High-Fidelity Road Network Generation with Latent Diffusion Models

Jinming Wang[1] , **Hongkai Wen**[2] , **Geyong Min**[1] , **Man Luo**[1*]

[1]Department of Computer Science, University of Exeter
[2]Department of Computer Science, University of Warwick
{jw1294, g.min, m.luo}@exeter.ac.uk, hongkai.wen@warwick.ac.uk

Abstract

Road networks are the vein of modern cities. Yet, maintaining up-to-date and accurate road network information is a persistent challenge, especially in areas with rapid urban changes or limited surveying resources. Crowdsourced trajectories, e.g., from GPS records collected by mobile devices and vehicles, have emerged as a powerful data source for continuously mapping the urban areas. However, the inherent noise, irregular and often sparse sampling rates, and the vast variability in movement patterns make the problem of road network generation from trajectories a non-trivial task. Existing methods often approach this from an appearance-based perspective: they typically render trajectories as 2D density maps and then employ heuristic algorithms to extract road networks - leading to inevitable information loss and thus poor performance especially when trajectories are sparse or ambiguities present, e.g. flyovers. In this paper, we propose a novel approach, called GraphWalker, to generate high-fidelity road network graphs from raw trajectories in an end-to-end manner. We achieve this by designing a bespoke latent diffusion transformer T2W-DiT, which treats input trajectories as generation conditions, and gradually denoises samples from a latent space to obtain the corresponding *walks* on the underlying road network graph - then assemble them together as the final road network. Extensive experiments on multiple datasets demonstrate the proposed GraphWalker can effectively generate high quality road networks from noisy and sparse trajectories, showcasing significant improvements over state-of-the-art.

1 Introduction

Road networks are fundamental to the functioning of our cities, supporting the efficient flow of goods and people, facilitating trade and economic growth, and enabling access to essential daily services. With cities expanding and road configurations evolving at a rapid pace, obtaining accurate digital representation of road networks has become increasingly vital yet difficult. In practice, roads in the real world may exhibit complex topologies, whose pattern can vary a lot across different regions. Traditional approaches for road network generation often rely on survey and/or remote sensing, using satellite or aerial images to manually or semi-automatically identify road segments [Huang *et al.*, 2020]. Although being able to offer the bird's-eye view of the landscapes, these approaches are often time-consuming and cost-intensive - high-resolution imagery can be expensive to acquire, while updates may not be frequent enough to reflect the rapid road infrastructure changes. The recent widespread use of location-aware devices, such as mobile phones and vehicles with GPS, has generated an abundance of crowdsourced trajectories when their users move across the urban space, offering a unique view of the structures of the real-world road network. Such data can provide near-continuous insights on how people or vehicles may move on the roads, and thus extracting road networks from that becomes a more cost-effective way than the traditional approaches. To generate road networks, existing methods [Hong *et al.*, 2024] typically project the trajectories on a 2D plane, where the location points of the trajectories are overlayed as density maps. These density maps are then converted to a heatmap with standard computer vision techniques, where each pixel encodes the likelihood of a valid road presents, i.e. an occupancy grid. They then apply heuristic-based algorithms such as non-maiximum suppression to extract road networks from the computed heatmaps, often in the format of a graph where vertices are road junctions while edges are segments.

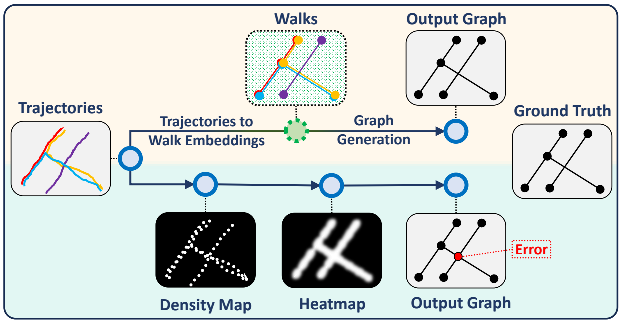

Figure 1: The proposed GraphWalker (top) vs. appearance-based methods (bottom) when generating road networks from trajectories.

[*] Corresponding author.

These approaches, although more efficient than survey or remote sensing, are still lacking in many aspects. Firstly, discretizing individual location points into heatmaps inevitably causes information loss due to limited resolution, while the directional information encoded naturally in trajectories has been ignored, which are valuable to infer road segment properties, e.g., one-way roads. Secondly, these appearance-based approaches tend to be sensitive to the quality of trajectories - they may work with dense and accurate trajectory - but in reality location data such as GPS records are likely to suffer from positional errors and irregular sampling rates, resulting in disjoint or misaligned roads generated. Finally, only looking at 2D heatmaps these approach often fail to discover non-trivial road topologies. With heuristic-based algorithms like corner/edge detection, it is impossible to tell if an intersection of two lines is a normal road junction or rather an overpass/underpass, as illustrated in Fig. 1.

In this paper, we aim to overcome the shortcomings of existing solutions, and design an end-to-end approach for high-fidelity road network generation. Instead of directly trying to knead trajectories into a road network graph, let us first take a step back and consider the inverse problem of generating noisy trajectories from a known road network graph. For a given road network graph, one can easily generate a trajectory by: i) select a finite sequence of edges which joins a sequence of vertices (referred to as a *walk* in graph theory); and ii) sampling noisy location points according to the edges of the selected walk. This indicates if there is a way to reverse this process, for any trajectory we can discover the corresponding walk describing how it has traversed the underlying road network. Then with sufficiently many trajectories one can generate a large amount of walks, each of which corresponds to a subgraph of the actual road network, and we should be able to reconstruct the final road network graph from them.

Building upon this insight, we propose a novel road network generation framework, called **GraphWalker** [1], based on latent diffusion model (LDM) [Rombach *et al.*, 2022]. We formulate the problem of road network generation from trajectories under the umbrella of denoising diffusion-based generative modeling. Essentially, we view trajectories as the noisy and "interpolated" version of corresponding walks on the graph, where GraphWalker is designed to generate those walks and assemble them into a valid road network graph. Concretely, we first train a Walk-to-Graph Variational Autoencoder (W2G-VAE) that can represent arbitrary walks on the road network graph in a latent space, and later reconstruct the graph from walk embeddings. Then we design a novel Trajectory-to-Walk Diffusion Transformer (T2W-DiT) that can generate valid latent walk embeddings with input trajectories as *conditions*, which are then decoded into actual walks by the previously trained W2G-VAE and assembled into the final road network graph via an end-to-end trainable neural network. In this way, we cast the problem of generating road networks from trajectories into a generative denoising process, and circumvent the challenges of directly reconstructing the whole graph by considering walks in a latent space as the intermediate representations. The technical contributions of this paper can be summarized as follows:

- To the best of our knowledge, we are the first to formulate the task of road network generation from trajectories with end-to-end trainable denoising diffusion-based generative modeling. Unlike existing solutions that rely heavily on hand-crafted heuristic algorithms, our approach generates high-fidelity road networks directly from noisy trajectories via recovering the corresponding walks.
- We design a new generative framework GraphWalker based on LDM, which features a cross-domain VAE named W2G-VAE and a novel T2W-DiT, both bespoke for this particular task. The encoder of W2G-VAE maps walks into a latent space, from which the latter T2W-DiT gradually generates walk embeddings given the input trajectories as conditions, while the W2G-VAE decoder recovers valid walks from the generated embeddings and reconstructs the final road networks.
- Extensive experiments conducted on diverse real-world datasets show that GraphWalker outperforms both appearance-based methods and state-of-the-art diffusion-based graph generation approaches, demonstrating significant improvements in generation accuracy and superior robustness with limited trajectory data and unusual road topologies present.

2 Preliminaries

2.1 Problem Formulation

Trajectory. Let $p = (longitude, latitude)$ be a GPS point. We define a trajectory τ as an ordered set of location points $\tau = \{p_1, p_2, ..., p_N\}$, which are often sampled from the continuous motion when an agent is moving across the space e.g., on a road network.

Road Network Graph. Following existing literature, we consider road network as a graph, where nodes are junctions and edges are road segments between two junctions. Instead of using the adjacency matrix representation, which often fails to capture many key properties of real-world road networks such as curved road segments, self-loops and multiple segments between two junctions, in this paper we adopt an *edge-list* representation. Concretely, we define road network graph as $G = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_{|G|}\}$, where each edge $\boldsymbol{e} = \{p_1, p_2, ..., p_K\}$ is a polyline containing K points. In our implementation for simplicity we assume a fixed K for all edges. The vertices of the graph are then the collection of start and end points of the polylines, i.e. p_1 and p_K of each $\boldsymbol{e}$. Note that G encodes both topological and absolute location information: each edge contains the locations of two junctions (p_1 and p_K).

Trajectory Walk. For a given trajectory τ, when mapped to the known road network graph G, the finite sequence of edges that τ traverses is defined as its *walk* $w = \{\boldsymbol{e}_i | i \in [1, |w|]\}$ on G, where $|w|$ is the number of edges τ visited. In this paper we assume for any trajectory τ there is always a corresponding walk w exists, i.e. many to one mapping.

End-to-End Road Network Generation. Given the input set of trajectories $\boldsymbol{\tau} = \{\tau_1, \tau_2, ..., \tau_M\}$, the problem of end-to-end road network generation aims to find a function f that

[1] Code at: https://github.com/JinmingWang/GraphWalker.

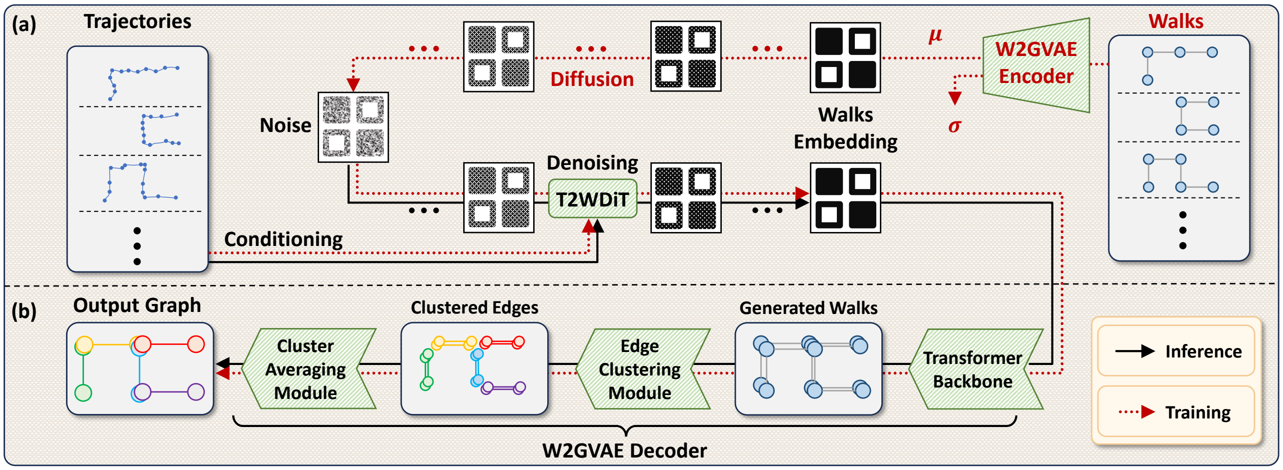

Figure 2: Overview of GraphWalker framework. (a) Trajecotries to walks embeddings. (b) Graph generation from walk embeddings.

directly generates the underlying road network graph G on which the trajectories have traversed, i.e. $f(\boldsymbol{\tau}) \to G$.

2.2 Diffusion Models

A discrete-time diffusion model [Ho *et al.*, 2020] defines a two-process paradigm. Given a clean data sample $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$, the forward process gradually transforms it into Gaussian noise over T steps according to a predefined schedule $\beta_1, \ldots, \beta_T$. At each step t, the noised sample is drawn from $q_(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$. This process can also be expressed directly in terms of $\mathbf{x}_0$: $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$.

The reverse process learns to denoise: a neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is trained to estimate the noise added at each step. This is done by minimizing the loss $\mathbb{E}_{t,\mathbf{x}_0,\epsilon}|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)|^2$. Once trained, the model can generate samples by iteratively applying the denoising step starting from Gaussian noise. Conditioning information (e.g., labels or other signals $\mathbf{c}$) can be included during both training and sampling by extending the network to $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})$.

To reduce computational cost while retaining high output quality, latent diffusion models (LDMs) [Rombach *et al.*, 2022] perform the diffusion process in a compressed latent space. An autoencoder is used to map input data into low-dimensional latent representations, where the denoising network operates. During generation, latent samples are first produced by the diffusion model, then decoded back into the original data space by the decoder.

3 Methodology

3.1 GraphWalker Overview

A naive solution to generate a road network graph from a set of trajectories in an end-to-end manner is to simply feed the trajectories to generative models such as GANs or diffusion models [Goodfellow *et al.*, 2014; Ho *et al.*, 2020]. However, this approach often fails to produce high-fidelity road network graphs, since the input trajectories are typically noisy - the locations of trajectory points may be imprecise (e.g., due to GPS drift), while their distribution can be sparse and/or skewed - thus simply overlying them together as the input of generative models will propagate or even enlarge such noises, leading to inferior results. To address this, we consider a different bottom-up approach. We observe that, although noisy, for a given road network graph G, a trajectory τ_i can be viewed as a noisy version of a partial graph traversal, i.e. it is possible to recover a *walk* w_i on the graph that corresponds exactly to τ_i. Once we are able to discover all the graph walks corresponding to input trajectories, it is also possible to assemble those into a consistent G, e.g. by merging duplicate edges/nodes.

GraphWalker follows this intuition and addresses the road network generation task in two stages: i) *trajectories to walk embeddings*; and ii) *graph generation from walk embeddings*, as illustrated in Fig. 2. In the first stage, we train a Walk-to-Graph Variational Autoencoder (W2G-VAE) that can encode arbitrary walks on G into a latent space, and later decode G from this space. Then for each input trajectory, we feed it as the condition to a LDM named Trajectories-to-Walks Diffusion Transformer (T2W-DiT) that operates in such space, and generates embeddings of the corresponding walks. In the latter stage, we use the trained decoder of our W2G-VAE to recover walks from their latent representations, and assemble them into the complete graph G. We now elaborate on each stage in more detail.

3.2 Trajectories to Walk Embeddings

Let $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots\tau_M\}$ be a set of M trajectories, and $\boldsymbol{w} = \{w_1, w_2, \ldots w_M\}$ be the corresponding walks on the known road network graph G. As explained above, the first stage of GraphWalker is to construct a latent space $\mathcal{Z}$ of walks in which: i) any walk can be mapped to its latent representations; and ii) sampling a latent vector from this continuous space (e.g. with appropriate denoising process) can lead to a valid walk embedding - meaning that the space should be smooth under interpolation and/or extrapolation. Note that the input walks, by definition, are chain-like structures. Therefore, we design our W2G-VAE encoder with simple yet effective 1D ResNet [He *et al.*, 2015], which essentially applies variational regularization given $\boldsymbol{w}$. Note that for the decoding part of our W2G-VAE, instead of recovering only the

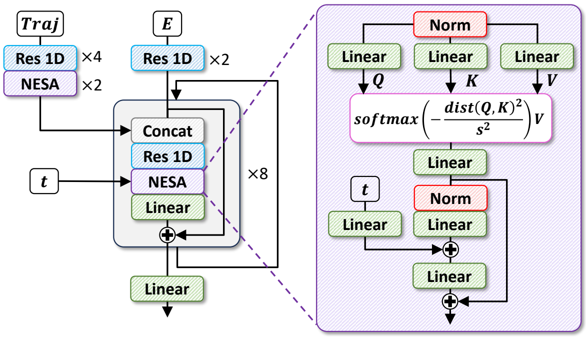

Figure 3: Architecture details of T2W-DiT, with NESA module highlighted on the right.

walks as in standard VAE, we consider an augmented decoder to directly reconstruct G, where modules assembling walks to graph are also end-to-end learnable. We will discuss the decoder details in the following Sec. 3.3.

Now we have a latent space $\mathcal{Z}$ induced by our W2G-VAE, which in theory, should be able to encode all possible walks on the road network graph G. The next step is to generate valid walk embeddings, i.e., a sequence of latent vector in $\mathcal{Z}$, given input trajectories $\boldsymbol{\tau}$. We design T2W-DiT, which is able to generate latent walk embeddings from $\mathcal{Z}$ conditioned on given trajectories. Essentially, our T2W-DiT starts from a Gaussian noise in $\mathcal{Z}$, gradually performs denoising with input trajectories $\boldsymbol{\tau}$ as the condition, and outputs latent embeddings $\boldsymbol{z} \in \mathcal{Z}$ corresponding to $\boldsymbol{\tau}$. When incorporating $\boldsymbol{\tau}$ as conditions, our T2W-DiT first concatenates features from $\boldsymbol{\tau}$ and the sampled latent embedding, and then applies a 1D ResNet block to process the concatenated feature sequences. We also design modules in our model to be permutation invariant to ensure that the generated embeddings, and also the final road network graph will not be affected by the order of input trajectories.

To capture the non-trivial spatio-temporal correlations in trajectories and walks, instead of using Multi-head Self-Attention (MHSA) with layer normalization, which is a standard practice in DiTs, we bespoke our T2W-DiT with Negative Euclidean Self-attention (NESA), and consider feature but not layer normalization. Concretely, we first replace the standard layer normalization with feature normalization as follows:

$$y_i = \frac{x_{:,i} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} \cdot \gamma + \beta \tag{1}$$

where $x \in \mathbb{R}^{M \times D}$ represents the input features, assuming we have M trajectories as input conditions and D is the feature dimensions. $\mu_i = \frac{1}{L}\sum_{j=1}^{L} x_{j,i}$ and $\sigma_i^2 = \frac{1}{L}\sum_{j=1}^{L}(x_{j,i} - \mu_i)^2$ are the mean and variance of the i-th feature, and ϵ is a small constant (set to 10^{-5}) to avoid division by zero. Then the NESA is defined as:

$$softmax\left(-\frac{dist(q,k)^2}{s^2}\right)v \tag{2}$$

where q, k, v are the projections of the input feature x, analogous to query, key and value in the standard MHSA. The operation $dist(\cdot,\cdot)$ computes the Euclidean distance between each pair of tokens, and s is a learnable parameter controlling the decay rate with respect to the distance. Note that here NESA does not work well with the standard layer normalization, which will result in tokens with similar distances in Euclidean spaces - leading to very close scores across tokens. Detailed architecture of the proposed T2W-DiT is shown in Fig. 3, with NESA module highlighted on the right.

3.3 Graph Generation From Walk Embeddings

Now given the M input trajectories $\boldsymbol{\tau} = \{\tau_1, \tau_2, ..., \tau_M\}$, our T2W-DiT has recovered the corresponding latent walk embeddings $\boldsymbol{z} = \{z_1, z_2, ..., z_M\}$. The next stage is to generate the final road network graph G from $\boldsymbol{z}$. As mentioned in Sec. 3.2, instead of employing a standard VAE decoder that only reconstructs walks and manually assembles the road network G later, our W2G-VAE considers an augmented decoder that is end-to-end trainable, and can directly output G. Concretely, our decoder uses a transformer backbone to firstly generate the corresponding walks $\boldsymbol{w} = \{w_1, w_2, ..., w_M\}$, which are then fed into an edge clustering module. Within that, the edges of all walks in $\boldsymbol{w}$ are firstly disassembled into individual edge candidates, which are then clustered based on their similarities. Let $\boldsymbol{E}$ denote the set of edge candidates. The edge clustering module constructs a matrix $P \in \mathbb{R}^{|\boldsymbol{E}| \times |\boldsymbol{E}| \times 4K}$, where each element of P is the concatenation of two edge candidates (K location points in each edge, with (x,y) coordinates). It then processes P with an MLP and a sigmoid activation function, outputting a symmetric affinity matrix $A \in \mathbb{R}^{|\boldsymbol{E}| \times |\boldsymbol{E}|}$. Each element $A_{ij} \in (0,1)$ represents the likelihood that the i-th and j-th candidates should belong to the same cluster. With A one can infer the cluster structures among $\boldsymbol{E}$, e.g., if we consider the affinity between edges as a binary relationship (exists when the affinity score is above a pre-defined threshold), one can discover the cluster that $\boldsymbol{e}_i$ belongs to in by looking at its transitive closure.

Let $\boldsymbol{C} \subseteq \boldsymbol{E}$ be a cluster of edge candidates, and $A^{\boldsymbol{C}}$ is a sub-matrix of A, containing only the affinity scores between those edge candidates in $\boldsymbol{C}$. Let $A^{\boldsymbol{C}}_{uv}$ be the largest value in $A^{\boldsymbol{C}}$, meaning that $\boldsymbol{e}_u$ and $\boldsymbol{e}_v$ are the most similar candidates. We then perform a row-wise softmax on the u-th row - $A^{\boldsymbol{C}_{u,:}}$ becomes a probability distribution, which can be seen as the weights of each candidate contributing to the cluster center. Now we can compute the cluster centre by averaging the location points between all edge candidates $\boldsymbol{e}_u \in \boldsymbol{C}$, weighted by the normalized $A^{\boldsymbol{C}_{u,:}}$. We iteratively perform the above for all clusters in $\boldsymbol{E}$, and obtain a new set of edges, one of each cluster, which is exactly the desired road network G.

3.4 Training

Our approach adopts the standard simulation-free LDM training paradigm. Concretely, we first pre-train W2G-VAE with $\boldsymbol{w}$ sampled from a known road network graph G. During training, we consider the following three loss functions: i) KL-divergence as in typical VAE training: $-\frac{1}{2}\sum_{i=1}^{n}\left(1 + \log\sigma_i^2 - \mu_i^2 - \sigma_i^2\right)$, where μ and σ are the mean and logarithmic variance of the latent embedding $\boldsymbol{z}$; ii) MSE loss to ensure correct reconstruction of walks: $\frac{1}{M}\sum_{i=1}^{M}(\boldsymbol{w_i} - \boldsymbol{\hat{w}_i})^2$, where $\boldsymbol{w}_i$ and $\boldsymbol{\hat{w}_i}$ denote

	Trained on:	TYO	TYO	TYO	SHA	SHA	SHA	LAS	LAS	LAS
Metric	**Tested on:**	TYO	SHA	LAS	TYO	SHA	LAS	TYO	SHA	LAS
Hungarian MAE	TR2RM	0.381	0.474	0.480	0.389	0.427	0.470	0.375	0.467	0.471
	DF-DRUNet	0.400	0.455	0.479	0.397	0.409	0.471	0.389	0.412	0.449
	SmallMap	**0.364**	0.508	0.525	0.364	0.537	0.550	**0.351**	0.554	0.553
	Graphusion	0.396	0.552	0.504	0.410	0.415	0.434	0.412	0.426	0.412
	GraphWalker	0.378	**0.411**	**0.415**	**0.349**	**0.241**	**0.286**	0.365	**0.251**	**0.299**
Hungarian MSE	TR2RM	0.334	0.507	0.489	0.344	0.452	0.485	0.316	0.502	0.486
	DF-DRUNet	0.369	0.481	0.501	0.362	0.428	0.500	0.346	0.432	0.479
	SmallMap	**0.305**	0.542	0.530	**0.301**	0.577	0.564	**0.282**	0.595	0.563
	Graphusion	0.348	0.613	0.552	0.371	0.485	0.471	0.374	0.482	0.450
	GraphWalker	0.377	**0.456**	**0.449**	0.329	**0.272**	**0.312**	0.350	**0.284**	**0.330**
Chamfer MAE	TR2RM	0.453	0.777	0.751	0.460	0.785	0.759	0.468	0.798	0.745
	DF-DRUNet	0.448	0.788	0.732	0.440	0.793	0.740	0.452	0.775	0.721
	SmallMap	0.440	0.800	0.707	0.440	0.802	0.702	0.444	0.804	0.703
	Graphusion	0.488	0.897	0.787	0.520	0.826	0.728	0.515	0.808	0.672
	GraphWalker	**0.291**	**0.521**	**0.454**	**0.332**	**0.456**	**0.420**	**0.315**	**0.445**	**0.418**
Chamfer MSE	TR2RM	0.232	0.659	0.570	0.228	0.689	0.590	0.236	0.705	0.579
	DF-DRUNet	0.227	0.672	0.563	0.217	0.707	0.574	0.232	0.658	0.543
	SmallMap	0.219	0.669	0.521	0.223	0.675	0.530	0.216	0.671	0.523
	Graphusion	0.270	0.818	0.646	0.291	0.856	0.614	0.295	0.829	0.576
	GraphWalker	**0.177**	**0.456**	**0.368**	**0.183**	**0.437**	**0.357**	**0.176**	**0.427**	**0.362**
Wasserstein Distance of Edge Length Distributions	TR2RM	0.250	0.848	0.572	0.293	0.998	0.746	0.272	0.942	0.706
	DF-DRUNet	0.286	0.846	0.602	0.294	1.076	0.740	0.268	0.969	0.730
	SmallMap	0.227	0.820	0.509	0.238	0.777	0.532	0.221	0.740	0.519
	Graphusion	0.206	0.733	0.567	0.293	0.747	0.499	0.330	0.776	0.491
	GraphWalker	**0.173**	**0.624**	**0.453**	**0.185**	**0.626**	**0.462**	**0.182**	**0.622**	**0.452**

Table 1: Performance comparison when competing approaches are trained and tested with data collected from different cities: Tokyo (TYO), Shanghai (SHA) and Las Vegas (LAS). Best in red while second best in blue.

the ground truth and predicted walks; and iii) a binary cross-entropy loss to measure graph reconstruction quality: $-\frac{1}{M}\sum_{i,j}\left(A_{ij}\log\hat{A}_{ij}+(1-A_{ij})\log(1-\hat{A}_{ij})\right)$, where A and $\hat{A}$ are the affinity matrices, calculated from the ground truth $\boldsymbol{w}$ and the reconstructed $\hat{\boldsymbol{w}}$ respectively. The final loss used for W2G-VAE training is a linear combination of the above three functions.

Then we fix the parameters of W2G-VAE and follow the standard training procedures for diffusion models to train our T2W-DiT. In particular, we encode $\boldsymbol{w}$ into latent encodings $\boldsymbol{z}$ with the pre-trained W2G-VAE, and add noise through the forward diffusion process at a randomly selected timestep t to obtain $\boldsymbol{z}_t$ - then T2W-DiT is trained to predict the added noise ϵ using the MSE loss: $\frac{1}{M}\sum_{i=1}^{M}(\epsilon_i-\hat{\epsilon}_i)^2$, where ϵ_i and $\hat{\epsilon}_i$ are the ground truth and predicted noise for the i-th data point.

4 Evaluation

4.1 Experimental Settings

Datasets. We evaluate the proposed GraphWalker on real-world datasets collected from three different cities: Tokyo (TYO), Shanghai (SHA) and Las Vegas (LAS). To ensure fair comparison with the appearance-based road network generation approaches, which requires both trajectory data and aerial or satellite images, we follow the existing literature [Hong *et al.*, 2024] and synthesize trajectories with respect to the ground truth road network. This also allows us to have trajectories with diverse motion patterns covering different areas across the cities, which is often limited in most of the publicly available trajectory datasets [Didi, 2017; Crailtap, 2018; Zheng *et al.*, 2011]. Concretely, for an area within the city specified by a bounding box of certain size, we collect its road network data using OpenStreetMap [OpenStreetMap contributors, 2017], and aerial images from Google Maps [Google, nd]. To generate trajectories, we employ the recent work ControlTraj, a diffusion-based trajectory generation technique [Zhu *et al.*, 2024], which can generate human-directed, high-fidelity trajectories constrained by the road network topology. For numerical stability, the generated location coordinates (in the format of GPS coordinates) are z-score normalized. We apply standard data augmentation techniques such as random rotations and flips, while also randomly shuffling the order of data to facilitate permutation-invariant learning.

Baselines. We compare the proposed approach against four strong baselines. i) **TR2RM** [Yang *et al.*, 2024b], which uses AD-LinkNet to process aerial images and trajectory density maps and capture multi-scale geographical features. ii) **DF-DRUNet** [Li *et al.*, 2024], which employs two separate UNet [Ronneberger *et al.*, 2015] to process aerial image and trajectory density maps, and then combine them with Gated Fusion Modules. iii) **SmallMap** [Hong *et al.*, 2024], which uses GANs to generate trajectories heatmaps and resolve irregularities in generated roads. Note that all the above baselines are appearance-based, i.e., eventually they need to extract road network graphs with heuristic algorithms like thinning, corner detection and flood-fill. We also consider a state-

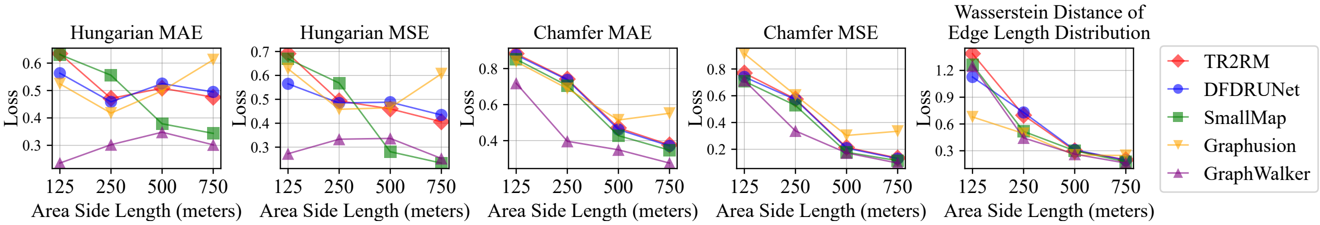

Figure 4: Performance of competing approaches when tested with data from areas of different sizes.

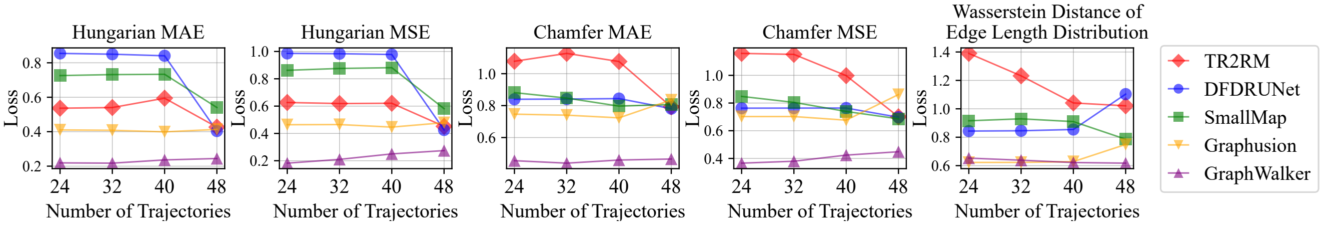

Figure 5: Performance of competing approaches when tested with different amounts of input trajectories.

of-the-art diffusion-based approach: iv) **Graphusion** [Yang *et al.*, 2024a], which is a generic graph generation approach based on LDM. We adapt Graphusion to work in our settings, and feed trajectories as the conditions into its generation process. Note that unlike the proposed approach, Graphusion does not consider walks, and its VAE directly encodes/decodes the entire graph.

Metrics. We consider the following metrics, Firstly, we care about the correct ordering and alignment between the generated edges and the ground truth. Therefore we first perform Hungarian algorithm to discover the optimal one-to-one mapping between the generated and ground truth edges, and then evaluate Mean Absolute Error (MAE) and Mean Squared Error (MSE) for each matched pair. This gives us two metrics: i) **Hungarian MAE**; and ii) **Hungarian MSE**. On the other hand, we are also interested in the point-wise matching error in both directions, and thus we also consider Chamfer Loss, resulting in two additional metrics: iii) **Chamfer MAE** and iv) **Chamfer MSE**. Finally, we consider the v) **Wasserstein Distance** between the distributions of edge lengths in the generated and ground truth graphs, which describes the global structural consistency of road network produced by different algorithms.

4.2 Results

Overall Performance. We evaluate the competing approaches with data across three different cities, each of which is trained on one city and tested on all three, to study their performance when generalized to unseen settings with heterogeneous road network topologies. The results are shown in Table 1. As we can see, appearance-based approaches generally underperform the proposed GraphWalker in most settings, despite TR2RM and DF-DRUNet which also use aerial images during generation. Surprisingly, the more recent diffusion-based Graphusion fails to demonstrate significant improvements over the existing appearance-based methods, indicating that although powerful, without careful customization LDM are not directly applicable to address the problem of road network generation from trajectories. On the other hand, the proposed GraphWalker outperforms all baselines in most cases, showcasing that our W2G-VAE and T2W-DiT, when work in tandem, can effectively generate high-fidelity road networks from noisy trajectories. In particular, GraphWalker consistently performs well when tested on different cities other than training - demonstrating up to 33% improvement over the baselines.

Performance vs. Area Sizes. In this experiment, we investigate how competing approaches perform when generating road networks for an area (bounding box) of different sizes. Specifically, we use data from a bounding box of approximately 250m×250m for training, and evaluate the quality of the generated road networks in areas of four different sizes, as shown in Fig. 4. We observe that most of the approaches except Graphusion enjoy an improved performance as the size increases, most likely due to the fact that more information is available in larger areas despite the noise in trajectories, i.e. the signal-to-noise ratio increases. Graphusion fails in such cases, potentially due to their diffusion models may not be robust enough when road network topologies change significantly as the area expands. Note that the proposed GraphWalker consistently outperforms all baselines in most cases, showing much lower errors especially in settings that they have not been trained on.

Performance vs. Limited Data. This experiment studies how competing approaches perform when input trajectories are limited. This is common in practice as real-world trajectories are often unevenly distributed across the cities: some areas may only have a few sparse trajectories available. In such cases, as shown in Fig. 6 (3rd row), the appearance-based approaches struggle to recover meaningful road networks. To evaluate quantitatively, we train all approaches with 48 trajectories, and during generation provide them with different amount of input trajectories. As shown in 5, the generation accuracy of appearance-based baselines degrades drastically with fewer input trajectories, while Graphusion and our GraphWalker maintain strong performance. This confirms that diffusion-based approaches are inherently robust to limited input data - the proposed GraphWalker is still the best

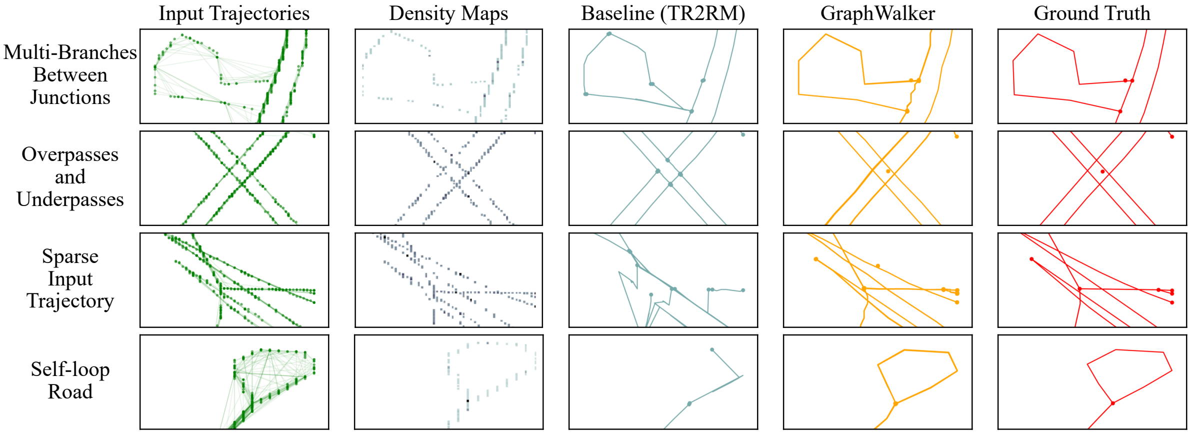

Figure 6: Visualization of road networks generated by appearance-based baselines and our GraphWalker when challenging cases present.

Method	Time	VAE FLOPs	DiT FLOPs
TR2RM	0.106s	-	-
DF-DRUNet	0.084s	-	-
SmallMap	0.085s	-	-
Graphusion	0.141s	50G	28G
GraphWalker	0.165s	32G	29G

Table 2: Inference time and model FLOPs of different approaches.

overall with much lower errors.

Robustness to Non-trivial Topology Patterns. In practice, road networks may exhibit complex topology patterns that are challenging to discover only from a bird's-eye view. In this experiment, we show that appearance-based approaches cannot recover road networks with non-trivial topology patterns. The results of different cases are visualized in Fig. 6. For instance, those baselines may generate erroneous road networks when there are multiple road segments between two junctions (row 1), there are overpasses/underpasses (row 2 and 3), or just roads with self-loops (row 4). On the contrary, we see that the proposed GraphWalker (column 4) is robust to such challenging cases and can generate graphs that are very close to the ground truth (column 5).

Computational Efficiency. Diffusion models can be more computationally intensive compared to appearance-based methods. In Table 2, we show the wall-clock inference time per input trajectory (averaged over 1000 randomly selected trajectories, measured on a single NVIDIA 4090 GPU). We also compute the FLOPS of both the VAE and DiT of our approach and also the diffusion-based Graphusion.

5 Related Work

Road Network Generation. Early road network generation methods rely on heuristic algorithms [Biagioni and Eriksson, 2012; Stanojevic *et al.*, 2018; Gao *et al.*, 2022; He *et al.*, 2018]. They are mainly based on trajectory data and human flow data. Benefit from the advancement in deep learning, many methods adopt CNNs to infer road networks from aerial imagery, such as SOC-RoadNet [Zhou *et al.*, 2022], CasNet [Cheng *et al.*, 2017] and SII-Net [Tao *et al.*, 2019]. RoadTracer [Bastani *et al.*, 2018] proposed an iterative graph construction algorithm, which is capable of accurately deriving the graph from the semantic segmentation output guided by a CNN decision function. Some other methods argue that adopting both remote sensing images and trajectory data makes the solution more robust, including DuARE [Yang *et al.*, 2022], TR2RM [Yang *et al.*, 2024b] and DF-DRUnet [Li *et al.*, 2024]. Moreover, some methods leverage power content generation tools such as generative adversarial network (GAN) [Goodfellow *et al.*, 2014; Yao *et al.*, 2024; Li *et al.*, 2022] and diffusion models [Gu *et al.*, 2024].

Graph Generation. Graph generation methods are popular in the chemistry and biology fields, where they are used to predict protein and molecular structures. They can be categorized into five classes based on the generative model used. The auto-regressive methods, including GraphRNN [You *et al.*, 2018] and MolecularRNN [Popova *et al.*, 2019]. The VAE-based methods, such as GraphVAE [Simonovsky and Komodakis, 2018], CGVAE [Liu *et al.*, 2019] and Graphite [Grover *et al.*, 2019]. The normalizing flow based methods, including MoFlow [Zang and Wang, 2020] and GraphDF [Luo *et al.*, 2021]. The GAN based methods, such as GraphGAN [Wang *et al.*, 2017] and Mol-CycleGAN [Maziarka *et al.*, 2020]. Finally, the diffusion model based methods including GraphGDP [Huang *et al.*, 2022], EDP-GNN [Niu *et al.*, 2020] and Graphusion [Yang *et al.*, 2024a].

6 Conclusion

This paper proposes GraphWalker, a novel end-to-end trainable framework for high-fidelity road network generation from noisy trajectories. We draw from the powerful latent diffusion models, and design a bespoke cross-domain VAE, W2G-VAE, as well as a novel diffusion transformer T2W-DiT, that can iteratively recover the road network graph via denoising walk embeddings in a latent space with the input trajectories as conditions. Extensive experiments with real-world trajectories collected from three different cities show that GraphWalker consistently outperforms both appearance-based and state-of-the-art diffusion-based baselines, and is more robust under different settings such as heterogenous area sizes, limited input data and challenging road topologies.

References

[Bastani *et al.*, 2018] Favyen Bastani, Songtao He, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, David J. DeWitt, and Sam Madden. Unthule: An incremental graph construction process for robust road map extraction from aerial images. *CoRR*, abs/1802.03680, 2018.

[Biagioni and Eriksson, 2012] James Biagioni and Jakob Eriksson. Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, page 79–88, New York, NY, USA, 2012. Association for Computing Machinery.

[Cheng *et al.*, 2017] Guangliang Cheng, Ying Wang, Shibiao Xu, Hongzhen Wang, Shiming Xiang, and Chunhong Pan. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3322–3337, 2017.

[Crailtap, 2018] Crailtap. Taxi trajectory dataset. https://www.kaggle.com/datasets/crailtap/taxi-trajectory/, 2018. Accessed: 2025-01-24.

[Didi, 2017] Didi. Didi chuxing outreach. https://outreach.didichuxing.com/, 2017. Accessed: 2025-01-24.

[Gao *et al.*, 2022] Lei Gao, Lu Wei, Jian Yang, and Jinhong Li. Automatic intersection extraction method for urban road networks based on trajectory intersection points. *Applied Sciences*, 12(12), 2022.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[Google, nd] Google. Google maps. https://maps.google.com, n.d. Accessed: 2024-12-12.

[Grover *et al.*, 2019] Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs, 2019.

[Gu *et al.*, 2024] Xiaoyan Gu, Mengmeng Zhang, Jinxin Lyu, and Quansheng Ge. Generating urban road networks with conditional diffusion models. *ISPRS International Journal of Geo-Information*, 13(6), 2024.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[He *et al.*, 2018] Songtao He, Favyen Bastani, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, and Sam Madden. Roadrunner: improving the precision of road network inference from gps trajectories. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '18, page 3–12, New York, NY, USA, 2018. Association for Computing Machinery.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[Hong *et al.*, 2024] Zhiqing Hong, Haotian Wang, Yi Ding, Guang Wang, Tian He, and Desheng Zhang. Smallmap: Low-cost community road map sensing with uncertain delivery behavior. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(2), May 2024.

[Huang *et al.*, 2020] Kang Huang, Jun Shi, Gaofeng Zhang, Benzhu Xu, and Liping Zheng. D-crosslinknet for automatic road extraction from aerial imagery. In Yuxin Peng, Qingshan Liu, Huchuan Lu, Zhenan Sun, Chenglin Liu, Xilin Chen, Hongbin Zha, and Jian Yang, editors, *Pattern Recognition and Computer Vision*, pages 315–327, Cham, 2020. Springer International Publishing.

[Huang *et al.*, 2022] Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. Graphgdp: Generative diffusion processes for permutation invariant graph generation, 2022.

[Li *et al.*, 2022] Jiawei Li, Linkun Lyu, Jia Shi, Jie Zhao, Junjie Xu, Jiuchong Gao, Renqing He, and Zhizhao Sun. Generating community road network from gps trajectories via style transfer. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, New York, NY, USA, 2022. Association for Computing Machinery.

[Li *et al.*, 2024] Bingnan Li, Jiuchong Gao, Shuiping Chen, Samsung Lim, and Hai Jiang. Df-drunet: A decoder fusion model for automatic road extraction leveraging remote sensing images and gps trajectory data. *International Journal of Applied Earth Observation and Geoinformation*, 127:103632, 2024.

[Liu *et al.*, 2019] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. Constrained graph variational autoencoders for molecule design, 2019.

[Luo *et al.*, 2021] Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation, 2021.

[Maziarka *et al.*, 2020] Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoł. Mol-cyclegan: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1), January 2020.

[Niu *et al.*, 2020] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling, 2020.

[OpenStreetMap contributors, 2017] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[Popova *et al.*, 2019] Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties, 2019.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[Simonovsky and Komodakis, 2018] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. *CoRR*, abs/1802.03480, 2018.

[Stanojevic *et al.*, 2018] Rade Stanojevic, Sofiane Abbar, Saravanan Thirumuruganathan, Sanjay Chawla, Fethi Filali, and Ahid Aleimat. *Robust Road Map Inference through Network Alignment of Trajectories*, pages 135–143. SIAM International Conference on Data Mining, 2018.

[Tao *et al.*, 2019] Chao Tao, Ji Qi, Yansheng Li, Hao Wang, and Haifeng Li. Spatial information inference net: Road extraction using road-specific contextual information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158:155–166, 2019.

[Wang *et al.*, 2017] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets, 2017.

[Yang *et al.*, 2022] Jianzhong Yang, Xiaoqing Ye, Bin Wu, Yanlei Gu, Ziyu Wang, Deguo Xia, and Jizhou Huang. Duare: Automatic road extraction with aerial images and trajectory data at baidu maps. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4321–4331, New York, NY, USA, 2022. Association for Computing Machinery.

[Yang *et al.*, 2024a] Ling Yang, Zhilin Huang, Zhilong Zhang, Zhongyi Liu, Shenda Hong, Wentao Zhang, Wenming Yang, Bin Cui, and Luxia Zhang. Graphusion: Latent diffusion for graph generation. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6358–6369, 2024.

[Yang *et al.*, 2024b] Xue Yang, Xiang Fan, Yichun Su, Qingfeng Guan, and Luliang Tang. Tr2rm: an urban road network generation model based on multisource big data. *International Journal of Digital Earth*, 17(1):2344596, 2024.

[Yao *et al.*, 2024] Xin Yao, Shaofu Lin, Xiliang Liu, Zhaolei Liu, and Xiaoying Zhi. Road extraction by using asymmetrical gan framework and structural similarity loss. In *Proceedings of the 16th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, IWCTS '23, page 70–77, New York, NY, USA, 2024. Association for Computing Machinery.

[You *et al.*, 2018] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models, 2018.

[Zang and Wang, 2020] Chengxi Zang and Fei Wang. Moflow: An invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20. ACM, August 2020.

[Zheng *et al.*, 2011] Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li. *Geolife GPS trajectory dataset - User Guide*, geolife gps trajectories 1.1 edition, July 2011.

[Zhou *et al.*, 2022] Mingting Zhou, Haigang Sui, Shanxiong Chen, Junyi Liu, Weiyue Shi, and Xu Chen. Large-scale road extraction from high-resolution remote sensing images based on a weakly-supervised structural and orientational consistency constraint network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 193:234–251, 2022.

[Zhu *et al.*, 2024] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. Controltraj: Controllable trajectory generation with topology-constrained diffusion model, 2024.