

DIFFPATH: GENERATING ROAD NETWORK BASED PATH WITH LATENT DIFFUSION MODEL

Anonymous authors

Paper under double-blind review

ABSTRACT

With the increasing use of GPS technology, path has become essential for applications such as navigation, urban planning, and traffic optimization. However, obtaining real-world path presents challenges due to privacy concerns and the difficulty of collecting large datasets. Existing methods, including count-based and deep learning approaches, struggle with two main challenges: handling complex distributions of road segments and ensuring global coherence in generated paths. To address these, we introduce DiffPath, a path generation model based on Latent Diffusion Models (LDMs). By embedding path into a continuous latent space and leveraging a transformer architecture, DiffPath captures both local transitions and global dependencies, ensuring the generation of realistic paths. Experimental results demonstrate that our model outperforms existing approaches in generating paths that adhere to real-world road network structures while maintaining privacy.

1 INTRODUCTION

With the widespread adoption of GPS technology and mobile devices, path has become essential for optimizing navigation systems (Thomason et al., 2020), supporting smart city planning (Lin et al., 2020), and route planning (Bibri, 2021). However, the collection and utilization of path raise significant privacy concerns, as such data often contains sensitive information regarding individuals’ movements (Lu et al., 2019; Monreale et al., 2023; Zhu et al., 2024a). Additionally, large-scale collection of this data faces challenges due to regulatory constraints, such as the General Data Protection Regulation (GDPR) in Europe.

Recent studies have highlighted synthetic data generation as a promising alternative that ensures privacy protection while preserving data utility (Long et al., 2023; Zhu et al., 2024a;b). Despite the potential of synthetic path to augment datasets and substitute real data in privacy-sensitive applications, generating synthetic data presents two significant challenges: 1) **capturing complex path distributions**, 2) **ensuring global coherence in generated paths**.

Path generation in urban road networks is particularly challenging due to the inherent inhomogeneity of the limited real-world dataset, which exhibits a clear long-tail distribution of road segment traversals. Figure 1 illustrates the statistics of road segment traversal in real world dataset (details provided in the 5.1.1), highlighting that some road segments, located in urban centers or transportation hubs, are frequently used and well-documented. Yet many road segments, particularly in more remote areas, are rarely traversed and thus suffer from limited data availability (Zhao et al., 2022; Zhu et al., 2022). This imbalance prevents the existing models from learning a comprehensive understanding of the complete road network in a city.

Existing path generation methods can be categorized into count-based methods and deep learning methods. The **count-based** methods generate paths by analyzing the frequency of transitions within historical path (Baratchi et al., 2014; Sutton et al., 1999). Due to the unevenness of limited real datasets, many road segments lack adequate historical observations, leading the model to make inaccurate or zero probability estimates and thus ignore these road segments during path generation. On the other hand, **deep learning methods** utilize neural networks to capture complex patterns in historical paths (Yin et al., 2017; Yu et al., 2017; Wang et al., 2022; Shi et al., 2024). Since

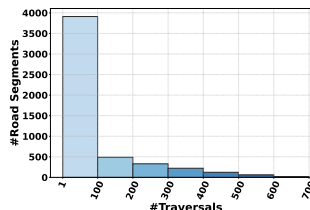


Figure 1: Statistics of Road Segment Traversals.

high-frequency paths appear more frequently in the dataset, the model can more easily fit to these common paths. However, low-frequency paths appear less in the data, and the model lacks sufficient training signals on them. As a result, the model struggles to generate paths that include these rare segments.

Another significant challenge in path generation for urban road networks is that the generated paths conform to the constraints of the road network, but are not realistic enough because they do not conform to most situations in reality. While ensuring connectivity between adjacent road segments is essential, models must also consider the long-range dependencies of non-adjacent road segments throughout the entire path to generate realistic paths (Yang et al., 2021). Ignoring the influence of non-adjacent road segments can lead to paths that are locally coherent but globally suboptimal or unrealistic. As illustrated in Figure 2, consider two paths from v_1 to v_7 : $P_1 = [v_1, v_2, v_7]$ (depicted in blue) and $P_2 = [v_1, v_2, v_4, v_5, v_6, v_7]$ (depicted in red). Path P_2 traverses multiple intermediate nodes, resulting in a longer and unrealistic path. Although it is in line with the constraints of the road network, P_2 does not consider that selecting v_4 will result in a longer path to reach v_7 , making the entire path unrealistic. Path P_1 represents the realistic path typically taken in most real-world scenarios. This example demonstrates how models focusing solely on adjacent road segments might generate unrealistic paths by failing to consider non-adjacent connections that lead to more optimal routes. Unfortunately, existing methods (Wang et al., 2022; Shi et al., 2024) neglect this critical issue.

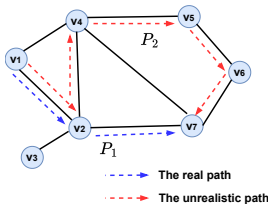


Figure 2: An example of real and unrealistic path.

the diffusion model. Instead of predicting the added noise at each time step, the model directly predicts the initial latent representation.

Additionally, transformers are capable of modeling long-range dependencies through a self-attention mechanism (Peebles & Xie, 2023). To capture the correlations of both adjacent and non-adjacent road segments, we integrate a transformer-based architecture into the diffusion model. The self-attention mechanism enables the model to consider the long-range dependencies, ensuring global path consistency and realistic path generation. Moreover, we introduce a new similarity score specifically designed to rigorously evaluate path realism by considering both local transitions and global coherence. In summary, we make the following main contributions:

- We present DiffPath, the first latent diffusion-based approach for path generation, effectively modeling complex data distributions and capturing long-range dependencies between road segments.
- We introduce a custom loss function to address the long-tail distribution of road segments, ensuring diverse path generation. Additionally, we enhance diffusion with positional embeddings and a clamping mechanism for topological validity and contextual coherence.
- We validate DiffPath on two real-world datasets, showing it outperforms state-of-the-art models in generating high-fidelity, realistic paths while preserving data privacy.

2 RELATED WORK

2.1 PATH GENERATION FOR PATTERN MINING

Many methods have been developed to apply for path pattern mining. These approaches can broadly be categorized into two main categories: **count-based** and **deep learning-based models**.

108 Early models primarily relied on count-based methods, such as Markov chain models, which gen-
109 erate paths by capturing local state dependencies (Sutton et al., 1999). These models assume that
110 the probability of transitioning to the next road segment depends solely on the current road segment,
111 effectively modeling short-term dependencies. However, in the large state spaces typical of urban
112 road networks, many state transitions are rare or absent from the training data, leading to sparsity
113 issues. To mitigate this, hierarchical hidden states were introduced to reduce state space complexity
114 and alleviate sparsity (Baratchi et al., 2014). Despite these improvements, count-based models still
115 predominantly rely on information from the current and neighboring states, making it difficult to
116 capture global dependencies and ensure long-range path consistency.

117 With advances in machine learning and reinforcement learning, deep learning-based models have
118 become increasingly prominent in path generation. Generative Adversarial Networks (GANs) have
119 been used to enhance path generation by integrating complex reward functions, improving both the
120 diversity and quality of generated paths (Choi et al., 2021; Yu et al., 2017). However, GAN-based
121 models often require large amounts of data and can struggle with the stability of training. Re-
122 searchers have also increasingly turned to sequence-to-sequence (seq2seq) models for path genera-
123 tion (Wu et al., 2017). These models show promise for modeling long paths by encoding sequences
124 of road segments and iteratively predicting subsequent road segments. However, adapting seq2seq
125 models to respect the topological constraints of road networks presents a significant challenge. Ar-
126 chitectures based on recurrent neural networks (RNNs) (Rao et al., 2020) have been explored to
127 capture sequential dependencies, but they often suffer from the vanishing gradient problem when
128 handling long sequences, limiting their ability to capture global path information.

129 To address some of these limitations, discrete diffusion models have been introduced (Shi et al.,
130 2024). These models represent road network topology using adjacency matrices during the diffusion
131 process, which is particularly effective for conditional task generation. However, in unconditional
132 generation tasks, maintaining symmetry in the adjacency matrix can inaccurately represent one-way
133 streets as bidirectional, introducing bias and inaccuracies in the generated paths.

134 135 2.2 DIFFUSION MODELS

136
137 The diffusion model, initially introduced as a probabilistic generation framework (Sohl-Dickstein
138 et al., 2015), operates through two continuous processes: a forward process that gradually perturbs
139 the data distribution by adding multiscale noise and a reverse process that reconstructs the data
140 by learning its underlying distribution (Ho et al., 2020). Subsequent advances have significantly
141 improved the quality of the generated samples and accelerated the sampling process. Further in-
142 novations include the introduction of a non-Markovian diffusion process (Song et al., 2021; Nichol
143 et al., 2022), which reduces the number of sampling steps, and the approach of learning the vari-
144 ance in the reverse process to further streamline sampling. In addition, structural optimizations have
145 been performed in reverse denoising neural networks to enhance the quality of the generated sam-
146 ples (Nichol & Dhariwal, 2021). As a cutting-edge generative model, the diffusion model has out-
147 performed other models in various tasks, including computer vision (Rombach et al., 2022), natural
148 language processing (Li et al., 2022), multimodal learning (Liu et al., 2023; Avrahami et al., 2022),
149 and traffic prediction (Wen et al., 2024). Recent research has used diffusion models to generate
150 synthetic data to improve datasets in specific domains (Zhu et al., 2024a;b), such as the generation
151 of spatio-temporal trajectory data. Despite their widespread application, diffusion models have not
152 been extensively explored for path generation. This is primarily due to the distinct nature of the
153 task: path generation requires adherence to road network topology and sequential constraints (Wang
154 et al., 2022; Shi et al., 2024), which are fundamentally different from the data structures handled in
155 typical computer vision or natural language tasks.

156 157 3 PRELIMINARIES

158 159 3.1 PROBLEM DEFINITION

160 In this section, we introduce the definitions and notation that we use in this paper.
161

Definition 1 (Road network) A road network is represented as a graph $G = (\mathbb{V}, \mathbb{E})$, where \mathbb{V} is the set of vertices v_i , each representing a road intersection, and $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is the set of edges $e_i = (v_j, v_k)$, representing road segments from v_j to v_k .

Definition 2 (Path) A path $p = \langle v_1, v_2, v_3, \dots, v_L \rangle$ is a sequence of connected vertices, where each $v_i \in \mathbb{V}$ represents a vertex in the path, and two consecutive vertices are connected by an edge in \mathbb{E} .

Definition 3 (Path Generation) The objective is to develop a generative model capable of producing synthetic paths $\mathbb{N} = \{\hat{p}_1, \hat{p}_2, \dots\}$, where the generated paths \hat{p}_i preserve the structural and spatial characteristics of real-world paths.

Table 1: Notations

Notation	Description
$G = (\mathbb{V}, \mathbb{E})$	A graph representing the road network with vertices \mathbb{V} and edges \mathbb{E} .
(v_0, v_1, \dots, v_n)	The sequence of vertices in path p .
e_i	The i -th road segment in path p .
v_i	The i -th vertex in path p .
p	A real-world path.
\hat{p}	A generated synthetic path.
z_t	The embedded path at time step t .
\mathbb{R}	The real-world dataset.
\mathbb{N}	The synthetic dataset.
α_t, β_t	Hyperparameters controlling the diffusion scale.

4 METHODOLOGY

4.1 OVERVIEW

The model framework is illustrated in Figure 3. **DiffPath** begins by embedding the discrete path of length l into a continuous latent space $\mathbb{R}^{l \times d}$ (referred to as **Encoding**). The embedded data then undergoes a forward and reverse diffusion process, allowing the model to effectively learn and represent complex patterns and dependencies within path in the latent space. The attention mechanism of the transformer enhances the model’s ability to process sequential data and capture long-range patterns, enabling it to learn complex data distributions and generate new paths (referred to as the **Generation process**). Finally, the sampled results are mapped from the latent space back to discrete path (referred to as **Decoding**).

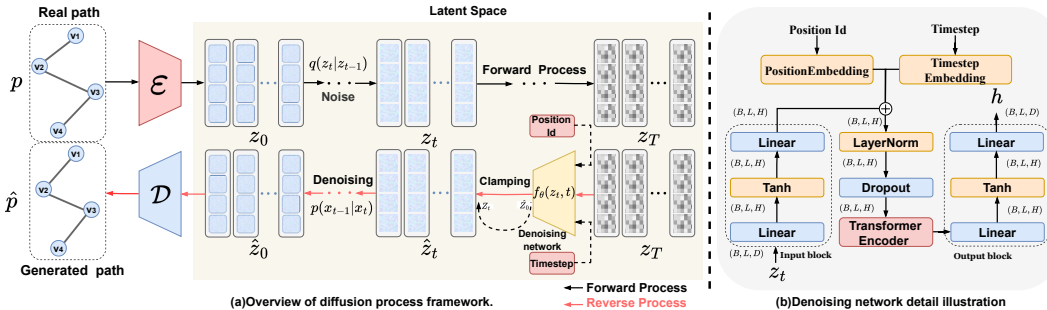


Figure 3: Overview of diffusion process framework. Discrete path is embedded into a continuous latent space and processed through a diffusion model to generate synthetic path.

4.2 ENCODING AND DECODING

To apply a latent diffusion model to the discrete path generation task, we first define an embedding function $E_{MB}(v_i)$ that maps each discrete path node v_i (e.g., intersections or road segments) into a

vector in \mathbb{R}^d . For a path \mathbf{p} of length l , the embedding of the path is represented as:

$$E_{MB}(\mathbf{p}) = [E_{MB}(v_1), E_{MB}(v_2), \dots, E_{MB}(v_l)] \in \mathbb{R}^{l \times d}. \quad (1)$$

In order to operate within the latent space, we leverage a Markov transition in the forward process from the discrete path \mathbf{p} to its latent continuous representation \mathbf{z}_0 . This transition is parameterized as:

$$E_\phi(\mathbf{z}_0|\mathbf{p}) = \mathcal{N}(E_{MB}(\mathbf{p}), \sigma_0 I), \quad (2)$$

where $E_{MB}(\mathbf{p})$ is the embedded representation of the path, and $\sigma_0 I$ represents the covariance matrix of the Gaussian noise. By introducing noise into the latent space, the model captures both the underlying structure of the path and the inherent uncertainty in the transitions. Unlike pre-trained or fixed embeddings, we jointly learn both the path embeddings and the parameters of the diffusion model within the training objective. This end-to-end training enables the model to learn optimal latent representations suited to the path generation task.

During the reverse process, the continuous latent representation is gradually denoised and mapped back to the discrete path domain. This reverse diffusion process restores the path by iteratively refining the latent representation, using learned parameters to approximate the true distribution of the path. We select the most probable road segment at each position based on the current denoised representation \mathbf{z}_0 . Formally, the mapping is defined as:

$$D_\theta(\mathbf{p}|\mathbf{z}_0) = \prod_{i=1}^l g_\theta(v_i|z_i), \quad (3)$$

where $g_\theta(v_i|z_i)$ is modeled by a softmax distribution over all possible road segments. The softmax distribution ensures that the probabilities of each possible road segment are normalized, and the road segment with the highest probability is selected. Rather than a single deterministic step, the reverse process involves a series of stochastic updates that gradually recover the discrete path from the noisy latent representation.

4.3 TRANSFORMER-BASED DIFFUSION PROCESS

Forward and Reverse Process The forward process begins with the initial latent representation \mathbf{z}_0 , progressively adding Gaussian noise over a series of steps, generating increasingly noisy states $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T$. Each transition is defined by a Gaussian distribution:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1 - \beta_t} \mathbf{z}_{t-1}, \beta_t I), \quad (4)$$

where β_t represents the noise variance at time step t . By the final step T , the latent variable \mathbf{z}_T becomes a near-Gaussian distribution.

The reverse process aims to reconstruct the original latent state \mathbf{z}_0 from the noisy state \mathbf{z}_T , using a learned denoising model. The reverse transitions follow the Gaussian distribution:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, t), \sigma_\theta(\mathbf{z}_t, t)^2 I), \quad (5)$$

where $\mu_\theta(\mathbf{z}_t, t)$ and $\sigma_\theta(\mathbf{z}_t, t)^2$ are the predicted mean and variance, respectively, at each time step t , both of which are learned parameters.

Training Objective Traditional diffusion models add noise at each step and learn to denoise the data in the reverse process to reconstruct the original data. This approach is effective for continuous data, where noise can be gradually removed due to the smooth nature of the data. This is typically done by minimizing a loss function designed to predict the added noise:

$$\min_{\theta} \mathbb{E}_{\mathbf{z}_t, \epsilon} [\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2], \quad (6)$$

where ϵ represents the noise added to the original data \mathbf{z}_0 , and $\epsilon_\theta(\mathbf{z}_t, t)$ is the model's predicted noise. \mathbf{z}_t represents the noisy data at time step t , and $\sqrt{\bar{\alpha}_t}, \sqrt{1 - \bar{\alpha}_t}$ are weighting coefficients of the diffusion process.

In path generation tasks, the data consists of discrete nodes (e.g., intersections or road segments) (Yang et al., 2023). Adding noise to such discrete data at each step can cause significant

270 deviations from the original structure, particularly for low-frequency road segments. Unlike contin-
 271 uous data, discrete data lacks a range of possible values, making noise introduction more likely to
 272 result in irreversible changes. Moreover, since the diffusion process involves iterative noise addition
 273 and removal, errors in predicting certain road segments, especially low-frequency ones, can accu-
 274 mulate over time. This accumulation can lead to the model ignoring or misrepresenting these road
 275 segments during path generation.

276 To overcome these limitations, we build on insights from previous work (Chen et al., 2023; Li et al.,
 277 2022) by directly predicting the latent representation \mathbf{z}_0 for the entire path. Instead of progressively
 278 predicting noise at each step, the model learns the full path representation in one pass. This allows
 279 the model to capture sparse or uncommon road segments by focusing on the overall path structure,
 280 enhancing accuracy in path generation:

$$281 \quad L_{train}(\mathbf{p}) = \sum_{t=1}^T \mathbb{E}_{q(\mathbf{z}_t|\mathbf{z}_0)} \|f_{\theta}(\mathbf{z}_t, t) - \mathbf{z}_0\|^2, \quad (7)$$

284 where $f_{\theta}(\mathbf{z}_t, t)$ is the model’s prediction of \mathbf{z}_0 at time step t . By directly learning to predict \mathbf{z}_0 . The
 285 predicted network structure is shown in the figure 3(b). We leverage transformer with its powerful
 286 self-focused mechanism to capture remote dependencies and sequential patterns in path. By combin-
 287 ing time step information and location information, the model can well take into account local node
 288 transformation and global path dependence. To further enhance the model’s performance during the
 289 reverse process, we apply the *clamping trick*, a method introduced in previous research (Li et al.,
 290 2022). In this method, the predicted latent state $f_{\theta}(\mathbf{z}_t, t)$ is clamped to the nearest valid road seg-
 291 ment embedding, ensuring that each step in the reverse process adheres closely to the true discrete
 292 structure of the data. The clamping is performed as follows:

$$293 \quad \mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_t} \cdot \text{Clamp}(f_{\theta}(\mathbf{z}_t, t)) + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad (8)$$

294 where $\epsilon \sim \mathcal{N}(0, I)$, and the clamping operation ensures that the predicted vector aligns with a valid
 295 road segment in the embedding space. This approach reduces errors during the decoding process,
 296 improving the accuracy of the generated paths. The overall training objective combines the losses
 297 from the embedding process, diffusion process, and decoding process. This ensures that the model
 298 captures the full complexity of path while maintaining both local and global structure. The final loss
 299 is expressed as:

$$300 \quad L_{train}(P) = \mathbb{E}_{q_{\phi}(\mathbf{z}_0|X)} \left[\sum_{t=1}^T \|f_{\theta}(\mathbf{z}_t, t) - \mathbf{z}_0\|^2 + \log E_{\phi}(\mathbf{z}_0|\mathbf{p}) - \log D_{\theta}(\mathbf{p}|\mathbf{z}_0) \right]. \quad (9)$$

303 The training process algorithm is as follows:

305 **Algorithm 1** Training Process of DiffPath

- 306 1: **Input:** Real path \mathbf{p} , number of diffusion steps T
 307 2: **Output:** Trained model parameters θ
 308 3: **for** $i = 1, 2, \dots$ **do**
 309 4: Embed \mathbf{p} to continuous field
 310 5: Sample $t \sim \text{Uniform}(\{1, \dots, T\})$, $\epsilon \sim \mathcal{N}(0, I)$, initialize \mathbf{z}_0
 311 6: Compute $\mathbf{z}_t = \sqrt{\alpha_t} \mathbf{z}_0 + \sqrt{1 - \alpha_t} \epsilon$
 312 7: Compute the predicted $\hat{\mathbf{z}}_0 = f_{\theta}(\mathbf{z}_t, t)$
 313 8: Compute loss $L_{train}(\mathbf{p})$
 314 9: Backpropagate and update model parameters θ
 315 10: **end for**
-

318 4.4 SAMPLING PROCESS

319 During the generation phase, the model begins with a noise vector \mathbf{z}_T randomly sampled from a
 320 Gaussian distribution. The process iteratively applies the learned denoising function to the noise vec-
 321 tor to generate realistic path. Specifically, at each time step t , the model uses the function $f_{\theta}(\mathbf{z}_t, t)$
 322 to predict \mathbf{z}_0 , and then samples of the distribution:

$$323 \quad \mathbf{z}_{t-1} \sim q(\mathbf{z}_{t-1} | f_{\theta}(\mathbf{z}_t, t), \mathbf{z}_t) \quad (10)$$

to obtain the next intermediate state. This iterative process continues until the model reaches the initial state z_0 . The final step is to decode z_0 back into the discrete path \hat{p} using the process $\text{argmax } p_\theta(\mathbf{p} \mid \hat{z}_0)$. This ensures that the generated path maps smoothly back into the discrete space, maintaining the structural and statistical properties of the original data.

Algorithm 2 Generating Process of DiffPath

```

1: Input: Number of diffusion steps  $T$ , trained model parameters  $\theta$ 
2: Output: Generated discrete path  $\hat{p}$ 
3: Random sampling  $z_T$ 
4: for  $t = T, T - 1, \dots, 1$  do
5:   Compute  $f_\theta(z_t, t)$ 
6:   Sample  $z_{t-1} \sim q(z_{t-1} \mid f_\theta(z_t, t), z_t)$ 
7: end for
8: Decode  $\hat{z}_0$  to discrete path  $\hat{p}$  using  $\text{argmax } p_\theta(\mathbf{p} \mid \hat{z}_0)$ 
9: return  $\hat{p}$ 

```

5 EXPERIMENTS

In this section, we provide the basic experimental setup, comparison of the main results, and visualization analysis. We conduct extensive experiments on two real-world pathsets to demonstrate the superior performance of the proposed model in synthetic path datasets generation.

5.1 EXPERIMENTAL SETUPS

5.1.1 DATASET

The raw datasets consist of GPS trajectories from two cities, referred to as Chengdu and Xi'an. The road network data for both cities was obtained from OpenStreetMap. Each road network is modeled as an directed graph. We then employed the map-matching algorithm proposed by (Meert & Verbeke, 2018) to align the GPS trajectory points to the road network. Consequently, the GPS trajectories were converted into path on the graph. For training, we randomly sampled 80% of the paths from each dataset, while 20% was set aside for testing. Please refer to the appendix B for more details

Table 2: Statistical Information of the Chengdu and Xi'an Datasets

City	Number of Vertices	Number of Paths	Average Path Length
Chengdu	2,865	91,070	24.737
Xi'an	2,675	63,110	24.921

5.1.2 BASELINES.

In order to directly evaluate the effectiveness of our generation model, we perform a comparative analysis of several path generation algorithms. The **N-gram** model, which estimates the probability of transition $p(v_t \mid v_{t-1}, \dots, v_{t-n+1})$ by counting the frequency of events, effectively captures local dependencies between paths. The Hidden Markov Model (**HMM**) (Yin et al., 2017) optimizes state reduction techniques to effectively simulate the underlying structure of path sequences. Additionally, the **MTNet** (Wang et al., 2022) model uses an architecture based on long-range memory (LSTM) to better capture long-range dependencies and complex sequential patterns in path. And the latest model **GDP** (Shi et al., 2024), which is modeled by the method of a discrete diffusion model. These comparisons provide a comprehensive assessment of the generative capabilities of our model relative to established methodologies.

5.1.3 EVALUATION METRICS

To evaluate the performance of our model, we employed several metrics to assess both local transitions and global structures in the generated paths. These metrics include **Kullback-Leibler Edge Visit Divergence (KLEV)**, **Jensen-Shannon Edge Visit Divergence (JSEV)** (Shi et al., 2024), and the newly introduced **Similarity Score (SS)**. **KLEV** measures the divergence between the edge visit frequency distributions of real and generated paths, capturing first-order transitions between adjacent nodes. It is defined as:

$$\text{KLEV}(P, P') = D_{\text{KL}}(\text{freq}(\forall(v_i, v_{i+1}) \in P) \parallel \text{freq}(\forall(v_i, v_{i+1}) \in P')), \quad (11)$$

where $\text{freq}(\cdot)$ calculates the visiting frequencies of all edges (v_i, v_{i+1}) across the real paths P and generated paths P' , and D_{KL} denotes the Kullback-Leibler divergence. Similarly, **JSEV** provides a more stable and symmetric alternative by replacing the Kullback-Leibler divergence with Jensen-Shannon divergence. It is given by:

$$\text{JSEV}(P, P') = \text{JS}(\text{freq}(\forall(v_i, v_{i+1}) \in P) \parallel \text{freq}(\forall(v_i, v_{i+1}) \in P')), \quad (12)$$

where $\text{JS}(p||q)$ denotes the Jensen-Shannon divergence. Like **KLEV**, lower **JSEV** values indicate better alignment between real and generated path distributions.

While **KLEV** and **JSEV** capture first-order transitions, they do not account for multi-hop dependencies between non-adjacent road segments. To better assess the overall similarity between the generated and real paths, we define the **Similarity Score (SS)** as follows:

$$S(G, R_{\text{best}}) = \frac{|\mathbb{P}(G) \cap \mathbb{P}(R_{\text{best}})|}{|\mathbb{P}(G)|}, \quad (13)$$

$\mathbb{P}(G)$ represents the set of adjacent node pairs in the generated path G , and R_{best} is the real path with the maximum overlap of adjacent node pairs with G . For a set of generated paths $\{G_1, G_2, \dots, G_h\}$, we compute the similarity score $S(G_i, R_{\text{best}})$ for each generated path G_i with its most similar real path R_{best} . The formula for the average similarity score across the entire set is given by:

$$S_{\text{avg}} = \frac{1}{h} \sum_{i=1}^h S(G_i, R_{\text{best}}), \quad (14)$$

where $S(G_i, R_{\text{best}})$ is the similarity score for the i -th generated path.

5.2 EXPERIMENTAL PERFORMANCE

5.2.1 OVERALL PERFORMANCE

Table 2 presents the performance comparison of our model and the selected baseline methods on two real-world datasets. We randomly generated 20,000 paths for each method and calculated all evaluation metrics. Our model outperforms other models based on neural networks in all metrics.

Table 3: Evaluations for Path generation.

City	Metrics	N-gram	HMM	GDP	MTnet	DiffPath (Ours)
Chengdu	SS	0.701	0.681	0.616	0.821	0.933
	KLEV	0.140	0.135	0.686	0.129	0.106
	JSEV	0.033	0.028	0.159	0.038	0.018
Xi'an	SS	0.628	0.633	0.571	0.772	0.893
	KLEV	0.133	0.130	0.697	0.127	0.122
	JSEV	0.031	0.025	0.147	0.033	0.023

We observe that for **KLEV** and **JSEV**, count-based models perform poorly as they depend on simple probability estimates for transitions between neighboring nodes. This limitation hinders their ability to capture rare or infrequent transitions, leading to lower performance on these metrics. Among the deep learning models, our approach achieves the lowest **KLEV** and **JSEV** values, indicating that the

distribution of the generated paths closely matches to the distribution of the real path distributions. All these observations demonstrate the effectiveness of **DiffPath**.

On **Similarity Score (SS)**, we can observe that **DiffPath** also achieves the SOTA compared to all other baselines. This shows that our model is especially good at generating paths that resemble real-world travel patterns. The transformer architecture plays a key role here by helping the model capture long-range dependencies across different road segments, ensuring that the generated paths reflect realistic travel routes over longer distances. In contrast, count-based models, which only focus on immediate transitions between adjacent nodes, struggle to represent the broader structure of paths. This limitation makes it difficult for these models to capture the full complexity of road networks, leading to lower SS values. As a result, while count-based models may produce paths that are locally correct, they fail to capture the overall travel patterns and structure seen in real-world data.

We also observed that the newly proposed **GDP** framework did not perform well on our dataset. This is primarily because GDP is designed for path planning tasks and focuses on conditional path generation. While it performs well in conditional scenarios, its unconditional path generation only ensures road network constraints without considering path distribution. Consequently, it performs poorly on metrics like KLEV and JSEV, which evaluate distributional characteristics.

5.2.2 VISUALIZATION ANALYSIS

We visualize the randomly selected real and generated paths in two cities. Figure 4 presents a visual comparison between the paths generated by **DiffPath** and the real-world paths for the cities of Chengdu and Xi'an. From these visualizations, it is evident that the generated paths align closely with the overall structure of the real paths, successfully capturing the major road networks in both cities. Key intersections and primary road segments are accurately represented in the generated paths, demonstrating the model's capacity to replicate real-world travel patterns.

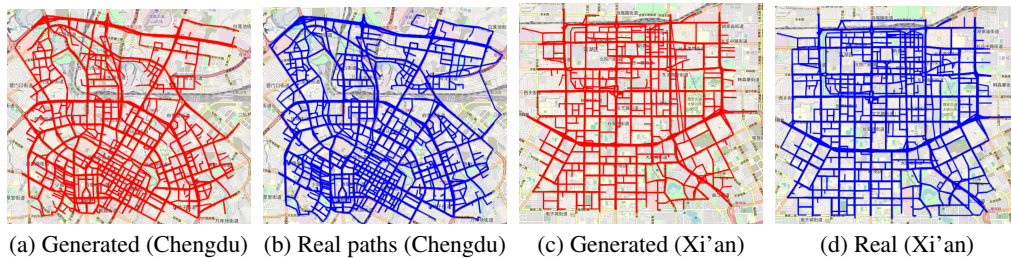


Figure 4: The visualization results of generated path and real path in Chengdu and Xi'an

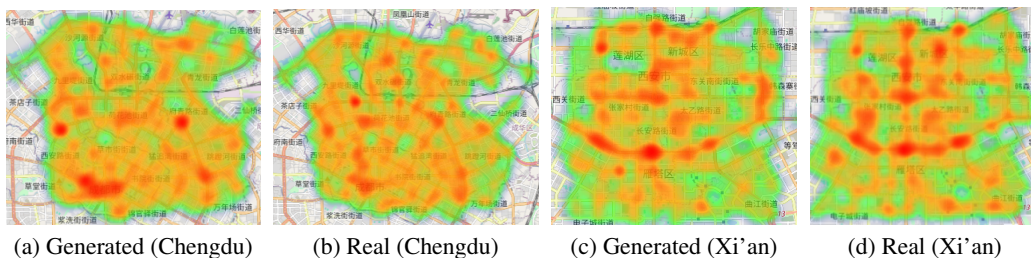


Figure 5: The visualized results of generated heat maps and real heat maps for Chengdu and Xi'an

To further evaluate the generated paths, Figure 5 compares the heatmaps of real and generated road segments for Chengdu and Xi'an. The heatmaps visualize the density and frequency of road segment usage, providing insights into the model's ability to capture the spatial distribution of traffic. In both cities, the generated heatmaps exhibit a distribution of high-traffic areas that closely mirrors the

actual heatmaps. This consistency indicates that the model effectively replicates high-frequency travel road segments observed in the real-world data, particularly in densely connected urban areas. More visualization results can be found in Appendix C.

5.2.3 ABLATION EXPERIMENT

For ablation experiment, we can see that the Transformer consistently performs better than UNet across all metrics in both cities from the results in Table 4. The higher Similarity Scores (SS) for the Transformer in Chengdu (0.933) and Xi’an (0.893) show that it captures the structure of the paths more accurately, both in terms of local connections and overall path flow. The lower KLEV and JSEV values for the Transformer also highlight its ability to model both common and rare road segments more effectively, addressing the issue of long-tail distributions. Overall, the Transformer aligns more closely with real-world paths, making it the better option for generating realistic paths in urban road networks compared to UNet.

5.2.4 PARAMETER EXPERIMENT

We observe that increasing the embedding dimension from 32 to 256 leads to significant improvements in all metrics. The results for both Chengdu and Xi’an are summarized in Table 5. For more detailed parameter settings, please refer to the appendix B

Table 4: Comparison of Transformer and UNet Architectures for Path Generation.

City	Metrics	UNet	Transformer
Chengdu	SS	0.915	0.933
	KLEV	0.122	0.106
	JSEV	0.023	0.018
Xi’an	SS	0.872	0.893
	KLEV	0.137	0.122
	JSEV	0.027	0.023

Table 5: Comparison of different embedding dimensions.

City	Metrics	32	64	128	256
Chengdu	SS	0.786	0.851	0.933	0.917
	KLEV	0.144	0.137	0.106	0.108
	JSEV	0.033	0.026	0.018	0.019
Xi’an	SS	0.708	0.816	0.893	0.887
	KLEV	0.144	0.137	0.122	0.125
	JSEV	0.033	0.026	0.023	0.023

6 CONCLUSION

In this work, we propose a path generation method based on a latent diffusion model (**DiffPath**). Our approach leverages the latent diffusion model’s strength in generating discrete data and the transformer’s ability to learn sequence features effectively. Specifically, real paths are first embedded into a latent space, where they undergo a forward noise trajectory process, gradually transforming them into random noise. The model then applies reverse trajectory denoising to reconstruct the paths from noise and finally decodes the synthesized paths. The effectiveness of **DiffPath** is validated through extensive experiments. Further experimental results show that the paths generated by the model align well with the statistical characteristics of real-world paths.

REFERENCES

- 540
541
542 Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of
543 natural images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*
544 *(CVPR)*. IEEE, June 2022. doi: 10.1109/cvpr52688.2022.01767. URL <http://dx.doi.org/10.1109/CVPR52688.2022.01767>.
545
- 546 Mitra Baratchi, Nirvana Meratnia, Paul JM Havinga, Andrew K Skidmore, and Bert AKG Toxopeus.
547 A hierarchical hidden semi-markov model for modeling mobility data. In *Proceedings of the 2014*
548 *ACM international joint conference on pervasive and ubiquitous computing*, pp. 401–412, 2014.
549
- 550 Simon Elias Bibri. Data-driven smart sustainable cities of the future: Urban computing and intel-
551 ligence for strategic, short-term, and joined-up planning. *Computational Urban Science*, 1(1):8,
552 2021.
- 553 Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object
554 detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*,
555 pp. 19830–19843, October 2023.
556
- 557 Seongjin Choi, Jiwon Kim, and Hwasoo Yeo. Trajgail: Generating urban vehicle trajectories using
558 generative adversarial imitation learning. *Transportation Research Part C: Emerging Technolo-*
559 *gies*, 128:103091, 2021.
- 560 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL
561 <https://arxiv.org/abs/2006.11239>.
562
- 563 Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto.
564 Diffusion-lm improves controllable text generation, 2022. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2205.14217)
565 [2205.14217](https://arxiv.org/abs/2205.14217).
566
- 567 Chuan Lin, Guangjie Han, Jiaxin Du, Tiantian Xu, Lei Shu, and Zhihan Lv. Spatiotemporal
568 congestion-aware path planning toward intelligent transportation systems in software-defined
569 smart city iot. *IEEE Internet of Things Journal*, 7(9):8012–8024, 2020.
- 570 Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian.
571 Diffusion augmentation for sequential recommendation, 2023. URL [https://arxiv.org/](https://arxiv.org/abs/2309.12858)
572 [abs/2309.12858](https://arxiv.org/abs/2309.12858).
573
- 574 Qingyue Long, Huandong Wang, Tong Li, Lisi Huang, Kun Wang, Qiong Wu, Guangyu Li, Yanping
575 Liang, Li Yu, and Yong Li. Practical synthetic human trajectories generation based on variational
576 point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery*
577 *and Data Mining*, pp. 4561–4571, 2023.
- 578 Zhaojun Lu, Gang Qu, and Zhenglin Liu. A survey on recent advances in vehicular network secu-
579 rity, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):760–776,
580 2019. doi: 10.1109/TITS.2018.2818888.
- 581 Wannes Meert and Mathias Verbeke. Hmm with non-emitting states for map matching. In *European*
582 *Conference on Data Analysis (ECDA), Paderborn, Germany, 2018*.
583
- 584 Anna Monreale, Roberto Pellungrini, et al. A survey on privacy in human mobility. *Transactions*
585 *on Data Privacy*, 16(1):51–82, 2023.
586
- 587 Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL
588 <https://arxiv.org/abs/2102.09672>.
- 589 Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew,
590 Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with
591 text-guided diffusion models, 2022. URL <https://arxiv.org/abs/2112.10741>.
592
- 593 William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of*
the IEEE/CVF International Conference on Computer Vision, pp. 4195–4205, 2023.

- 594 Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. Lstm-trajgan: A deep learning approach
595 to trajectory privacy protection. *arXiv preprint arXiv:2006.10521*, 2020.
596
- 597 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
598 resolution image synthesis with latent diffusion models, 2022. URL [https://arxiv.org/
599 abs/2112.10752](https://arxiv.org/abs/2112.10752).
- 600 Dingyuan Shi, Yongxin Tong, Zimu Zhou, Ke Xu, Zheng Wang, and Jieping Ye. Graph-constrained
601 diffusion for end-to-end path planning. In *The Twelfth International Conference on Learning
602 Representations*, 2024.
603
- 604 Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsuper-
605 vised learning using nonequilibrium thermodynamics, 2015. URL [https://arxiv.org/
606 abs/1503.03585](https://arxiv.org/abs/1503.03585).
- 607 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL
608 <https://arxiv.org/abs/2010.02502>.
609
- 610 Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben
611 Poole. Score-based generative modeling through stochastic differential equations, 2021. URL
612 <https://arxiv.org/abs/2011.13456>.
- 613 Richard S Sutton, Andrew G Barto, et al. Reinforcement learning. *Journal of Cognitive Neuro-
614 science*, 11(1):126–134, 1999.
- 615 Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navi-
616 gation. In *Conference on Robot Learning*, pp. 394–406. PMLR, 2020.
617
- 618 Yong Wang, Guoliang Li, Kaiyu Li, and Haitao Yuan. A deep generative model for trajectory
619 modeling and utilization. *Proceedings of the VLDB Endowment*, 16(4):973–985, 2022.
620
- 621 Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and
622 Yuxuan Liang. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion
623 models, 2024. URL <https://arxiv.org/abs/2301.13629>.
- 624 Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. Modeling trajectories with
625 recurrent neural networks. *IJCAI*, 2017.
- 626 Sean Bin Yang, Chenjuan Guo, Jilin Hu, Jian Tang, and Bin Yang. Unsupervised path representation
627 learning with curriculum negative sampling, 2021. URL [https://arxiv.org/abs/2106.
628 09373](https://arxiv.org/abs/2106.09373).
629
- 630 Sean Bin Yang, Jilin Hu, Chenjuan Guo, Bin Yang, and Christian S Jensen. Lightpath: Lightweight
631 and scalable path representation learning. In *Proceedings of the 29th ACM SIGKDD Conference
632 on Knowledge Discovery and Data Mining*, pp. 2999–3010, 2023.
- 633 Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-François Paiement, and Alexei Pozd-
634 noukhov. A generative model of urban activities from cellular data. *IEEE Transactions on Intel-
635 ligent Transportation Systems*, 19(6):1682–1696, 2017.
636
- 637 Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets
638 with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31,
639 2017.
- 640 Xiangyu Zhao, Wenqi Fan, Hui Liu, and Jiliang Tang. Multi-type urban crime prediction. In *Pro-
641 ceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4388–4396, 2022.
642
- 643 Yuanshao Zhu, Yongchao Ye, Yi Liu, and JQ James. Cross-area travel time uncertainty estimation
644 from trajectory data: a federated learning approach. *IEEE Transactions on Intelligent Transporta-
645 tion Systems*, 23(12):24966–24978, 2022.
- 646 Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. Difftraj: Generating
647 gps trajectory with diffusion probabilistic model. *Advances in Neural Information Processing
Systems*, 36, 2024a.

648 Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian
649 Zhang, Xuetao Wei, and Yuxuan Liang. Controltraj: Controllable trajectory generation with
650 topology-constrained diffusion model. *arXiv preprint arXiv:2404.15380*, 2024b.
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

A TRAINING OBJECTIVE DERIVATION

In the context of continuous diffusion models, we start with the goal of minimizing the difference between the true posterior mean $\hat{\mu}(x_t, x_0)$ and the predicted mean $\mu_\theta(x_t, t)$ in each diffusion step. The objective function is represented as:

$$\|\hat{\mu}(x_t, x_0) - \mu_\theta(x_t, t)\|^2. \quad (15)$$

Substituting the definitions of $\hat{\mu}(x_t, x_0)$ and $\mu_\theta(x_t, t)$, we have:

$$\hat{\mu}(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}x_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t,$$

$$\mu_\theta(x_t, t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}f_\theta(x_t, t) + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t.$$

Substituting these into the equation:

$$\|\hat{\mu}(x_t, x_0) - \mu_\theta(x_t, t)\|^2 \quad (16)$$

$$= \left\| \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}x_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t \right) - \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}f_\theta(x_t, t) + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t \right) \right\|^2 \quad (17)$$

$$= \left\| \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}(x_0 - f_\theta(x_t, t)) \right\|^2. \quad (18)$$

Factoring in the constant $\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}$:

$$= \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \right)^2 \|x_0 - f_\theta(x_t, t)\|^2. \quad (19)$$

Since $\left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \right)^2$ is a constant with respect to the parameters that are optimized, minimizing this expression is equivalent to the following.

$$\|x_0 - f_\theta(x_t, t)\|^2. \quad (20)$$

This final form highlights that the model is trained to predict x_0 directly at each diffusion step, which simplifies the optimization process and improves stability.

B IMPLEMENTATION DETAIL

B.1 CONFIGURATION

The paths in each dataset were selected based on recorded travel data, capturing real-world road usage across various regions of the cities. Each path consists of a sequence of connected vertices (road segments) representing actual vehicle routes. All experiments were implemented in PyTorch and conducted on a single NVIDIA GeForce RTX 3090 GPU.

B.2 HYPERPARAMETERS

The hyperparameters specific to **DiffPath** include the number of diffusion steps, the embedding dimension, and the noise schedule. We set the number of diffusion steps to 2000 and the sequence length to 144. For the embedding dimension, taking into account computing resource consumption, we experiment with values in $d \in \{16, 64, 128, 256\}$, choosing $d = 128$ for both city datasets.

DiffPath is trained using the AdamW optimizer with a linear learning rate decay starting at $1e-4$, a dropout rate of 0.1, and a batch size of 64. The total number of training iterations is set to 350K. For the noise schedule, we design a square-root (sqrt) schedule to better handle the diffusion process.

Figure 6 illustrates the evolution of the Kullback-Leibler Edge Visit Divergence (KLEV) and the Jensen-Shannon Edge Visit Divergence (JSEV) as the number of training iterations increases.

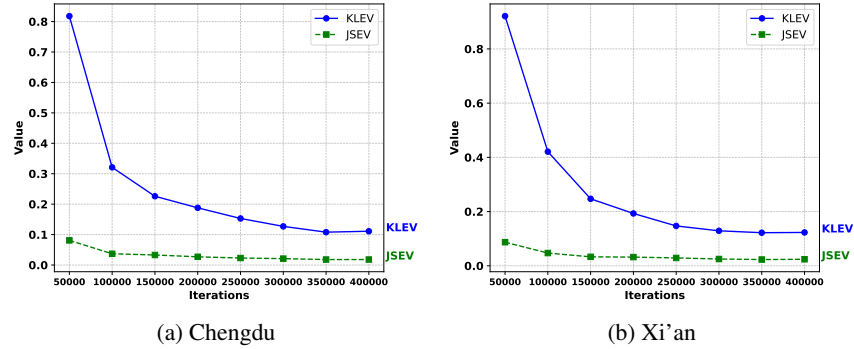


Figure 6: The evolution of the KLEV and JSEV as the number of training iterations increases.

C ANALYSIS OF VISUAL EXPERIMENT RESULTS

C.1 PATH LENGTH STATISTICS

We compared the frequency distribution of path lengths between the generated and real-world paths. Figure 10 presents the path length frequency distributions for both the generated and real-world paths in Chengdu and Xi'an. The visual comparison demonstrates that the length distribution of the generated paths is generally consistent with that of the real paths. However, slight deviations are observed, particularly for longer paths, where the generated paths exhibit a somewhat narrower distribution compared to the real paths. This suggests that while the model effectively generates paths of typical lengths, there may be room for improvement in generating longer, less frequent paths. Despite these minor discrepancies, the overall similarity between the length distributions indicates that the model performs well in replicating the key characteristics of the real data.

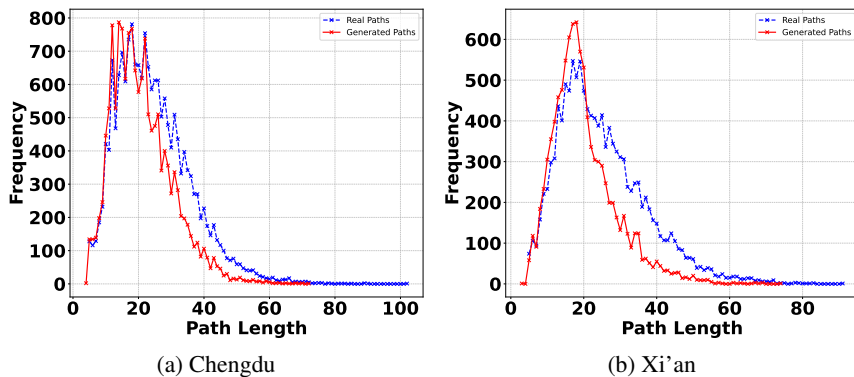


Figure 7: Comparison of Real and Generated Path Length Distributions

C.2 PATH DISTRIBUTION ADDITIONAL STATISTICS

To evaluate the effectiveness of our model in learning path distributions, we conducted additional experiments and visualized the results. Specifically, we divided the dataset’s geographical area into a 3×3 grid, resulting in nine regions. We then calculated the distribution of path starting and ending points. For instance, a path starting in region 1 and ending in region 2 is recorded in cell (1, 2) of the grid. The distributions of paths generated by the real dataset and DiffPath are compared as shown below.

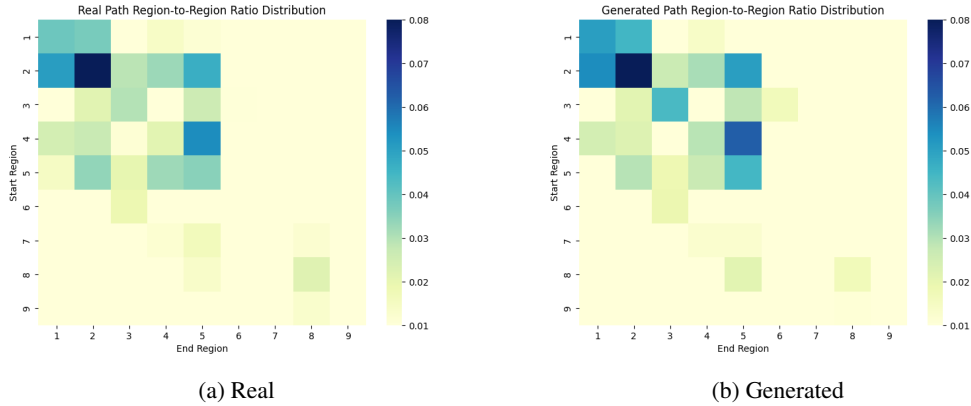


Figure 8: Path distribution statistics based on origin and destination flow(Chengdu)

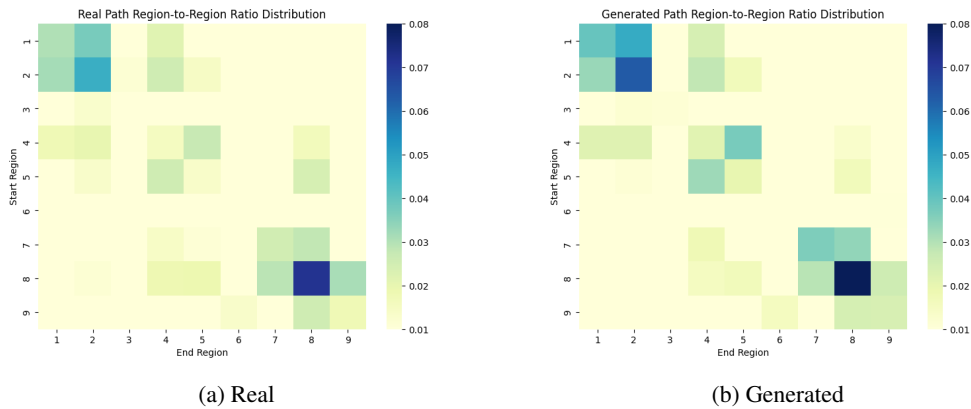


Figure 9: Path distribution statistics based on origin and destination flow(Xi'an)

The visualization demonstrates that the statistical grid of paths generated by DiffPath closely resembles that of the real dataset, indicating that the overall distribution of paths generated by DiffPath aligns well with the real data. Furthermore, in combination with the results presented in Table 3, we can confirm that DiffPath effectively captures the real-world path distribution.

C.3 LOW FREQUENCY ROAD SEGMENTS STATISTICS

To assess the model’s capability in capturing complex path distributions, particularly low-frequency paths, we conducted additional experiments. In these experiments, we determined the average number of road segments in the Chengdu and Xi’an datasets to be 633.90 and 473.77, respectively. Road segments that appeared fewer than 50 times in the real data were categorized as low-frequency

segments. The generation proportions of these low-frequency segments across various models are presented in the figures below.

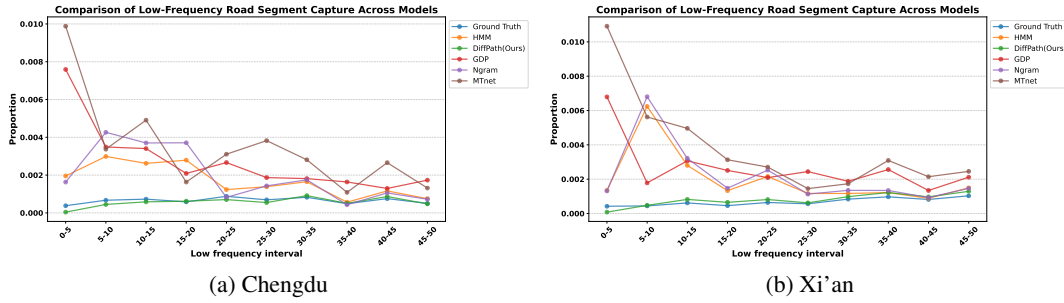


Figure 10: Comparison of results generated by low-frequency road segments

The results demonstrate that the generation ratios of our model align most closely with the trend line of the real dataset, indicating its superior performance in accurately capturing the distribution of low-frequency road segments within the real data.