and realism in the generated trajectories. From the symmetry perspective, the JSD-based metrics evaluate whether the generated trajectories maintain distributional symmetry with real trajectories, while RSC evaluates whether the asymmetric road-network transitions are preserved. These metrics evaluate complementary distribution-level properties of generated trajectories, including travel distance, local movement granularity, spatial density, road-segment usage, and topological feasibility. However, they do not fully capture individual-level trajectory semantics, such as destination intent, temporal context, or sequence-level behavioral meaning. The reported results should therefore be interpreted as evidence of statistical fidelity and spatial validity, rather than complete semantic correctness.

### *5.3. Baselines*

To evaluate the performance of our proposed approach, we compare it against the following baseline methods:

- Random Walk on the Road Network (RWRN): This baseline implements a straightforward rule-based generation strategy. It begins by selecting an arbitrary location as the starting point, then iteratively extends the trajectory by randomly choosing the next road segment and travel speed until reaching the desired trajectory length;
- Markov [45]: This approach models trajectory generation as a Markovian process over road segments. It learns first-order transition probabilities between road segments from historical data and generates new trajectories by sequentially sampling the next road segment conditioned on the current one;
- Diff-RNTraj [25]: This represents the current state-of-the-art in road network-constrained trajectory generation. The model leverages a diffusion-based generative framework that iteratively denoises sampled Gaussian noise into spatially valid trajectories. Key components include pre-learned embeddings for road segments and a specially designed loss function that penalizes invalid road transitions during training.

We selected RWRN, Markov, and Diff-RNTraj because they are directly comparable road-network-constrained baselines. All methods use the same map-matched road network, preprocessing pipeline, and evaluation metrics. Although VAE/GAN-based trajectory generators are important, many of them operate on GPS coordinates or grids rather than road-segment-constrained RoadTraj sequences. Adapting them to this setting is therefore left for future work.

### *5.4. Implementation Details*

For map matching, we use TrackIt with an HMM-based algorithm, where the candidate search buffer is set to 200 m, top-$k$ is set to 10, the GPS noise standard deviation $\sigma$ is set to 50 m, and $\beta = 10$. The most likely road-segment sequence is decoded by Viterbi-style dynamic programming, and unmatched or invalid GPS observations are removed before road segment IDs are remapped to contiguous indices.

We implement our conditional diffusion model using PyTorch 2.4.1 with the Adam optimizer for 30 epochs and a batch size of 256. The learning rate is set to $2 \times 10^{-4}$ with a weight decay of $1 \times 10^{-4}$. For the denoising network, we set the channel dimension

$C = 512$, the positional encoding dimension $F = 512$, and the kernel size to 3 in dilation convolution with padding. We adopt a quadratic schedule for the variance schedule:

$$\beta_n = \left( \frac{N-n}{N-1} \sqrt{\beta_1} + \frac{n-1}{N-1} \sqrt{\beta_N} \right)^2, \tag{39}$$

with the minimum noise level $\beta_1 = 0.0001$ and the maximum noise level $\beta_N = 0.02$. We search the number of diffusion steps $N$ and residual dilation convolution layers $L$ within the parameter space ($N \in \{100, 200, 500, 1000\}$, $L \in \{4, 8, 10, 15, 20\}$).

To handle variable-length trajectories, we adopt an exact-length bucketing strategy during training. Specifically, trajectories are grouped by sequence length, and each mini-batch contains only trajectories with the same length, which avoids padding tokens and positional masks in the primary training pipeline. During inference, the target length $L$ is sampled from the empirical length distribution $P(L)$ estimated from the training set, and the model generates a trajectory with exactly $L$ time steps. Accordingly, all loss terms are computed over the valid $L$ time steps within each batch.

For the road segment representation, we use pre-trained embeddings derived from the UMTG via Node2Vec. For reproducibility, we provide the Node2Vec configuration used to learn road-segment embeddings. Specifically, we perform biased random walks on the directed weighted UMTG, where edge weights correspond to transition counts between consecutive road segments. The walk length is set to 30, with 200 walks per node, and the context window size is set to 10. The return parameter and in-out parameter are set to $p = 1$ and $q = 2$, respectively, encouraging the walks to focus more on local neighborhood structures and thereby better capture local road-segment connectivity. The embedding dimensionality is selected from $16, 32, 64, 128$ and finally set to $D = 64$ according to the hyperparameter analysis. The embeddings are trained using the skip-gram objective with 5 negative samples for 10 epochs. The learned embedding matrix $E \in \mathbb{R}^{|R| \times D}$ is then fixed and used for RoadTraj vectorization and reconstruction.

To construct user-specific road preference conditions, we extract the top-$K = 100$ most frequently visited road segments from each user's historical trajectories and compute a weighted preference vector using softmax with temperature $\tau = 0.5$. We incorporate Classifier-Free Guidance (CFG) during training by randomly dropping the condition vector with probability $p_{\text{uncond}} = 0.1$, and apply CFG with guidance scale $\omega = 3$ during sampling. During evaluation, we generate the same number of synthetic trajectories as the real trajectories used for comparison. Each trajectory is independently sampled from Gaussian noise using stochastic DDPM sampling with CFG scale $\omega = 3$, and all metrics are computed over the complete generated dataset. All experiments are conducted on a single NVIDIA A100 GPU (40 GB).

### *5.5. Experimental Results*

We compare our method with baselines by generating equal-sized trajectory datasets. Table 2 reports the performance across JSD metrics and RSC. We define our key observations as follows:

First, rule-based methods (RWRN, Markov) ensure spatial validity with 100% RSC. However, they fail to capture real movement patterns. For instance, they yield high JSD-TD errors. This confirms that simple rules cannot model complex travel behaviors.

Second, the diffusion-based Diff-RNTraj significantly outperforms rule-based methods. It reduces JSD-TD by 84.35% on average compared to both RWRN and Markov. However, its spatial validity drops, with RSC scores of 76.82% and 92.45%. This suggests it struggles to balance mobility diversity and spatial constraint.

Finally, our method achieves the best JSD results and improves RSC over the diffusion-based baseline. Compared to the strongest baseline, Diff-RNTraj, it reduces the trajectory distribution divergence (JSD) by an average of 12.83% across both datasets. Notably, it also boosts RSC scores to 79.26% and 93.82%. These consistent gains demonstrate our model's effectiveness: the conditional encoding captures user-specific patterns without losing geographical coherence.

**Table 2.** Comparison of the proposed method with baseline models on two trajectory datasets. Bold values indicate the best performance.

| Methods | Dataset I: Beijing (Geolife-GPS) | | | | | Dataset II: Porto (Taxi) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | JSD-TD | JSD-SD | JSD-GPD | JSD-RS | RSC | JSD-TD | JSD-SD | JSD-GPD | JSD-RS | RSC |
| RWRN | 0.6854 | 0.2832 | 0.7742 | 0.9325 | 100.0% | 0.5846 | 0.0137 | 0.1656 | 0.8957 | 100.0% |
| Markov | 0.5147 | 0.1623 | 0.5597 | 0.7622 | 100.0% | 0.4821 | 0.0098 | 0.0953 | 0.5821 | 100.0% |
| Diff-RNTraj | 0.1656 | 0.1596 | 0.2178 | 0.1716 | 76.82% | 0.0195 | 0.0006 | 0.0246 | 0.0438 | 92.45% |
| w/o Top-K Selection | 0.1735 | 0.1552 | 0.2145 | 0.1798 | 78.15% | 0.0184 | 0.0005 | 0.0229 | 0.0415 | 93.13% |
| Uniform Weighting | 0.1642 | 0.1585 | 0.2259 | 0.1695 | 77.30% | 0.0188 | 0.0006 | 0.0235 | 0.0412 | 92.81% |
| Random Condition | 0.1968 | 0.1768 | 0.2487 | 0.1874 | 70.42% | 0.0455 | 0.0015 | 0.0384 | 0.0827 | 89.18% |
| Ours | **0.1453** | **0.1304** | **0.1855** | **0.1575** | **79.26%** | **0.0172** | **0.0005** | **0.0215** | **0.0403** | **93.82%** |

RWRN, Markov, and Diff-RNTraj were selected because they are directly comparable road-network-constrained baselines. All methods use the same map-matched road network, preprocessing pipeline, and evaluation metrics. VAE/GAN-based trajectory generators are important alternatives, but many operate on GPS coordinates or grids rather than road-segment-constrained RoadTraj sequences. Adapting them to this setting is left for future work.

### *5.6. Ablation Study*

To explore the effectiveness of different components in our proposed method, we conduct three ablation experiments on the Geolife and Porto datasets. These experiments involve: removing the Top-*K* selection mechanism (w/o Top-*K* Selection), replacing frequency-based weighting with uniform weighting (Uniform Weighting), and using random road segments instead of user-specific history (Random Condition). The experimental results are shown in Table 2.

We observe that removing the Top-*K* selection mechanism or using uniform weighting leads to performance degradation, with JSD-TD increases of up to 19.4% and 13.0%, respectively, on Geolife. This indicates that filtering occasional road segments and weighting frequently visited locations are both essential for capturing typical user behavior. The Random Condition experiment demonstrates the most significant impact, with JSD-TD increasing by 35.4% on Geolife and 164.5% on Porto, and RSC dropping from 79.26% to 70.42% and from 93.82% to 89.18% respectively. This dramatic degradation confirms that performance improvements arise from meaningful user preference information rather than simply providing additional conditioning signals. In summary, all three components contribute significantly to generating realistic and personalized trajectories.

The current ablation study focuses on the semantic preference encoding module, including top-K selection, frequency-based weighting, and user-specific conditioning. These variants show that the improvement is not caused merely by adding an arbitrary condition vector. We acknowledge that additional ablations, such as removing the spatial validity loss or replacing the frequency-based encoder with a sequence-aware user encoder, would further isolate the contribution of each component. We leave these variants for future work because they require retraining separate model configurations.

### *5.7. Hyperparameter Analysis*

We further explore the impact of various hyperparameters in our proposed model on both the GeoLife and Porto datasets. These hyperparameters include the road segment representation dimension $D$, the number of residual dilation convolution layers $L$, and the total diffusion steps $N$.

Pre-trained road segment representation dimension $D$. We vary $D$ within $\{16, 32, 64, 128\}$ to analyze its impact on trajectory generation. As illustrated in Figures 2a and 3a, we observe that as $D$ increases, the performance improves on both datasets. However, the sparse GeoLife dataset shows degradation at $D = 128$, suggesting that overly high-dimensional space introduces noise when road coverage is sparse. In contrast, the dense Porto dataset continues to improve at $D = 128$. Therefore, we choose $D = 64$ to ensure robustness across varying data densities.

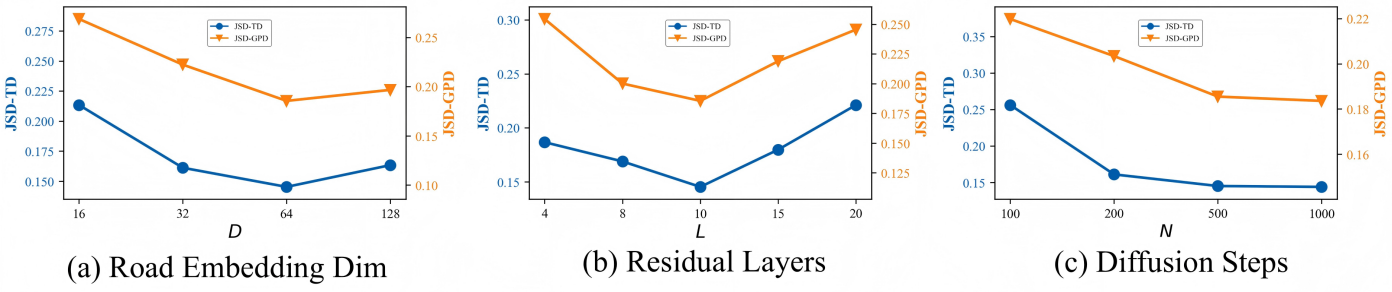



**Figure 2.** Hyperparameter sensitivity analysis on the GeoLife dataset.

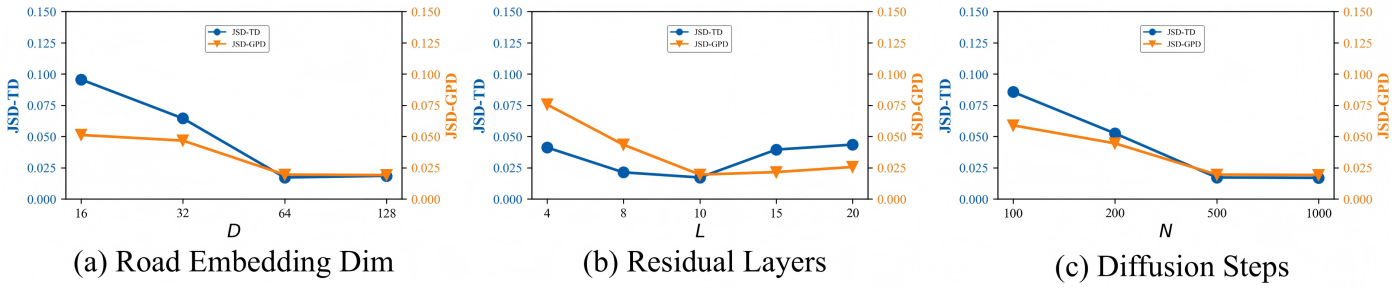



**Figure 3.** Hyperparameter sensitivity analysis on the Porto dataset.

The number of residual dilation convolution layers $L$ in the denoising network. We conduct experiments using various values of $L$, including 4, 8, 10, 15, and 20. As depicted in Figures 2b and 3b, an increase in $L$ enables trajectory points to better capture long-term temporal dependencies. However, when $L$ exceeds 10, GeoLife shows sharp performance degradation due to over-smoothing in sparse data, while Porto maintains robustness. We set $L = 10$ to balance receptive field size and training stability.

Diffusion steps $N$. As shown in Figures 2c and 3c, the generation quality improves as $N$ increases from 100 to 500. When $N$ exceeds 500, the performance gains become marginal while inference time doubles. Therefore, we set $N = 500$ to balance effectiveness and efficiency.

### *5.8. Visualization Analysis*

To intuitively assess the generation quality, we conduct two distinct visualization experiments involving trajectory paths and spatial heatmaps. We first focus on trajectory path visualization to inspect spatial validity by overlaying the generated path geometries onto the road network. As illustrated in Figures 4 and 5, we observe distinct behaviors across the two datasets. On the data-scarce GeoLife dataset, the baseline Diff-RNTraj tends

to produce fragmented paths with visible disconnects. In contrast, our model generates significantly more coherent trajectories that effectively bridge topological gaps. Conversely, on the dense Porto dataset, both our model and the baseline successfully reconstruct the road network structure, exhibiting high consistency with the ground truth. This suggests that given sufficient data, diffusion-based models can capture urban topology well. Therefore, in this high-density scenario, the superiority of our method is highlighted more by the quantitative metrics than by visual inspection.

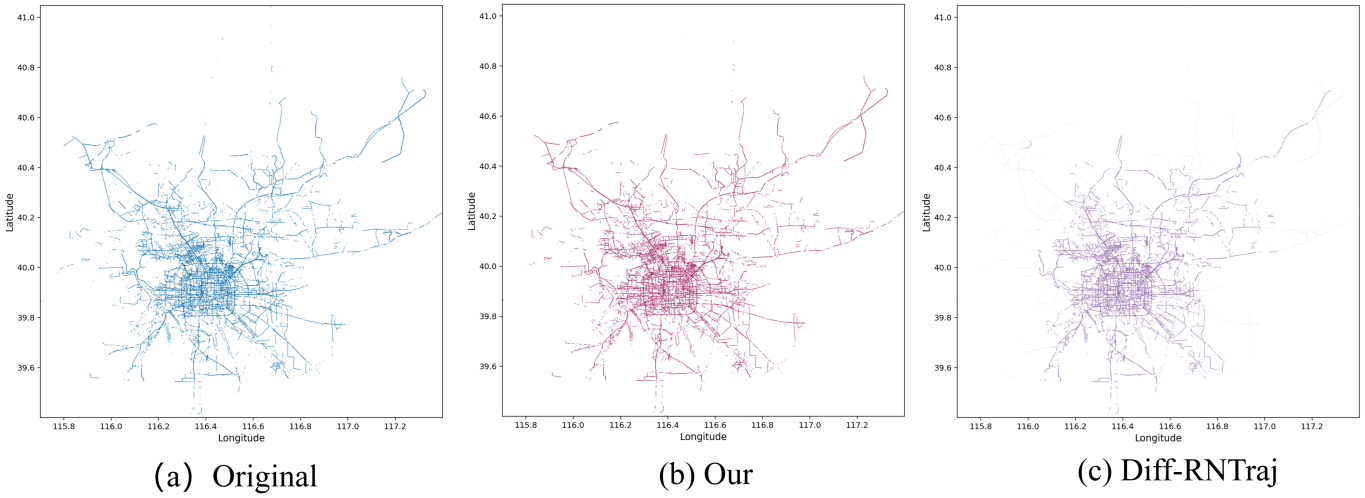



**Figure 4.** Trajectory path visualization on GeoLife.

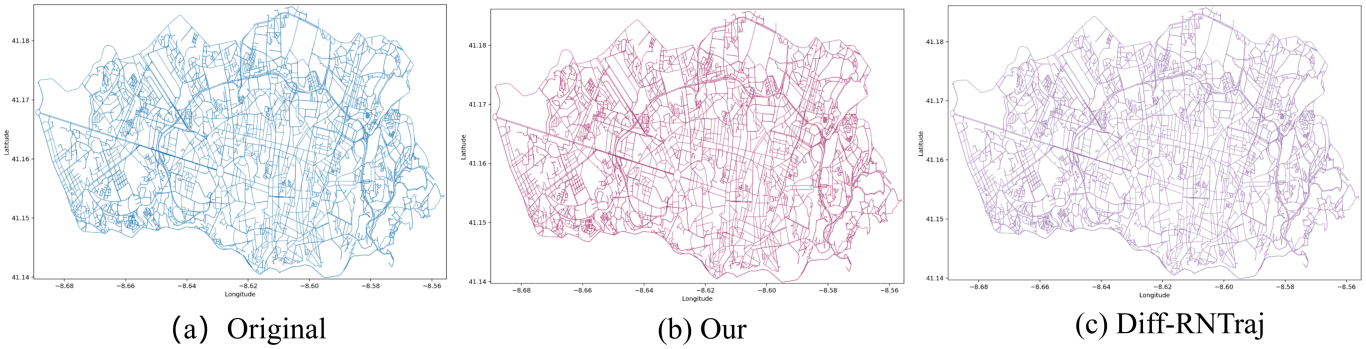



**Figure 5.** Trajectory path visualization on Porto.

Subsequently, we analyze the spatial distribution heatmaps to compare global patterns, as shown in Figures 6 and 7. We generate an equal volume of trajectory points as the original dataset and visualize their density using the Probability Density Function (PDF). By comparing the heatmaps from our model, Diff-RNTraj, and the real data, we observe a clear distinction on GeoLife: the baseline tends to yield over-smoothed distributions, blurring fine-grained details. Our model, however, preserves the sharper boundaries of hotspots, closely matching reality. On Porto, both models mirror the ground truth distribution, demonstrating their effectiveness in capturing traffic intensity. Collectively, these visualization results, combined with the quantitative metrics in Table 2, indicate that Sem-RoadDiff generates trajectories with superior distributional realism and improved road-network consistency.

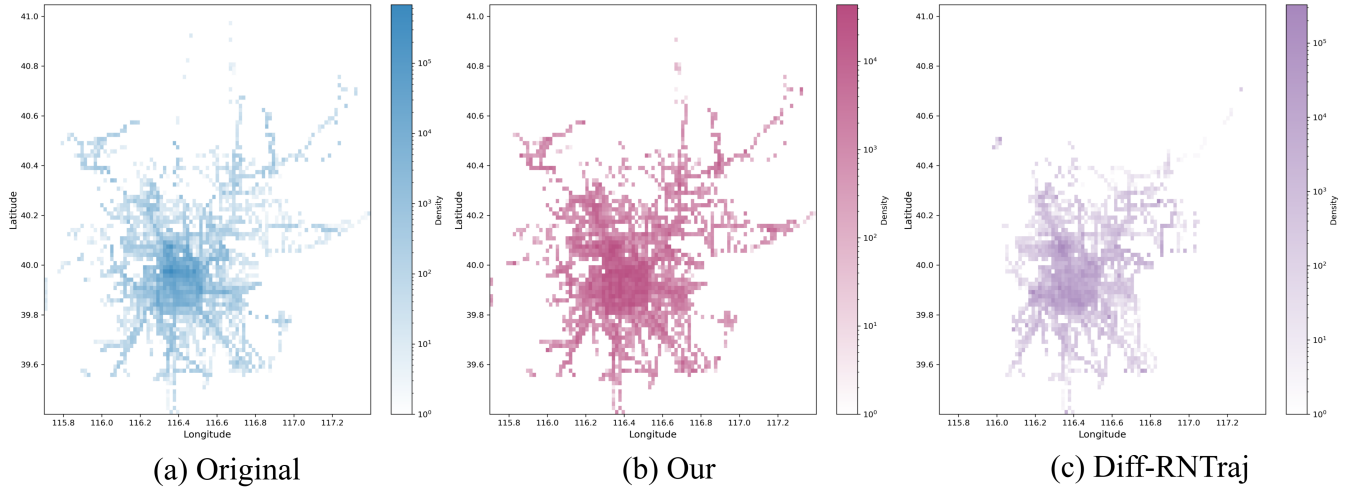



**Figure 6.** Spatial distribution heatmaps on GeoLife.

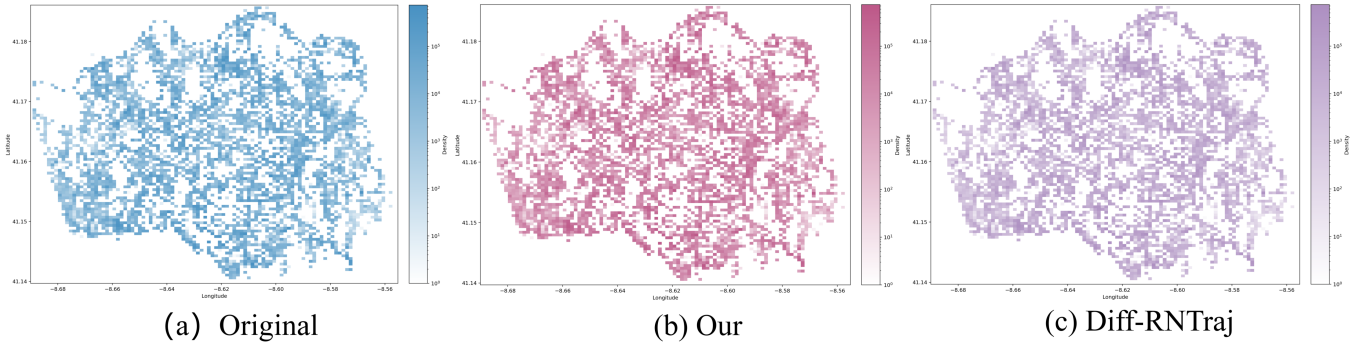



**Figure 7.** Spatial distribution heatmaps on Porto.

## 6. Conclusions

In this paper, we addressed trajectory generation from a perspective that considers both distributional symmetry and asymmetric road-network transitions, focusing on preserving the consistency between real and generated trajectory distributions and the direction-dependent mobility structure.

To tackle this, we proposed Sem-RoadDiff, a dual-guided diffusion model for personalized and road network-constrained trajectory generation. Specifically, our model incorporates two key components. First, on the semantic side, we designed a semantic preference guidance mechanism that encodes user history into a preference-weighted embedding. This enables the model to capture user-level mobility patterns without directly relying on raw trip-level records. Second, on the constraint side, we introduced road-aware constraints to facilitate topologically valid transitions. By employing a spatial validity loss derived from the User Mobility Transition Graph, coupled with a post-generation alignment strategy, our model effectively penalizes invalid road transitions and improves overall spatial validity. Extensive experiments on the GeoLife and Porto datasets demonstrated the effectiveness of our approach, achieving an average improvement of 12.8% across JSD metrics and enhancing road segment connectivity by an absolute average of 1.91 percentage points. These results indicate that explicitly considering both distributional symmetry and asymmetric road-network transitions provides an effective perspective for generating realistic and spatially valid trajectories.

Although real-world trajectories are inherently spatiotemporal, the current version of Sem-RoadDiff primarily targets spatial generation under road-network constraints. The model generates road-segment sequences and segment-level moving ratios, but does not explicitly model timestamps, travel-time intervals, speed, or acceleration. The proposed method should therefore not be interpreted as a complete spatiotemporal mobility simulator; incorporating these temporal dynamics into the diffusion process remains a primary direction for extension.

Furthermore, although the spatial validity loss improves road-segment connectivity, residual invalid transitions may still occur. This limitation arises because the generated continuous embeddings are decoded into discrete road segments via nearest-neighbor matching. Small embedding deviations, sparse transition regions, map-matching noise, or limited transition coverage may map a point to a nearby but topologically disconnected segment. Addressing this through constrained decoding, shortest-path repair, or graph-based projection poses an open challenge for subsequent iterations.

Finally, the reported JSD-based metrics and RSC should be interpreted as aggregate measures of distributional fidelity and road-segment connectivity, rather than as complete validations of individual-level semantic correctness or trip intent. Because the evaluation is conducted on GeoLife and Porto with a selected baseline set, it does not establish universal superiority across all cities, data densities, or mobility scenarios. Additionally, deterministic metric values are reported under the adopted experimental protocol without claiming statistical significance from multi-seed testing. Although synthetic trajectory generation is partly motivated by privacy concerns, Sem-RoadDiff does not yet provide a formal differential privacy guarantee. Future work will systematically address these contextual factors, privacy-preserving mechanisms, and multi-seed statistical evaluations to build upon the current framework.

**Author Contributions:** Conceptualization, J.C. and Y.Z.; methodology, Y.Z.; software, Y.Z.; validation, Y.Z. and J.C.; formal analysis, Y.Z.; investigation, Y.Z.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, J.C., J.Z. and X.S.; visualization, Y.Z.; supervision, J.C., J.Z. and X.S.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key R&D Program of China (Grant No. 2024YFB4303400).

**Data Availability Statement:** The Geolife trajectory dataset is available at https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/ (accessed on 1 December 2025). The Porto taxi dataset can be accessed via Kaggle at https://www.kaggle.com/competitions/pkdd-15-taxi-trip-time-prediction-ii/data (accessed on 1 December 2025). Road network data are obtained from OpenStreetMap (https://www.openstreetmap.org/) (accessed on 1 December 2025).

**Acknowledgments:** The authors would like to thank the anonymous reviewers for their constructive comments.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CFG | Classifier-Free Guidance |
| DDPM | Denoising Diffusion Probabilistic Models |
| GAN | Generative Adversarial Network |
| GPS | Global Positioning System |

| | |
|---|---|
| HMM | Hidden Markov Model |
| JSD | Jensen-Shannon Divergence |
| JSD-TD | JSD Total Distance |
| JSD-SD | JSD Segment Distance |
| JSD-GPD | JSD Grid Point Density |
| JSD-RS | JSD Road Segment |
| MLP | Multi-Layer Perceptron |
| RSC | Road Segment Connectivity |
| RWRN | Random Walk on the Road Network |
| UMTG | User Mobility Transition Graph |
| VAE | Variational Autoencoder |

## References


1. Ruan, S.; Long, C.; Bao, J.; Li, C.; Yu, Z.; Li, R.; Liang, Y.; He, T.; Zheng, Y. Learning to Generate Maps from Trajectories. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 890–897.
2. Ji, J.; Wang, J.; Wu, J.; Han, B.; Zhang, J.; Zheng, Y. Precision CityShield Against Hazardous Chemicals Threats via Location Mining and Self-Supervised Learning. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 3072–3080.
3. Yuan, Y.; Ding, J.; Wang, H.; Jin, D.; Li, Y. Activity Trajectory Generation via Modeling Spatiotemporal Dynamics. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 4752–4762.
4. Dai, J.; Yang, B.; Guo, C.; Ding, Z. Personalized Route Recommendation Using Big Trajectory Data. In *2015 IEEE 31st International Conference on Data Engineering*; IEEE: Piscataway, NJ, USA, 2015; pp. 543–554.
5. Lim, K.H.; Chan, J.; Karunasekera, S.; Leckie, C. Tour Recommendation and Trip Planning Using Location-Based Social Media: A Survey. *Knowl. Inf. Syst.* **2019**, *60*, 1247–1275.
6. Li, K.; Rao, X.; Pang, X.; Chen, L.; Fan, S. Route Search and Planning: A Survey. *Big Data Res.* **2021**, *26*, 100246. [CrossRef]
7. Han, X.; Zhao, X.; Zhang, L.; Wang, W. Mitigating Action Hysteresis in Traffic Signal Control with Traffic Predictive Reinforcement Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 673–684.
8. Zhu, Y.; Liu, Y.; James, J.Q.; Yuan, X. Semi-Supervised Federated Learning for Travel Mode Identification from GPS Trajectories. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 2380–2391. [CrossRef]
9. Zhang, J.; Huang, Q.; Huang, Y.; Ding, Q.; Tsai, P.W. DP-TrajGAN: A privacy-aware trajectory generation model with differential privacy. *Future Gener. Comput. Syst.* **2023**, *142*, 25–40. [CrossRef]
10. Zheng, Y. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* **2015**, *6*, 1–14.
11. Luca, M.; Barlacchi, G.; Lepri, B.; Pappalardo, L. A Survey on Deep Learning for Human Mobility. *ACM Comput. Surv. (CSUR)* **2021**, *55*, 1–44. [CrossRef]
12. Jiang, S.; Yang, Y.; Gupta, S.; Veneziano, D.; Athavale, S.; González, M.C. The TimeGeo modeling framework for urban mobility without travel surveys. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, E5370–E5378. [CrossRef] [PubMed]
13. Wang, J.; Dong, L.; Cheng, X.; Yang, W.; Liu, Y. An extended exploration and preferential return model for human mobility simulation at individual and collective levels. *Phys. Stat. Mech. Its Appl.* **2019**, *534*, 121921. [CrossRef]
14. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent Development and Applications of SUMO-Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.
15. Ding, W.; Wang, W.; Zhao, D. A Multi-Vehicle Trajectories Generator to Simulate Vehicle-to-Vehicle Encountering Scenarios. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2019; pp. 4255–4261. [CrossRef]
16. Huang, D.; Song, X.; Fan, Z.; Jiang, R.; Shibasaki, R.; Zhang, Y.; Wang, H.; Kato, Y. A Variational Autoencoder Based Generative Model of Urban Human Mobility. In *Proceedings of the 2019 2nd IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 425–430. [CrossRef]
17. Long, Q.; Wang, H.; Li, T.; Huang, L.; Wang, K.; Wu, Q.; Li, G.; Liang, Y.; Yu, L.; Li, Y. Practical Synthetic Human Trajectories Generation Based on Variational Point Processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 4561–4571. [CrossRef]
18. Cao, C.; Li, M. Generating Mobility Trajectories with Retained Data Utility. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 2610–2620. [CrossRef]

19. Feng, J.; Yang, Z.; Xu, F.; Yu, H.; Wang, M.; Li, Y. Learning to Simulate Human Mobility. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 3426–3433. [CrossRef]
20. Ouyang, K.; Shokri, R.; Rosenblum, D.S.; Yang, W. A Non-Parametric Generative Model for Human Trajectories. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; Volume 18, pp. 3812–3817.
21. Rao, J.; Gao, S.; Kang, Y.; Huang, Q. LSTM-TrajGAN: A Deep Learning Approach to Trajectory Privacy Protection. *arXiv* **2020**, arXiv:2006.10521.
22. Wang, X.; Liu, X.; Lu, Z.; Yang, H. Large Scale GPS Trajectory Generation Using Map Based on Two Stage GAN. *J. Data Sci.* **2021**, *19*, 126–141. [CrossRef]
23. Zhu, Y.; Ye, Y.; Zhang, S.; Zhao, X.; Yu, J. DiffTraj: Generating GPS Trajectory with Diffusion Probabilistic Model. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 65168–65188.
24. Chu, C.; Zhang, H.; Wang, P.; Lu, F. Simulating human mobility with a trajectory generation framework based on diffusion model. *Int. J. Geogr. Inf. Sci.* **2024**, *38*, 847–878. [CrossRef]
25. Wei, T.; Lin, Y.; Guo, S.; Lin, Y.; Huang, Y.; Xiang, C.; Bai, Y.; Wan, H. Diff-RNTraj: A Structure-Aware Diffusion Model for Road Network-Constrained Trajectory Generation. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 7940–7953. [CrossRef]
26. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33, pp. 6840–6851.
27. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 3634–3640. [CrossRef] [PubMed]
28. Jiang, X. Designing a Time-Driven Simulation Framework for Large-Scale Traffic Networks. In *Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 79–80. [CrossRef]
29. Wu, Y.; Zhang, X.; Wang, Y. Adaptive Graph Neural Diffusion for Traffic Demand Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 4350–4354. [CrossRef]
30. Li, W.; Yang, X.; Tang, X.; Xia, S. SDCN: Sparsity and Diversity Driven Correlation Networks for Traffic Demand Forecasting. In *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8. [CrossRef]
31. Chen, X.; Huang, C.; Wang, C.; Chen, L. Trajectory generation: A survey on methods and techniques. *GeoInformatica* **2025**, *29*, 351–376. [CrossRef]
32. Sohl-Dickstein, J.; Weiss, E.A.; Maheswaranathan, N.; Ganguli, S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 7–9 July 2015; pp. 2256–2265.
33. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.L.; Ghasemipour, K.; Lopes, R.G.; Ayan, B.K.; Salimans, T.; et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 36479–36494. [CrossRef]
34. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. *arXiv* **2021**, arXiv:2010.02502.
35. Nichol, A.Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; Mcgrew, B.; Sutskever, I.; Chen, M. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 16784–16804.
36. Dhariwal, P.; Nichol, A. Diffusion Models Beat GANs on Image Synthesis. In Proceedings of the 35th International Conference on Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 8832–8848.
37. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
38. Voleti, V.S.; Jolicoeur-Martineau, A.; Pal, C.J. MCVD-Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation. In Proceedings of the Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 23371–23385.
39. Li, X.L.; Thickstun, J.; Gulrajani, I.; Liang, P.; Hashimoto, T. Diffusion-LM Improves Controllable Text Generation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4328–4343. [CrossRef]
40. Liu, J.; Li, C.; Ren, Y.; Chen, F.; Zhao, Z. DiffSinger: Singing Voice Synthesis via Shallow Diffusion Mechanism. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 11020–11028.

41. Wen, H.; Lin, Y.; Xia, Y.; Wan, H.; Wen, Q.; Zimmermann, R.; Liang, Y. DiffSTG: Probabilistic Spatio-Temporal Graph Forecasting with Denoising Diffusion Models. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1–12.
42. Chen, H.; Ding, J.; Li, Y.; Wang, Y.; Zhang, X.P. Social Physics Informed Diffusion Model for Crowd Simulation. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 474–482. [CrossRef]
43. Yuan, Y.; Ding, J.; Shao, C.; Jin, D.; Li, Y. Spatio-temporal Diffusion Point Processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 3173–3184.
44. Newson, P.; Krumm, J. Hidden Markov Map Matching through Noise and Sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*; Association for Computing Machinery: New York, NY, USA, 2009; pp. 336–343.
45. Gambs, S.; Killijian, M.O.; del Prado Cortez, M.N.n. Next Place Prediction Using Mobility Markov Chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*; Association for Computing Machinery: New York, NY, USA, 2012; pp. 1–6. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.