Map2Traj: Street Map Piloted Zero-shot Trajectory Generation Method for Wireless Network Optimization

Zhenyu Tao[1,2], **Wei Xu**[1,2], **Xiaohu You**[1,2*]
[1]National Mobile Communications Research Lab, Southeast University, Nanjing, China
[2]Pervasive Communication Research Center, Purple Mountain Laboratories, Nanjing, China
{zhenyu_tao, wxu, xhyu}@seu.edu.cn

Abstract

In modern wireless networks, user mobility modeling plays a pivotal role in learning-based network optimization, particularly in tasks such as user association and resource allocation. Traditional random mobility models, e.g., random waypoint and Gauss Markov model, often fail to accurately capture the distribution patterns of users within real-world areas. While trace-based mobility models and advanced learning-based trajectory generation methods offer improvements, they are frequently limited by the scarcity of real-world trajectory data in target areas, primarily due to privacy concerns. This paper introduces Map2Traj, a novel zero-shot trajectory generation method that leverages the diffusion model to capture the intrinsic relationship between street maps and user mobility. With solely the street map of an unobserved area, Map2Traj generates synthetic user trajectories that closely resemble the real-world ones in trajectory pattern and spatial distribution. This enables the creation of high-fidelity individual user channel states and an accurate representation of the overall network user distribution, facilitating effective wireless network optimization. Extensive experiments across multiple regions in Xi'an and Chengdu, China demonstrate the effectiveness of our proposed method for zero-shot trajectory generation. A case study applying Map2Traj to user association and load balancing in wireless networks is also presented to validate its efficacy in network optimization.

1 Introduction

With the long-term evolution of cellular networks in terms of heterogeneity, density, and multi-band usage, the accuracy of user mobility modeling has become increasingly crucial for performance evaluation and optimization of wireless communication networks. In the realm of learning-based network optimization, involving resource management [Naderializadeh *et al.*, 2021], user association [Gupta *et al.*, 2024], and edge computing [Xu *et al.*, 2023], user mobility models stand as the cornerstone for constructing virtual environments and digital twins [Tao *et al.*, 2024] for training artificial intelligence (AI) models, for instance, the deep reinforcement learning (DRL) agents.

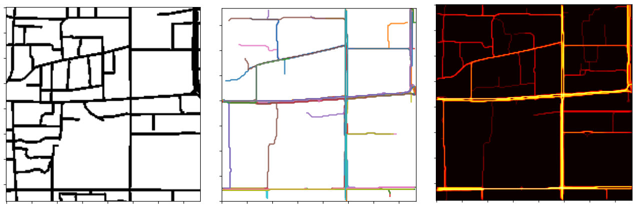
Most existing network optimization studies [Zhao *et al.*, 2019; Gupta *et al.*, 2021; Jia and Wang, 2023] employed random mobility models, typically the random waypoint model [Johnson and Maltz, 1996] and the Gauss Markov model [Liang and Haas, 1999], to represent user mobility patterns. While these models can partially simulate user movement, their direct adherence to random probability distributions causes a significant mismatch in the spatial distribution of users compared to real-world scenarios. This mismatch can lead to considerable performance degradation when deploying these models in practice [Feriani and Hossain, 2021]. Although this issue can be alleviated by incorporating real trace-based mobility models to some extent, user trajectories are often unfortunately inaccessible due to data acquisition costs and privacy concerns [Tabassum *et al.*, 2019].

In recent years, AI models such as generative adversarial network (GAN) [Goodfellow *et al.*, 2020] and diffusion model [Ho *et al.*, 2020] have been applied to trajectory generation with promising results [Zhu *et al.*, 2023b]. However, these methods typically require a substantial number of real trajectories to learn the specific trajectory distribution within a given area, creating a **paradox**. That is, AI models struggle to capture user mobility without ample real trajectories, yet when real data becomes sufficient to create trace-based mobility models, the AI models tend to be redundant for wireless network optimization.

Inspired by zero-shot image generation [Ramesh *et al.*, 2021], which enables the creation of images from descriptions unseen during training, we try to devise a similar approach for trajectory generation. This method, termed **zero-shot trajectory generation**, aims to generate realistic user trajectories for unobserved areas. The question arises: Is there some form of auxiliary data that is both readily accessible and closely related to real user trajectories, akin to the relationship between texts and images? Our answer is the street map, which is usually open-source and available on platforms like OpenStreetMap[1]. Street maps exhibit a strong correlation with user trajectories as illustrated in Figure 1, including

*Corresponding author

[1]https://www.openstreetmap.org/



(a) Street map (b) Trajectories (c) Heatmap

Figure 1: Correlation between street map and user trajectories

a street map for a specific area, 100 trajectories in this area, each assigned a color, and a heatmap of the trajectory distribution. This correlation forms the foundation of our proposed methodology, which uses street maps as pilots in trajectory generation.

Building on these motivations and the observed correlation, we propose a street-**Map**-pilot zero-shot **Traj**ectory generation (Map2Traj) method. With solely a street map input, Map2Traj generates synthetic user trajectories similar to real ones in both *trajectory pattern* and *spatial distribution*. Such similarities are pivotal for optimizing wireless networks, as they enable learning-based methods, e.g., DRL, trained in Map2Traj-based environments to be effectively deployed in real-world scenarios with minimal performance degradation caused by mobility discrepancy. To summarize, the contributions of this work are as follows.

- We develop the Map2Traj method for zero-shot trajectory generation, leveraging the diffusion model to generate real-world user trajectories solely based on street maps. To the best of our knowledge, this is the first work to achieve zero-shot trajectory generation, especially for mobile users in wireless networks.
- We validate the efficacy of Map2Traj through comprehensive experiments, proving that our method can generate high-fidelity trajectories for areas beyond the training set, with considerable similarity to real trajectories in both trajectory pattern and spatial distribution.
- We examine the efficacy of Map2Traj in a network optimization task, specifically the user association and load balancing in a multi-cell and multiuser wireless communication network. The results indicate that the Map2Traj-based mobility model significantly outperforms traditional random mobility models and exhibits nearly the same efficacy as the model using real trajectories.

2 Related Work

Before introducing our proposed method, we review related works on trajectory generation and analyze the limitations that prevent these methods from achieving zero-shot trajectory generation in wireless network optimization.

Initially, trajectory generation methods were developed to synthesize mobility data and safeguard the privacy of data providers. Liu *et al.* [Liu *et al.*, 2018] first proposed to use GANs for trajectory generation, albeit without providing a detailed approach. TrajGAIL [Zhang *et al.*, 2020] employed generative adversarial imitation learning (GAIL), combining DRL and GAN to generate trajectories through a series of next-location predictions. TrajGen [Cao and Li, 2021] transformed trajectories into images and used a deep convolutional GAN (DCGAN) to generate virtual trajectory images. TS-TrajGen [Jiang *et al.*, 2023] integrated GAN with the mobility analysis method, including the A* algorithm and mobility yaw reward, to enhance the model performance. DiffTraj (SynMob) [Zhu *et al.*, 2023a; Zhu *et al.*, 2023b] applied a diffusion model to generate synthetic trajectories while preserving spatial-temporal features extracted from real trajectories. These studies have demonstrated commendable performance in generating privacy-preserving synthetic trajectories.

However, these methods fall short when it comes to zero-shot trajectory generation for unobserved new areas. Specifically, TrajGAIL simply samples actions from generated action probability distribution and constructs trajectory autoregressively, without the capacity to introduce data from new areas. Although TrajGen uses street map data to filter and calibrate generated trajectories through map matching [Newson and Krumm, 2009], the generated trajectories adhere to the training set distribution, rather than that in new areas. Building further upon TrajGen, TS-TrajGen utilizes street maps to select and construct the best continuous trajectory with the A* algorithm. This process, however, remains confined to trajectories that adhere to the original distribution. Alternatively in a conditional generation manner, DiffTraj employs the diffusion model and incorporates prior knowledge of trip data, such as the travel time, average speed, and distance. While these complementary knowledge do improve generation performance, they do not enable the transfer of trajectory generation to new areas.

In contrast to these approaches, our Map2Traj method integrates the street map with complete information into the trajectory generation process via a diffusion model. Our training set encompasses a diverse range of trajectories and corresponding street maps from various areas, instead of area-specific trajectories, allowing the model to learn the intrinsic relationship between street maps and trajectories. These innovations endow our model with the unique capability for zero-shot trajectory generation. A comparative analysis of our method against existing works is detailed in Table 1.

3 Preliminary

Map2Traj is primarily designed for wireless network optimization and differs from conventional trajectory generation approaches in important ways. Specifically, it focuses on the fidelity of trajectory pattern and spatial distribution, which are critical for shaping both individual user channel states and overall network user distribution. In terms of traffic patterns, this paper concentrates on heavy-load traffic scenarios, e.g., the rush hour, where network optimization is required to address potential congestion and load imbalances. Additionally, given the spatial consistency of wireless channels [Huang *et al.*, 2022], fine-grained trajectory details are less critical, especially for large-scale network optimization tasks such as user association and resource allocation. As a result,

Method	Model	Training data	Inference data	Zero-shot generation
TrajGAIL (ICDM)	GAIL	Trajectories	Random sampling	✗
TrajGen (KDD)	DCGAN	Trajectories + maps	Noise + maps	✗
TS-TrajGen (AAAI)	GAN	Trajectories + maps	Noise + maps	✗
DiffTraj (NIPS)	Diffusion model	Trajectories + trip data	Noise + trip data	✗
Map2Traj (Proposed)	Diffusion model	Trajectories + maps	Noise + maps	✓

Table 1: Comparison of trajectory generation methods

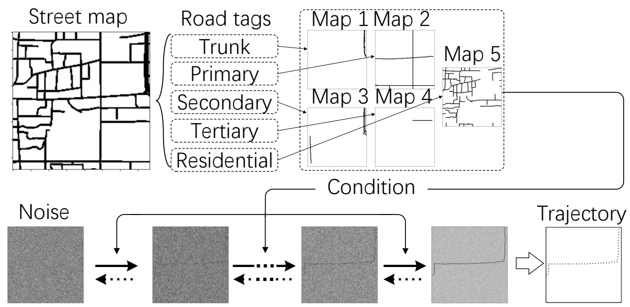


Figure 2: Map2Traj framework.

this paper discretizes the trajectory within a 1.92 km×1.92 km region into a 192×192 image, emphasizing macroscopic location changes. Nevertheless, the proposed method is applicable to other traffic patterns, such as those in mid-night, and is scalable to larger areas (e.g., 256×256). Relevant experiments and results are provided in the technical appendix. We now define the key terms and notations used in this paper.

Trajectory: In this paper, the trajectory is defined by a series of coordinates $\{c_1, c_2, \dots, c_n\}$ restricted in a 1.92 km×1.92 km urban area, where each c_i represents a 10-meter-spaced point. This discrete trajectory sequence can be easily transformed into a 192×192 image, denoted by $\boldsymbol{l}$, for further processing, while the sequence can also be reconstructed from the image [Endo *et al.*, 2016].

Street Map: A street map is conventionally denoted as a graph, where edges correspond to road segments and nodes to road junctions. To align with the trajectories, we also convert each street map into an image of the same 192×192 dimension, denoted by $\boldsymbol{m}$.

Problem Statement: The training set includes a set of street maps $\mathcal{M} = \{\boldsymbol{m}^1, \boldsymbol{m}^2, \dots\}$ and corresponding sets of real-world trajectories $\mathcal{T} = \{\mathcal{L}^1, \mathcal{L}^2, \dots\}$. Each $\mathcal{L}^i = \{\boldsymbol{l}^{i,1}, \boldsymbol{l}^{i,2}, \dots\}$ is a set of real trajectories within the area of the street map $\boldsymbol{m}^i$. The objective of zero-shot trajectory generation is to develop a generative model trained on this training set. For an unobserved street map $\boldsymbol{m}^o \notin \mathcal{M}$, this model should be capable of generating synthetic trajectories that closely resemble real ones in trajectory pattern and spatial distribution, as well as have an efficacy close to that of real trajectories in network optimization.

4 The Map2Traj Approach

The key to zero-shot learning is to associate observed and unobserved objects through some form of auxiliary information, which encodes the inherent properties of objects [Xian *et al.*, 2017]. In our study, the objects are trajectories following different area-specific distributions, while the auxiliary information is the street map. By learning the relationship between trajectories and maps from extensive training data, Map2Traj generates synthetic trajectories for unobserved areas based on the street map, i.e., the zero-shot trajectory generation.

In particular, our Map2Traj approach is based on the diffusion model, consisting of a forward diffusion process and a reverse diffusion process (denoising) for generation. By integrating trajectories from various areas and utilizing relevant street maps as conditional inputs, Map2Traj extends the original single target distribution in diffusion model into multiple area-specific target distributions corresponding to given street maps. As a result, Map2Traj can estimate the trajectory distribution through an unobserved street map and generate synthetic trajectories that conform to this distribution through sampling. The entire process is illustrated in Figure 2. For subsequent network optimization, the generated image undergoes post-processing to reconstruct the trajectory sequence.

4.1 Forward Diffusion Process

The forward diffusion process in Map2Traj is a Markovian process that iteratively adds Gaussian noise $\mathcal{N}(\cdot)$ to a trajectory data $\boldsymbol{l}_0 \equiv \boldsymbol{l}$ over T time steps:

$$q\left(\boldsymbol{l}_{t+1} \mid \boldsymbol{l}_t\right) = \mathcal{N}\left(\boldsymbol{l}_{t+1}; \sqrt{\alpha_t}\boldsymbol{l}_t, (1-\alpha_t)\mathbf{I}\right), \tag{1}$$

$$q\left(\boldsymbol{l}_{1:T} \mid \boldsymbol{l}_0\right) = \prod_{t=1}^{T} q\left(\boldsymbol{l}_t \mid \boldsymbol{l}_{t-1}\right), \tag{2}$$

where α_t for $t = 1, 2, \dots, T$ are hyper-parameters of the noise schedule, and $\mathcal{N}(x; \mu, \sigma)$ represents the normal distribution of mean μ and covariance σ that produces x. The forward process with α_t is constructed to make $\boldsymbol{l}_T$ virtually indistinguishable from Gaussian noise at the T-th step. The forward process at the t-th step can also be marginalized as follows:

$$q\left(\boldsymbol{l}_t \mid \boldsymbol{l}_0\right) = \mathcal{N}\left(\boldsymbol{l}_t; \sqrt{\gamma_t}\boldsymbol{l}_0, (1-\gamma_t)\mathbf{I}\right), \tag{3}$$

where $\gamma_t = \prod_{i=1}^{t} \alpha_i$. Additionally, the parameterization of the Gaussian distribution of the forward process allows a closed-form formulation of the posterior distribution of $\boldsymbol{l}_{t-1}$ given $(\boldsymbol{l}_0, \boldsymbol{l}_t)$. It follows

$$q\left(\boldsymbol{l}_{t-1} \mid \boldsymbol{l}_0, \boldsymbol{l}_t\right) = \mathcal{N}\left(\boldsymbol{l}_{t-1}; \boldsymbol{\mu}, \sigma^2\mathbf{I}\right), \tag{4}$$

where $\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t}\boldsymbol{l}_0 + \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t}\boldsymbol{l}_t$ and $\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}$.

4.2 Reverse Diffusion Process

In Map2Traj, the reverse diffusion process, also known as the denoising process, is formulated as follows:

$$p_{\boldsymbol{\theta}}\left(\boldsymbol{l}_{0:T} \mid \boldsymbol{m}\right) = p\left(\boldsymbol{l}_T\right) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}\left(\boldsymbol{l}_{t-1} \mid \boldsymbol{l}_t, \boldsymbol{m}\right), \tag{5}$$

where $p(\boldsymbol{l}_T) = \mathcal{N}(\boldsymbol{l}_T; 0, \mathbf{I})$. Map2Traj undergoes training and performs inference through the reverse diffusion process.

Training

Given a noisy trajectory $\boldsymbol{l}_t$ sampling from Eq. (3), we have

$$\boldsymbol{l}_t = \sqrt{\gamma_t}\boldsymbol{l}_0 + \sqrt{1-\gamma_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{6}$$

where the goal is to recover the target trajectory $\boldsymbol{l}_0$. Our neural network model is parameterized by $f_{\boldsymbol{\theta}}(\boldsymbol{m}, \boldsymbol{l}_t, t)$, conditioned on the street map $\boldsymbol{m}$, a noisy trajectory $\boldsymbol{l}_t$, and the noise level indicated by the time step t. Training of Map2Traj involves predicting the noise vector $\boldsymbol{\epsilon}$ by minimizing the mean squared error loss. That is,

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{m},\boldsymbol{l},t,\boldsymbol{\epsilon}} \| f_{\boldsymbol{\theta}}(\boldsymbol{m}, \underbrace{\sqrt{\gamma_t}\boldsymbol{l}_0 + \sqrt{1-\gamma_t}\boldsymbol{\epsilon}}_{\boldsymbol{l}_t}, t) - \boldsymbol{\epsilon} \|^2. \tag{7}$$

Inference

The sampling process of the diffusion model starts at pure Gaussian noise $\boldsymbol{l}_T$, followed by T refinement steps. Given any noisy trajectory $\boldsymbol{l}_t$, we can approximate the target trajectory by rearranging the terms in Eq. (6) as

$$\hat{\boldsymbol{l}}_0 = \frac{1}{\sqrt{\gamma_t}}\left(\boldsymbol{l}_t - \sqrt{1-\gamma_t} f_{\boldsymbol{\theta}}\left(\boldsymbol{m}, \boldsymbol{l}_t, t\right)\right). \tag{8}$$

Substituting estimate $\hat{\boldsymbol{l}}_0$ into Eq. (4), we parameterize the mean of $p_{\boldsymbol{\theta}}\left(\boldsymbol{l}_{t-1} \mid \boldsymbol{l}_t, \boldsymbol{m}\right)$ in Eq. (5) as

$$\mu_{\boldsymbol{\theta}}\left(\boldsymbol{m}, \boldsymbol{l}_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{l}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_{\boldsymbol{\theta}}(\boldsymbol{m}, \boldsymbol{l}_t, t)\right). \tag{9}$$

And the variance of $p_{\boldsymbol{\theta}}\left(\boldsymbol{l}_{t-1} \mid \boldsymbol{l}_t, \boldsymbol{m}\right)$ is approximated as $(1-\alpha_t)$, following the setting in [Ho *et al.*, 2020]. With this parameterization, the sampling can be executed iteratively as follows:

$$\boldsymbol{l}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{l}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_{\boldsymbol{\theta}}\left(\boldsymbol{m}, \boldsymbol{l}_t, t\right)\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \tag{10}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

4.3 Architecture of Map2Traj

The architecture of Map2Traj is based on a classic U-Net model [Ronneberger *et al.*, 2015], with multiple modifications to improve its performance such as attention blocks [Oktay *et al.*, 2018] and group normalization [Wu and He, 2018]. A distinctive feature of Map2Traj is the incorporation of street map data through concatenation, which allows the model to be conditioned on the spatial information inherent in the maps. The architecture is depicted in Figure 3.

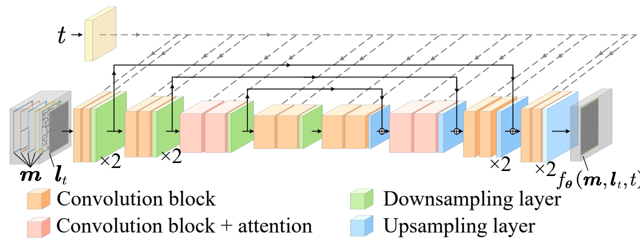


Figure 3: Map2Traj architecture.

4.4 Other Technical Details

Street Map Splitting

While a single binary image can effectively convey the spatial layout of a street map, it falls short in depicting the distinct characteristics of various road types. In the OpenStreetMap dataset, roads are tagged with attributes such as *Trunk*, *Primary*, and *Residential*. To exploit these attributes, we categorize roads into multiple groups, create binary images for each group, and merge them into a multi-channel binary image, as illustrated in Figure 2.

Data Augmentation

We notice that the correlation between street map and trajectory, as shown in Figure 1, remains consistent under transformations such as rotation and reflection. This inherent property can be leveraged for data augmentation during training. We randomly rotate and flip both street maps and trajectory data to enhance the generalization capability of Map2Traj.

Post Processing of Trajectory Image

The trajectory sequence needs to be reconstructed from the image to model user mobility for wireless network optimization. Since each image represents only one trajectory per user, we used a simple search algorithm to reconstruct the sequence from one end to the other, where each point represents the coordinate of the pixel center. An extra deep neural network is utilized to estimate the sojourn time at each point, as utilized in [Endo *et al.*, 2016] and [Jiang and Fei, 2017], from which the speed and relative time stamp can be derived.

5 Experiments

In this section, we evaluate the efficacy of Map2Traj in the zero-shot trajectory generation task by comparing the fidelity of generated user trajectories with real ones. In addition, we employ Map2Traj in a typical wireless network optimization task to validate its efficacy.

5.1 Dataset Description

Map2Traj is trained on a real-world trajectory dataset from Xi'an, China, recorded in 2016 [Didi-Chuxing, 2017], alongside the OpenStreetMap dataset from the same year. The training set is restricted to longitudes between 108.912 and 108.974, while the test set spans from 108.974 to 108.996 to avoid data leakage. Regions in Chengdu, China are also used to evaluate the zero-shot generation performance of

Map2Traj across different geographical layouts and urban planning. All datasets are sourced from the ChinaGEOSS[2].

5.2 Evaluation Metrics

We employ a suite of metrics to evaluate the quality of generated trajectories, in terms of similarities in both trajectory pattern and spatial distribution.

Trajectory Pattern Similarity

- **Edit Distance on Real Sequences (EDR):** EDR [Chen *et al.*, 2005] quantifies the minimum number of operations required to make two trajectories match. A match is defined when the distance between corresponding points is less than a threshold of $\tau = 20$ meters.
- **Dynamic Time Wrapping (DTW):** DTW [Berndt and Clifford, 1994] calculates the squared Euclidean distance between two trajectories through a dynamic programming alignment algorithm.

Both metrics are widely used in mobility analysis [Tao *et al.*, 2021]. Since EDR and DTW are designed for sequential data, the trajectory sequences are reconstructed from images for comparison. Since the direction of the trajectory is not discernible from the image, both the original and reversed sequences are evaluated, and the minimum value is selected as the final result.

Spatial Distribution Similarity

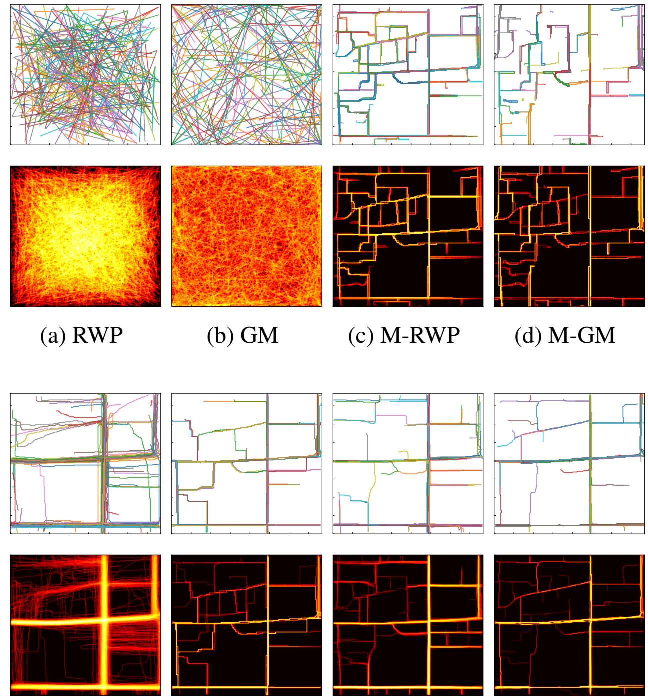
- **Cosine Similarity:** Cosine similarity is a widely used measure of similarity between two vectors. While it reflects the similarity between probability distributions, it falls short in expressing the spatial correlation between adjacent blocks in two-dimensional (2D) distributions.
- **Wasserstein Distance:** To overcome the limitations of cosine similarity, we introduce the Wasserstein distance [Rüschendorf, 1985], defined as the cost of the optimal transport plan for moving the mass in the predicted distribution to match that in the target. In this context, it quantifies the effort needed to align the spatial distribution of generated trajectories with that of real trajectories. For improved computational efficiency, we use the sliced Wasserstein distance instead [Kolouri *et al.*, 2019].

5.3 Baseline Methods

For comparison, we use several traditional random mobility models, along with two real trajectory-based methods, despite their inability to zero-shot trajectory generation.

For traditional random mobility models, the most commonly used one in wireless network optimization is the random waypoint model (RWP), where the trajectory is formed by constantly moving to a randomly chosen destination and then selecting the next one arbitrarily. To adapt this model to geographical constraints, we develop a variant, termed map-restricted random waypoint (M-RWP), where the destinations are confined within the street map area. The trajectory between points is determined using a breadth-first search (BFS) algorithm to ensure that the shortest path remains within streets. Additionally, we considered the Gauss Markov model (GM), characterized by using a stochastic process to model changes in user velocity and direction. Similarly, we introduce the map-restricted Gauss Markov model (M-GM) to restrict user movements within street areas.

In terms of the real trajectory-based ones, the state-of-the-art trajectory generation model, DiffTraj [Zhu *et al.*, 2023b], is included to benchmark the generation quality of our proposed model. It is important to note that for this comparison, DiffTraj was trained on the complete trajectory dataset [Didi-Chuxing, 2017], including those from the test area. The relevant data is sourced from the DiffTraj-generated synthetic dataset, SynMob, provided by the authors of DiffTraj in [Zhu *et al.*, 2023a]. An original-destination (OD) method is also used, where the origin and destination locations are sampled from real trajectories, and the trajectory is determined by the shortest path calculated using the BFS algorithm.



(a) RWP (b) GM (c) M-RWP (d) M-GM

(e) DiffTraj (f) OD method (g) Map2Traj (h) Real data

Figure 4: Trajectory generation in Xi'an (Top: Trajectory; Bottom: Heatmap).

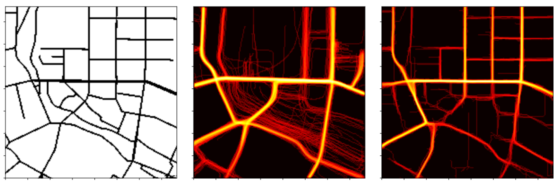
5.4 Generation Performance

We select the area depicted at the beginning of this paper as the test area. Figure 4 displays the generated trajectories by all methods alongside corresponding heatmaps. Traditional random mobility models, i.e., RWP and GM, result in chaotic trajectories and heatmaps that bear no resemblance to real-world patterns. While map-restricted models, M-RWP and M-GM, show some similarity in trajectory pattern to real ones, they still fall short in distribution similarity

[2]https://chinageoss.cn/

Category	Method	Trajectory pattern similarity		Spatial distribution similarity	
		EDR ($\tau = 20$) ↓	DTW ↓	Cosine similarity ↑	Wasserstein distance ↓
Random mobility	RWP	264.1	76.74	0.1537	21.01
	GM	213.1	82.96	0.1698	19.21
	M-RWP	192.0	21.44	0.3081	22.91
	M-GM	155.4	33.57	0.2793	26.32
Zero-shot	Map2Traj	**21.47**	**8.933**	**0.6834**	**6.096**
Real trajectory-based	DiffTraj	68.35	13.63	0.5573	9.134
	OD method	30.12	9.064	0.5531	6.147
Real trajectories		7.570	1.018	0.9959	2.569

Table 2: Quantified evaluation of trajectory generation performance



(a) Street map (b) DiffTraj (c) Map2Traj

Figure 5: Trajectory generation in Chengdu.

due to the absence of a learning mechanism. As expected, real trajectory-based methods, including DiffTraj and the OD method, demonstrate high similarity to real trajectories and heatmaps. Our proposed Map2Traj, even in a zero-shot scenario, generates results that closely match real trajectories and, in some cases, outperform DiffTraj in trajectory detail. Additional experimental results, including data from other regions in the two cities, varied traffic patterns, and higher resolutions, are provided in the technical appendix.

Further, we quantify the similarity of the generated trajectories with real ones using the metrics mentioned earlier. Considering the stochastic nature of trajectory generation, we generate 1,000 trajectories per method and compare them to a benchmark of 1,000 real trajectories. For each generated trajectory, we calculate the EDR and DTW to all real trajectories and select the minimum value as the representative measure. This process is repeated for all 1,000 generated trajectories to calculate the average similarity between the generated and real sets. The spatial distribution is calculated by summing and normalizing the binary trajectory images. An additional set of 1,000 real trajectories is included in the calculation to represent the optimal similarities. Table 2 presents the quantified similarity comparison among different trajectory sets. The results indicate that our proposed Map2Traj significantly exceeds the traditional random mobility models, producing synthetic trajectories closely resembling real ones in both trajectory and distribution similarities. This suggests that Map2Traj has effectively learned the correlation between street maps and actual trajectories.

It is encouraging to see our zero-shot Map2Traj model outperforms the real trajectory-based DiffTraj and OD method. The advantage of Map2Traj lies in the continuous guidance from street maps throughout the denoising process, while DiffTraj only involves some trip information as the condition. Also, DiffTraj fixes the trajectories into uniform shapes through sampling, instead of transforming them into images, potentially leading to information loss. However, real trajectory-based methods, such as DiffTraj, have the potential to outmatch Map2Traj given a sufficiently large training dataset, a wider and deeper network structure, or in some specific test scenarios. It should be noted that the primary contribution of this work is the development of a zero-shot trajectory generation method, rather than merely surpassing existing real-trajectories-based generation techniques.

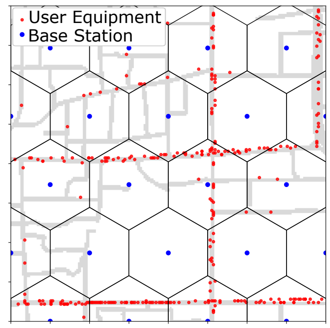


Figure 6: User association and load balancing scenario.

5.5 Case Study in Wireless Network Optimization

In this case study, we employ Map2Traj in wireless network optimization, specifically the user association and load balancing task.

System Model and Task Overview

We consider a typical urban area, i.e., the test area in Figure 4, where base stations are densely deployed in a hexagonal pattern, with a 500 m interval. Each base station possesses multi-band capabilities, supporting connections at 3.7 GHz with a 40 MHz bandwidth and 0.7 GHz with a 10 MHz bandwidth. As illustrated in Figure 6, users move continuously within this area, connecting to base stations based on a specific user association policy. Traditional user association methods that maximize signal-to-interference-plus-noise ratio (SINR) can lead to load imbalances and frequent handovers, resulting in user rate degradation when users are not uniformly distributed. An advanced user association strategy is required to balance network loads and minimize handovers, thereby enhancing user experience and connection stability.

Mehthodology

Solving the user association problem through integer linear programming is unfeasible due to real-time constraints and the computational complexity introduced by a large number of users, along with the dynamics of mobility and handover. In this context, current works primarily employ DRL to address this problem. However, the direct training of DRL agents within the real-world wireless network is fraught with challenges, including prohibitive trial-and-error costs and the risk of compromising the quality of service [Feriani and Hossain, 2021]. Consequently, constructing a realistic training environment is crucial for applying DRL in wireless network optimization. While there is extensive research on wireless channel measurement and modeling, studies on user mobility models are limited. The current use of random mobility models, such as RWP and GM, does not ensure the efficacy of DRL agents when applied to real-world scenarios. Map2Traj is utilized as the user mobility model in training environments to enhance the performance of DRL agents.

Case Study Setup and Metrics

The main purpose of this case study is to investigate whether DRL agents trained with traditional random mobility models experience performance degradation when deployed into real environments, and whether incorporating Map2Traj can mitigate this issue. The case study consists of two phases. Initially, DRL agents are trained in environments based on various mobility models, including Map2Traj and baselines. After achieving convergence, these agents are deployed into the real environment where user movements adhere to real trajectories to assess performance. The performance of a DRL agent trained directly in the real environment is also provided as a benchmark for optimal performance.

To focus on mobility model comparisons, wireless channels are kept constant across different environments, using the urban macrocell path-loss model from 3GPP [3GPP, 2020] and the shadow fading model implemented via the sum-of-sinusoids method, as used in QuaDRiGa [Jaeckel *et al.*, 2018]. The DRL method employed here is the state-of-the-art proximal policy optimization (PPO) algorithms [Schulman *et al.*, 2017], with the actor-network built on a long short-term memory (LSTM) network [Hochreiter and Schmidhuber, 1997] to incorporate memory capabilities. Performance metrics for user association encompass the 5th percentile user rate (5% rate) to evaluate cell-edge performance, and the logarithmic mean of all user rates, which serves as an indicator of the overall utility of the wireless network.

Results

We first thoroughly train DRL agents in various training environments. it is observed that all DRL agents notably surpass the traditional Max SINR approach, with detailed figures provided in the technical appendix. Subsequently, these agents are deployed in the real scenario, and the results are presented in Table 3. Consistent with our expectations, agents trained with random mobility models exhibit substantial performance degradation, in some cases deteriorating to levels comparable to or even worse than the traditional Max SINR method. In contrast, the DRL agent trained with the Map2Traj-based mobility model maintains superior performance.

Category	Method	5% **rate** ↑	**Utility** ↑
Random mobility	DRL (RWP)	2.098	6.571
	DRL (GM)	3.032	6.592
	DRL (M-RWP)	5.622	6.638
	DRL (M-GM)	2.894	6.595
Zero-shot	DRL (Map2Traj)	**7.688**	**6.669**
Real traj-based	DRL (DiffTraj)	6.976	6.664
	DRL (OD)	7.295	6.662
Non-DRL (Max SINR)		5.394	6.574
DRL (Real)		8.503	6.682

Table 3: User association performance in deployment

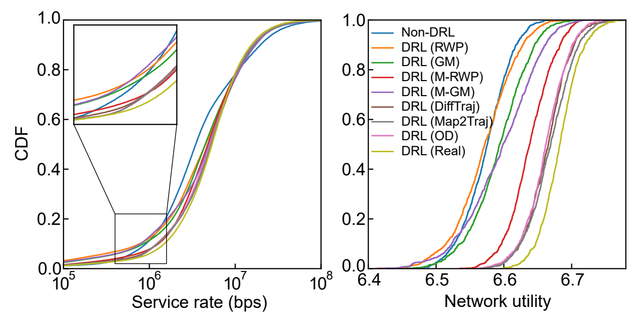


Figure 7: CDF of performance in deployment.

To provide a more nuanced view of the performance, we present the cumulative density function (CDF) of the metrics. As illustrated in Figure 7, the Map2Traj-based DRL agent not only outperforms those trained with random mobility models but also closely approaches the performance of the agent trained in the real environment. All these results demonstrate that synthetic trajectories generated by Map2Traj have efficacy comparable to real ones for learning-based wireless network optimization.

6 Conclusion

In this paper, we delve into the correlation between street maps and user mobility, introducing a novel Map2Traj method based on a diffusion model to achieve zero-shot trajectory generation for wireless network optimization. While prior research has successfully generated trajectories for specific regions using real datasets, we are the first to directly generate synthetic trajectories for new and unobserved areas. Extensive experiments demonstrate our method outperforms the traditional random mobility model and even some real trajectory-based models in terms of trajectory pattern and spatial distribution similarities. This enables the creation of high-fidelity individual user channel states and an accurate representation of the overall network user distribution. Through a case study of user association and load balancing in wireless networks, we validate that trajectories generated by Map2Traj exhibit comparable efficacy to real ones for wireless network optimization.

Acknowledgments

This work is supported in part by the Postgraduate Research&Practice Innovation Program of Jiangsu Province under Grant KYCX25_0468.

References

[3GPP, 2020] 3GPP. Study on channel model for frequencies from 0.5 to 100 ghz. https://www.3gpp.org/ftp/Specs/archive/38_series/38.901/38901-g10.zip, 2020. Accessed: 2020-01-11.

[Berndt and Clifford, 1994] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, page 359–370, 1994.

[Cao and Li, 2021] Chu Cao and Mo Li. Generating mobility trajectories with retained data utility. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2610–2620, New York, NY, 2021.

[Chen *et al.*, 2005] Lei Chen, M. Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, page 491–502, New York, NY, 2005.

[Didi-Chuxing, 2017] Didi-Chuxing. GAIA Initiative. https://outreach.didichuxing.com, 2017.

[Endo *et al.*, 2016] Yuki Endo, Hiroyuki Toda, Kyosuke Nishida, and Jotaro Ikedo. Classifying spatial trajectories using representation learning. *International Journal of Data Science and Analytics*, 2(3):107–117, 2016.

[Feriani and Hossain, 2021] Amal Feriani and Ekram Hossain. Single and multi-agent deep reinforcement learning for ai-enabled wireless networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 23(2):1226–1252, 2021.

[Goodfellow *et al.*, 2020] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[Gupta *et al.*, 2021] Manan Gupta, Ryan M. Dreifuerst, Ali Yazdan, Po-Han Huang, Sanjay Kasturia, and Jeffrey G. Andrews. Load balancing and handover optimization in multi-band networks using deep reinforcement learning. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Madrid, Spain, 2021.

[Gupta *et al.*, 2024] Manan Gupta, Sandeep Chinchali, Paul Parayil Varkey, and Jeffrey G. Andrews. Forecaster-aided user association and load balancing in multi-band mobile networks. *IEEE Transactions on Wireless Communications*, 23(5):5157–5171, 2024.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 33, pages 6840–6851, virtual, 2020. Curran Associates, Inc.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Huang *et al.*, 2022] Ziwei Huang, Xiang Cheng, and Xuefeng Yin. A general 3d non-stationary 6g channel model with time-space consistency. *IEEE Transactions on Communications*, 70(5):3436–3450, May 2022.

[Jaeckel *et al.*, 2018] Stephana Jaeckel, Leszek Raschkowski, Frank Burkhardt, and Lars Thiele. Efficient sum-of-sinusoids-based spatial consistency for the 3gpp new-radio channel model. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, Abu Dhabi, United Arab Emirates, 2018.

[Jia and Wang, 2023] Pengyi Jia and Xianbin Wang. A new virtual network topology-based digital twin for spatial-temporal load-balanced user association in 6G hetnets. *IEEE Journal on Selected Areas in Communications*, 41(10):3080–3094, Aug. 2023.

[Jiang and Fei, 2017] Bingnan Jiang and Yunsi Fei. Vehicle speed prediction by two-level data driven models in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1793–1801, Nov. 2017.

[Jiang *et al.*, 2023] Wenjun Jiang, Wayne Xin Zhao, Jingyuan Wang, and Jiawei Jiang. Continuous trajectory generation based on two-stage gan. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4374–4382, Washington, DC, Jun. 2023.

[Johnson and Maltz, 1996] David B. Johnson and David A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, pages 153–181. Springer US, Boston, MA, 1996.

[Kolouri *et al.*, 2019] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced wasserstein distances. In *Advances in Neural Information Processing Systems*, volume 32, Vancouver, BC, 2019. Curran Associates, Inc.

[Liang and Haas, 1999] B. Liang and Z.J. Haas. Predictive distance-based mobility management for pcs networks. In *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings.*, volume 3, pages 1377–1384 vol.3, New York, NY, 1999.

[Liu *et al.*, 2018] Xi Liu, Hanzhou Chen, and Clio Andris. Trajgans: Using generative adversarial networks for geo-privacy protection of trajectory data (vision paper). In *Location privacy and security workshop*, pages 1–7, 2018.

[Naderializadeh *et al.*, 2021] Navid Naderializadeh, Jaroslaw J. Sydir, Meryem Simsek, and Hosein Nikopour. Resource management in wireless networks via multi-agent deep reinforcement learning. *IEEE Transactions on Wireless Communications*, 20(6):3507–3523, 2021.

[Newson and Krumm, 2009] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL*

International Conference on Advances in Geographic Information Systems, page 336–343, New York, NY, 2009. Association for Computing Machinery.

[Oktay *et al.*, 2018] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas, 2018.

[Ramesh *et al.*, 2021] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 8821–8831. PMLR, 18–24 Jul 2021.

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

[Rüschendorf, 1985] Ludger Rüschendorf. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1):117–129, 1985.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[Tabassum *et al.*, 2019] Hina Tabassum, Mohammad Salehi, and Ekram Hossain. Fundamentals of mobility-aware performance characterization of cellular networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 21(3):2288–2308, 2019.

[Tao *et al.*, 2021] Yaguang Tao, Alan Both, Rodrigo I. Silveira, Kevin Buchin, Stef Sijben, Ross S. Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. A comparative analysis of trajectory similarity measures. *GIScience & Remote Sensing*, 58(5):643–669, 2021.

[Tao *et al.*, 2024] Zhenyu Tao, Wei Xu, Yongming Huang, Xiaoyun Wang, and Xiaohu You. Wireless network digital twin for 6G: Generative AI as a key enabler. *IEEE Wireless Communication*, 31(4):24–31, Aug. 2024.

[Wu and He, 2018] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.

[Xian *et al.*, 2017] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, July 2017.

[Xu *et al.*, 2023] Wei Xu, Zhaohui Yang, Derrick Wing Kwan Ng, Marco Levorato, Yonina C. Eldar, and Mérouane Debbah. Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):9–39, 2023.

[Zhang *et al.*, 2020] Xin Zhang, Yanhua Li, Xun Zhou, Ziming Zhang, and Jun Luo. Trajgail: Trajectory generative adversarial imitation learning for long-term decision analysis. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 801–810, Sorrento, Italy, 2020.

[Zhao *et al.*, 2019] Nan Zhao, Ying-Chang Liang, Dusit Niyato, Yiyang Pei, Minghu Wu, and Yunhao Jiang. Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks. *IEEE Transactions on Wireless Communications*, 18(11):5141–5152, 2019.

[Zhu *et al.*, 2023a] Yuanshao Zhu, Yongchao Ye, Ying Wu, Xiangyu Zhao, and James Yu. Synmob: Creating high-fidelity synthetic gps trajectory dataset for urban mobility analysis. In *Advances in Neural Information Processing Systems*, volume 36, pages 22961–22977, New Orleans, LA, Dec. 2023.

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