Holistic Semantic Representation for Navigational Trajectory Generation

Ji Cao[1], Tongya Zheng[2,3,*], Qinghong Guo[1], Yu Wang[1], Junshu Dai[1], Shunyu Liu[4], Jie Yang[1], Jie Song[1], Mingli Song[3,5]

[1]Zhejiang University
[2]Big Graph Center, Hangzhou City University
[3]State Key Laboratory of Blockchain and Data Security, Zhejiang University
[4]Nanyang Technological Univerisity
[5]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
{caoji2001, q.h_guo, yu.wang, djs, yang_jie, sjie, brooksong}@zju.edu.cn,
doujiang_zheng@163.com, shunyu.liu@ntu.edu.sg

Abstract

Trajectory generation has garnered significant attention from researchers in the field of spatio-temporal analysis, as it can generate substantial synthesized human mobility trajectories that enhance user privacy and alleviate data scarcity. However, existing trajectory generation methods often focus on improving trajectory generation quality from a singular perspective, lacking a comprehensive semantic understanding across various scales. Consequently, we are inspired to develop a **HO**listic **SE**mantic **R**epresentation (HOSER) framework for navigational trajectory generation. Given an origin-and-destination (OD) pair and the starting time point of a latent trajectory, we first propose a Road Network Encoder to expand the receptive field of road- and zone-level semantics. Second, we design a Multi-Granularity Trajectory Encoder to integrate the spatio-temporal semantics of the generated trajectory at both the point and trajectory levels. Finally, we employ a Destination-Oriented Navigator to seamlessly integrate destination-oriented guidance. Extensive experiments on three real-world datasets demonstrate that HOSER outperforms *state-of-the-art* baselines by a significant margin. Moreover, the model's performance in few-shot learning and zero-shot learning scenarios further verifies the effectiveness of our holistic semantic representation.

Code — https://github.com/caoji2001/HOSER
Extended version — https://arxiv.org/abs/2501.02737

1 Introduction

With the rapid development of Global Positioning Systems (GPS) and Geographic Information Systems (GIS), the number of human mobility trajectories has soared, significantly advancing research in spatio-temporal data mining, such as urban planning (Bao et al. 2017; Wang et al. 2023, 2024b,c), business location selection (Li et al. 2016), and travel time estimation (Reich et al. 2019; Wen et al. 2024). However, due to obstacles including privacy issues (Cao and Li 2021), government regulations (Chen et al. 2024a), and data processing costs (Zheng 2015), it is not easy for researchers to obtain high-quality real-world trajectory data. A promising solution to these challenges is trajectory generation, which not only meets privacy requirements but also allows for the creation of diverse high-fidelity trajectories. These trajectories are capable of producing similar data-analysis results, supporting broader research and application needs.

In addition to traditional statistical methods (Song et al. 2010; Jiang et al. 2016), deep learning has improved trajectory generation by encoding fine-grained human mobility semantics in high-dimensional representations. A series of trajectory generation methods employ RNNs and CNNs to capture spatio-temporal features in the trajectories, along with various generative models such as VAEs (Huang et al. 2019; Lestyán, Ács, and Biczók 2022), GANs (Cao and Li 2021; Wang et al. 2021), and diffusion models (Zhu et al. 2023b). In addition, another line of methods incorporates the connectivity of spatio-temporal points by embedding the topological semantics of road networks into trajectory generation (Feng et al. 2020; Wang et al. 2024a; Zhu et al. 2024). However, since experienced drivers (Yuan et al. 2010) often identify the fastest spatio-temporal routes to their destinations, previous methods have substantially overlooked the impact of destination on generated trajectories, resulting in a deviation from practical realities.

To the best of our knowledge, only TS-TrajGen (Jiang et al. 2023) incorporates both origin and destination locations in trajectory generation based on the A* algorithm. However, TS-TrajGen strictly adheres to the principle of the A* algorithm to separately model the semantics of the trajectory level and the road level in a two-tower paradigm, which hinders semantic sharing and end-to-end learning in trajectory generation. In general, existing methods lack a comprehensive understanding of the relationships between road segments, trajectories, and their origins and destinations.

Therefore, we are motivated by the semantic relationships to develop a **HO**listic **SE**mantic **R**epresentation (HOSER) framework for navigational trajectory generation. Using a bottom-up approach, we first derive long-range road semantics by partitioning road networks into a hierarchical topology. The trajectory representations are then encoded in a multi-granularity manner to integrate spatio-temporal dynamics with road-level semantics. Finally, we guide the trajectory generation process by incorporating both the semantic context of partial trajectories and the semantics of the

*Corresponding author.

destination. During the generation phase, HOSER iteratively predicts the probabilities of candidate road segments based on a progressively generated trajectory and its destination. Extensive experimental results and visualization analyses on three real-world trajectory datasets demonstrate that our proposed HOSER framework achieves significantly better trajectory generation quality than *state-of-the-art* baselines in both global and local level metrics. Furthermore, these generated trajectories can be effectively applied to downstream tasks, demonstrating their great potential to replace real trajectories for spatio-temporal data analysis. In addition, due to its outstanding architectural design, HOSER demonstrates exceptional performance in few-shot and zero-shot learning scenarios. In summary, our contributions can be summarized as follows:

- We identify a significant representation gap among road segments, trajectories, and their respective origins and destinations in trajectory generation, which is frequently overlooked by existing trajectory generation methods.
- We propose a novel **HO**listic **SE**mantic **R**epresentation (HOSER) framework, which is designed to bridge the aforementioned semantic gap in trajectory generation by holistically modeling human mobility patterns.
- We validate HOSER on three real-world trajectory datasets, demonstrating its ability to generate high-fidelity trajectories that surpass baselines at both the global and local levels. Furthermore, HOSER achieves satisfactory results in few-shot and zero-shot learning.

2 Preliminary

Definition 1: ***Road Network***. *The road network is represented as a directed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ denotes the set of road segments (nodes), and $\mathcal{E}$ denotes the set of intersections (edges) between adjacent road segments.*

Note that road segments are defined as nodes rather than edges, following the widely adopted settings in previous studies (Jepsen, Jensen, and Nielsen 2019; Wu et al. 2020).

Definition 2: ***Trajectory***. *We denote a trajectory as a sequence of spatio-temporal points $\tau = [x_1, x_2, \ldots, x_n]$. Each spatio-temporal point is represented as $x_i = (r_i, t_i)$, which is a pair of road segment ID and timestamp. The sequence ensures that each road segment r_i is reachable from the previous segment r_{i-1} for all $i \in [2, n]$.*

Note that not all adjacent segments are reachable due to the prescribed driving direction on each road segment.

Definition 3: ***Trajectory Generation***. *Given a set of real-world trajectories $\mathcal{T} = \{\tau^1, \tau^2, \ldots, \tau^m\}$, the objective of our trajectory generation task is to learn a θ-parameterized generative model G_θ. When given a triplet containing the origin road segment, the departure time, and the destination road segment $(r_{org}, t_{org}, r_{dest})$ as conditions, model G_θ is capable of generating a synthetic trajectory $[x_1, x_2, \ldots, x_n]$ such that $x_1 = (r_{org}, t_{org})$, and $r_n = r_{dest}$.*

Definition 4: ***Human Movement Modeling***. *We approach the problem of generating high-quality trajectories by modeling the human movement policy $\pi(a|s)$, which gives the probability of taking action a given the state s. Here, state s includes the current partial trajectory $x_{1:i} = [x_1, \ldots, x_i]$ and the destination r_{dest}, action a denotes moving to a currently reachable road segment, which can be written as:*

$$\pi(a|s) = P(r_{i+1}|x_{1:i}, r_{dest}). \tag{1}$$

Then the generation process can be seen as searching for the optimal trajectory with the maximum probability:

$$\begin{aligned} \max \prod_{i=1}^{n-1} \pi(a_i, s_i) = \max \prod_{i=1}^{n-1} P(r_{i+1} \mid x_{1:i}, r_{dest}), \\ \text{s.t.} \quad x_1 = (r_{org}, t_{org}), \quad r_n = r_{dest}. \end{aligned} \tag{2}$$

Our task is to use neural networks to estimate the movement strategy $P_\theta(r_{i+1}|x_{1:i}, r_{dest})$ and the corresponding timestamp t_{i+1} for the next spatio-temporal point.

3 Methodology

In this section, we detail the proposed HOSER framework, which predicts the next spatio-temporal point based on the current state and generates the trajectory between the given OD pair through a search-based method. As illustrated in Fig. 1, HOSER first employs a Road Network Encoder to model the road network at different levels. Based on the road network representation, a Multi-Granularity Trajectory Encoder is proposed to extract the semantic information from the current partial trajectory. To better incorporate prior knowledge of human mobility, a Destination-Oriented Navigator is used to seamlessly integrate the current partial trajectory semantics with the destination guidance.

3.1 Road Network Encoder

The road network is a fundamental part of the transportation system, and accurately modeling it is crucial for generating high-quality trajectories. However, designing effective representation learning methods remains challenging (Han et al. 2021). On the one hand, the road network's inherent topological structure means that connected road segments often correlate; on the other hand, non-connected road segments can still exhibit functional similarities, such as belonging to the same commercial or residential zone. Inspired by HRNR (Wu et al. 2020), we model the road network at both the road segment and zone levels to better capture the long-distance dependencies between road segments. Additionally, we use a deterministic road segment-to-zone allocation mechanism, which simplifies the complex allocation matrix learning process seen in HRNR (Wu et al. 2020).

Road-Level Semantic Representation. As outlined in Definition 1, we represent the road segments in the road network as nodes in the graph, with intersections between adjacent roads depicted as edges. The Road Network Encoder then encodes the road segments and intersections separately.

For the i-th road segment in the road network, we encode its road segment ID and its attributes (comprising four kinds: length, type, longitude, and latitude). These encoded attributes are then concatenated to form the road segment embedding $\boldsymbol{v}_i \in \mathbb{R}^d$, which can be written as $\boldsymbol{v}_i =$

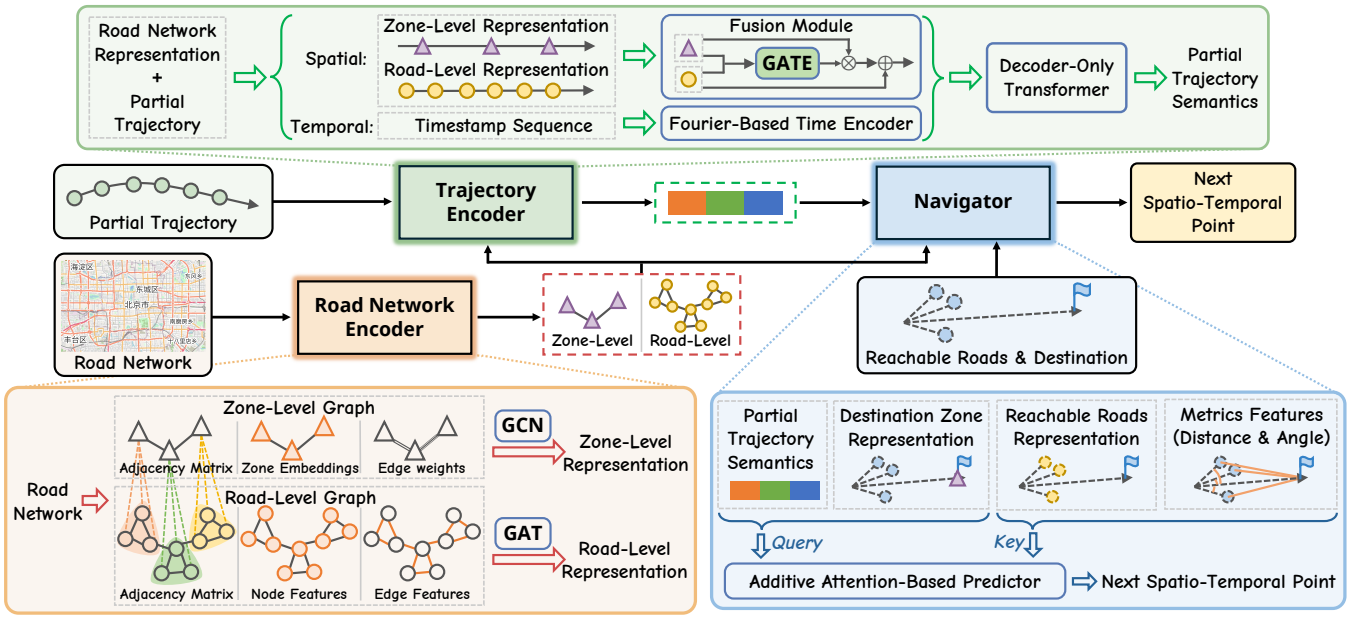

Figure 1: The overview of the proposed HOSER framework. The Road Network Encoder is responsible for modeling the road network at different levels. The Trajectory Encoder is used to extract semantic information from the partial trajectory, which is then fed into the Navigator and combined with destination guidance to generate the next spatio-temporal point.

$\boldsymbol{v}_{\text{ID}} \parallel \boldsymbol{v}_{\text{len}} \parallel \boldsymbol{v}_{\text{type}} \parallel \boldsymbol{v}_{\text{lon}} \parallel \boldsymbol{v}_{\text{lat}}$, where $\boldsymbol{v}_{(\cdot)}$ denotes the embedding vector for a certain type of the road segment feature and "$\parallel$" is the concatenation operation.

For the intersection between road segments i and j, we strengthen the directed road network by bidirected intersection embedding $\boldsymbol{e}_{ij} \in \mathbb{R}^d$, concatenating various features as $\boldsymbol{e}_{ij} = \mathbb{1}_{ij} \parallel \boldsymbol{\phi}_{ij}$, where $\mathbb{1}_{ij}$ and $\boldsymbol{\phi}_{ij}$ denote the embeddings for reachability and steering angle, respectively.

Then we employ GATv2 (Brody, Alon, and Yahav 2022) to fuse the aforementioned contextual embeddings of road segments and intersections, obtaining representations for the road segments within the road network at the $(\ell+1)$-th layer:

$$\boldsymbol{v}_i^{(\ell+1)} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{(\ell)} \boldsymbol{v}_j^{(\ell)} \boldsymbol{\Theta}_t, \tag{3}$$

where the attention coefficients $\alpha_{ij}^{(\ell)}$ are computed as:

$$\alpha_{ij}^{(\ell)} = \text{Softmax}\big(\sigma(\boldsymbol{v}_i \boldsymbol{\Theta}_s^{(\ell)} + \boldsymbol{v}_j \boldsymbol{\Theta}_t^{(\ell)} + \boldsymbol{e}_{ij})(\boldsymbol{a}^{(\ell)})^\top\big), \tag{4}$$

which incorporates both road- and intersection-aware semantics. Here, σ represents the LeakyReLU activation function, $\boldsymbol{\Theta}_s^{(\ell)}, \boldsymbol{\Theta}_t^{(\ell)} \in \mathbb{R}^{d \times d}$ are learnable transformation matrices, and $\boldsymbol{a}^{(\ell)} \in \mathbb{R}^d$ is a learnable projection vector.

Zone-Level Semantic Representation. To study the correlation between road segments that belong to the same functional zone, we first employ a multilevel graph partitioning algorithm (Sanders and Schulz 2013) to divide the road network into k zones based on its topological structure, ensuring that each road segment belongs to a single traffic zone. Each traffic zone contains several road segments, and the number of road segments in different zones is relatively balanced. Then for a given traffic zone z_i, we assign an embedding vector $\boldsymbol{z}_i \in \mathbb{R}^d$ to its ID.

After defining the traffic zones, our goal is to capture the relationships between adjacent zones. Under the assumption that a higher traffic flow between two zones indicates a stronger connection, we first calculate the traffic flow between adjacent zones using training data to construct the matrix $\boldsymbol{F} \in \mathbb{R}^{k \times k}$, where $\boldsymbol{F}_{ij}$ represents the traffic flow between zones i and j. Using this matrix, we apply GCN (Kipf and Welling 2017) to effectively obtain zone-level representations from their neighborhoods. Let $\boldsymbol{Z}^{(\ell)} = [\boldsymbol{z}_1^{(\ell)}, \boldsymbol{z}_2^{(\ell)}, \ldots, \boldsymbol{z}_k^{(\ell)}]^\top \in \mathbb{R}^{k \times d}$ denote the matrix of contextual representations of the traffic zones at the ℓ-th layer, then the update process can be expressed as:

$$\boldsymbol{Z}^{(\ell+1)} = \hat{\boldsymbol{D}}^{-1/2} \hat{\boldsymbol{F}} \hat{\boldsymbol{D}}^{-1/2} \boldsymbol{Z}^{(\ell)} \boldsymbol{\Theta}. \tag{5}$$

Here, $\hat{\boldsymbol{F}} = \boldsymbol{F} / \max(\boldsymbol{F}) + \boldsymbol{I}$ denotes the 0-1 normalized matrix $\boldsymbol{F}$ with inserted self-loops, $\hat{\boldsymbol{D}}_{ii} = \sum_{j=1}^{k} \hat{\boldsymbol{F}}_{ij}$ is its diagonal degree matrix, and $\boldsymbol{\Theta} \in \mathbb{R}^{d \times d}$ is a trainable weight matrix used for the linear transformation.

3.2 Multi-Granularity Trajectory Encoder

Trajectory data contains rich semantic information, but effectively extracting it involves overcoming challenges at various granularities. At a fine granularity, it requires precise modeling of spatio-temporal points within the trajectory. At a coarse granularity, it necessitates capturing the dependencies between these spatio-temporal points. To address these challenges, we propose the Multi-Granularity Trajectory Encoder, which integrates both levels of modeling to fully capture the trajectory's semantic information.

Spatio-temporal Point Semantics. For the i-th spatio-temporal point $x_i = (r_i, t_i)$ in the trajectory, let $\text{zone}(r_i)$ be the zone index of r_i. In the modeling of spatial features, we utilize the road- and zone-level road network representations obtained from the previous Road Network Encoder, denoted as $\boldsymbol{v}_{r_i}$ and $\boldsymbol{z}_{\text{zone}(r_i)}$, respectively. Subsequently, we utilize a gating unit to fuse the representations at different levels to obtain the spatial representation $\boldsymbol{x}_i^{\text{spatial}} \in \mathbb{R}^d$:

$$\boldsymbol{x}_i^{\text{spatial}} = \boldsymbol{v}_{r_i} + \text{Sigmoid}\big(\text{MLP}(\boldsymbol{v}_{r_i} \parallel \boldsymbol{z}_{\text{zone}(r_i)})\big) \cdot \boldsymbol{z}_{\text{zone}(r_i)}, \quad (6)$$

where MLP converts a vector of length $2d$ into a scalar. To model temporal features, we employ the Fourier encoding strategy (Xu et al. 2020) to obtain the temporal representation $\boldsymbol{x}_i^{\text{temporal}} \in \mathbb{R}^d$ for the i-th spatio-temporal point:

$$\boldsymbol{x}_i^{\text{temporal}} = \sqrt{1/2d}\,\big[\cos(\omega_l t_i), \sin(\omega_l t_i)\big]_{l=1}^{d/2}. \quad (7)$$

By concatenating the two aforementioned vectors, we obtain the representation of the i-th spatio-temporal point in the trajectory, denoted as $\boldsymbol{x}_i \in \mathbb{R}^{2d}$:

$$\boldsymbol{x}_i = \boldsymbol{x}_i^{\text{spatial}} \parallel \boldsymbol{x}_i^{\text{temporal}}. \quad (8)$$

Trajectory Semantics. After obtaining the representations of all spatio-temporal points in the trajectory, we employ a Decoder-Only Transformer (Radford et al. 2018) to extract the semantic information embedded within the trajectory. To more accurately capture the spatio-temporal relationships between these points, we introduce a relative position encoding technique (Shaw, Uszkoreit, and Vaswani 2018) based on spatio-temporal distances. Let $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$ be the representations of the input trajectory, we first encode the spatio-temporal interval between the input $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ as vectors $\boldsymbol{a}_{ij} \in \mathbb{R}^{2d/N_h}$:

$$\boldsymbol{a}_{ij} = d(r_i, r_j)\theta_d \parallel \Delta t(t_i, t_j)\theta_t. \quad (9)$$

Here, $d(r_i, r_j)$ represents the distance between road segment r_i and r_j, $\Delta t(t_i, t_j)$ represents the time interval between timestamp t_i and t_j; $\theta_d, \theta_t \in \mathbb{R}^{d/N_h}$ are projection vectors and N_h is the number of heads. The spatio-temporal relative encodings $\boldsymbol{a}_{ij}$ are built separately for the key and value (Eq. (10) and (11)) computation of the Decoder-Only Transformer, then the operation of one head in the multi-head self-attention is:

$$\boldsymbol{\tau}_{1:i}^h = \sum_{j=1}^{i} \alpha_{ij}(\boldsymbol{x}_j \boldsymbol{\Theta}_v + \boldsymbol{a}_{ij}^v), \quad (10)$$

where the attention coefficients α_{ij} are computed as follows:

$$\alpha_{ij} = \text{Softmax}\big(\boldsymbol{x}_i \boldsymbol{\Theta}_q (\boldsymbol{x}_j \boldsymbol{\Theta}_k + \boldsymbol{a}_{ij}^k)^\top / d_k\big). \quad (11)$$

Here, $\boldsymbol{\Theta}_q, \boldsymbol{\Theta}_k, \boldsymbol{\Theta}_v \in \mathbb{R}^{2d \times 2d/N_h}$ are the learnable matrices for query, key and value projections, respectively. The remainder of ours aligns with the structure of the Transformer Decoder. For the aforementioned input spatio-temporal points, we denote the output of the Decoder-Only Transformer as $(\boldsymbol{\tau}_{1:1}^{\text{out}}, \boldsymbol{\tau}_{1:2}^{\text{out}}, \cdots, \boldsymbol{\tau}_{1:n}^{\text{out}})$, where $\boldsymbol{\tau}_{1:i}^{\text{out}}$ corresponds to the semantics of the input trajectory $x_{1:i}$.

3.3 Destination-Oriented Navigator

Given that human movement frequently demonstrates clear intentionality and destination-oriented behavior, it is essential to integrate destination guidance within the modeling framework. To this end, we propose a novel Destination-Oriented Navigator, which predicts the next spatio-temporal point by effectively integrating partial trajectory features with destination guidance. Let the current partial trajectory be denoted as $x_{1:i} = [x_1, x_2, \ldots, x_i]$. Additionally, let $R(r_i)$ represent the set of road segments that are reachable from the current road segment r_i. When predicting the probability of a candidate road segment $r_{\text{c}} \in R(r_i)$ as the next step, we consider not only the semantics of the current partial trajectory $\boldsymbol{\tau}_{1:i}^{\text{out}}$ and the representations of the candidate road segment $\boldsymbol{v}_{r_{\text{c}}}$, but also the feature of the destination zone $\boldsymbol{z}_{z_{\text{dest}}}$ and the metric characteristics from the candidate road segment to the destination $\boldsymbol{h}_{r_{\text{c}}, r_{\text{dest}}}$ (including distances and angles, more details in Appendix A.1).

We then utilize an additive attention mechanism (Bahdanau, Cho, and Bengio 2015) to integrate the aforementioned features. Specifically, the semantics of the partial trajectory $\boldsymbol{\tau}_{1:i}^{\text{out}} \in \mathbb{R}^{2d}$ and the representation of the destination zone $\boldsymbol{z}_{z_{\text{dest}}} \in \mathbb{R}^d$ are used as queries, while the representations of candidate road segments $\boldsymbol{v}_{r_{\text{c}}} \in \mathbb{R}^d$ and the metric information from the candidate road segment to the destination $\boldsymbol{h}_{r_{\text{c}}, r_{\text{dest}}} \in \mathbb{R}^{2d}$ are used as keys, then the logit of the candidate road segment r_{c} can be written as:

$$p_{r_{\text{c}}} = \tanh\big((\boldsymbol{\tau}_{1:i}^{\text{out}} \parallel \boldsymbol{z}_{z_{\text{dest}}})\mathbf{W}_q + (\boldsymbol{v}_{r_{\text{c}}} \parallel \boldsymbol{h}_{r_{\text{c}}, r_{\text{dest}}})\mathbf{W}_k\big)\mathbf{w}_v^\top, \quad (12)$$

where $\mathbf{W}_q \in \mathbb{R}^{3d \times d}, \mathbf{W}_k \in \mathbb{R}^{3d \times d}, \mathbf{w}_v \in \mathbb{R}^d$ are the learnable parameters for query, key, and value, respectively. After applying the Softmax, the probability can be obtained:

$$\hat{P}_\theta(r_{\text{c}} | x_{1:i}, r_{\text{dest}}) = \frac{\exp(p_{r_{\text{c}}})}{\sum_{r'_{\text{c}} \in R(r_i)} \exp(p_{r_{\text{c}}})}. \quad (13)$$

To predict the timestamp t_{i+1} for the aforementioned candidate road segment r_{c}, we reformulate it as predicting the time interval to the next position $\Delta t_{i+1} = t_{i+1} - t_i$. This prediction utilizes both the semantics of the partial trajectory $x_{1:i}$ and the features of the candidate road segment r_{c}, employing a MLP to yield a single numerical output:

$$\Delta \hat{t}_{i+1} = \text{MLP}(\boldsymbol{\tau}_{1:i}^{\text{out}} \parallel \boldsymbol{v}_{r_{\text{c}}}). \quad (14)$$

3.4 End-to-End Learning

Optimization. During training, we predict the next reachable road segment and the corresponding time interval, based on partial real trajectories $x_{1:i}$ and the destination r_{dest}. The negative log-likelihood loss $\mathcal{L}_r$ is used for road segment prediction, while the mean absolute error loss $\mathcal{L}_t$ is used for interval time prediction. We add them together to optimize the model, written as:

$$\mathcal{L} = \frac{1}{n-1} \sum_{i=1}^{n-1} \underbrace{-\log \hat{P}_\theta(r_{i+1} | x_{1:i}, r_{\text{dest}})}_{\mathcal{L}_r} + \underbrace{|\Delta \hat{t}_{i+1} - \Delta t_{i+1}|}_{\mathcal{L}_t}. \quad (15)$$

Generation. Given the conditional information $(r_{\text{org}}, t_{\text{org}}, r_{\text{dest}})$, we search the trajectory with the maximum probability as the final generated trajectory, as described in Eq. (2). In practice, a heap is utilized to accelerate the process (more details in Appendix A.2).

4 Experiments

We conducted extensive experiments on three real-world trajectory datasets to validate the performance of HOSER. This section outlines the basic experimental setup and the main experimental results, while additional details are available in the Appendix due to space constraints. All experiments are conducted on a single NVIDIA RTX A6000 GPU.

4.1 Experimental Setups

Datasets. We assess the performance of HOSER and other baselines using three trajectory datasets from Beijing, Porto, and San Francisco. Each dataset is randomly split into training, validation, and test sets in a 7:1:2 ratio. Further dataset details are provided in Appendix B.1.

Evaluation Metrics. To comprehensively evaluate the quality of synthetic trajectories, we compare the trajectories generated by HOSER and other baselines with real trajectories from the following global and local perspectives, which follow the design in (Jiang et al. 2023; Wang et al. 2024a). For more details, please refer to Appendix B.2.

From the global perspective, we measure the overall distribution of the trajectories using the following three metrics: *Distance*, *Radius*, and *Duration*. To obtain quantitative results, we employ Jensen-Shannon divergence (JSD) to measure the distribution similarity of the three metrics.

From the local perspective, we exclusively compare the similarity between real and generated trajectories that have the same OD pairs, using the following three metrics for evaluation, i.e., *Hausdorff distance*, *DTW* and *EDR*.

Baselines. We compare HOSER with a series of baselines, including both traditional methods and a suite of deep learning-based methods. The former includes Markov (Gambs, Killijian, and del Prado Cortez 2012) and Dijkstra's algorithm (Dijkstra 1959), while the latter comprises SeqGAN (Yu et al. 2017), SVAE (Huang et al. 2019), MoveSim (Feng et al. 2020), TSG (Wang et al. 2021), TrajGen (Cao and Li 2021), DiffTraj (Zhu et al. 2023b), STEGA (Wang et al. 2024a), and TS-TrajGen (Jiang et al. 2023). See Appendix B.3 for more details.

4.2 Overall Performance

Quantitative Analysis. The global and local metrics on three real-world trajectory datasets are shown in Table 1. Due to space limitations, the DTW and EDR metrics for these three datasets are provided in Appendix C.1. The results demonstrate that compared to other *state-of-the-art* baselines, the trajectories generated by HOSER are closer to real-world trajectories in terms of both global and local similarity. This satisfactory result can be largely attributed to our comprehensive modeling of human mobility patterns. Among the baseline methods, DiffTraj demonstrates superior performance due to its advanced diffusion architecture. TS-TrajGen also achieves commendable results by integrating neural networks with the A* algorithm to model human mobility patterns. Additionally, and somewhat unexpectedly, Dijkstra's algorithm outperforms most deep learning-based approaches. This can be explained by the fact that people typically choose the quickest route to their destination

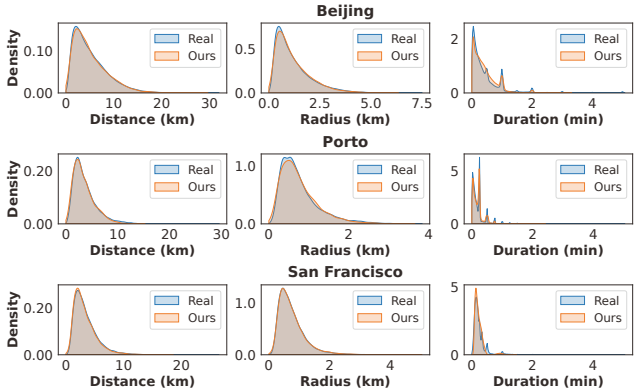

Figure 2: Visualization of metrics distributions.

based on their personal experience, and this route often approximates the shortest route (Yuan et al. 2010). However, due to factors such as traffic signals and road congestion, the quickest route does not always align with the shortest. HOSER effectively accounts for these discrepancies through its novel network architecture, resulting in superior performance and further highlighting the significance of modeling human mobility patterns holistically.

Discussion on Baselines with the Search Algorithm. Since our method, along with TS-TrajGen, utilizes a search-based paradigm (as described in Eq. (2)) to find the optimal trajectory with the highest probability between OD pairs, rather than relying solely on autoregressively generating entire trajectories based on previously generated points, we conduct additional experiments by modifying some baseline models to a search-based paradigm to investigate the impact of this paradigm on the effectiveness of trajectory generation. Specifically, we reformulate Markov, SeqGAN, MoveSim, and STEGA into search-based forms, appending "*" to denote the corresponding variants. It can be observed from Table 1 that after switching to a search-based paradigm, their performance has improved to some extent compared to the original approach. However, since they do not comprehensively model the semantics of human movement and instead simply use the partially generated trajectory to predict the next spatio-temporal point, there remains a gap between their performance and ours. This also indirectly suggests that the effectiveness of our method is not solely due to the adoption of the search-based paradigm.

Ablation Studies. HOSER comprises three key modules: the Road Network Encoder which models the road network at different levels, the Trajectory Encoder which extracts the semantics of partial trajectories, and the Navigator which seamlessly integrates destination guidance. To assess the contribution of each module to the overall performance, we perform ablation studies on three HOSER variants, each corresponding to the removal of one module, denoted as Ours w/o RNE, Ours w/o TrajE, and Ours w/o Nav, respectively (please refer to Appendix C.1 for more details).

Table 1 shows that performance declines when any module is removed, indicating their necessity for high-fidelity

Methods	Beijing				Porto				San Francisco			
	Global (↓)			Local (↓)	Global (↓)			Local (↓)	Global (↓)			Local (↓)
	Distance	Radius	Duration	Hausdorff	Distance	Radius	Duration	Hausdorff	Distance	Radius	Duration	Hausdorff
Markov	0.0048	0.0168	✗	0.8164	0.0047	0.0294	✗	0.7158	0.0052	0.0250	✗	0.7546
Dijkstra	0.0062	0.0064	✗	0.6239	0.0177	0.0099	✗	0.6011	0.0128	0.0060	✗	0.5567
SeqGAN	0.0068	0.0077	✗	0.6982	0.0089	0.0082	✗	0.7049	0.0092	0.0043	✗	0.6959
SVAE	0.0077	0.0124	✗	0.7180	0.0095	0.0250	✗	0.6669	0.0188	0.0422	✗	0.5908
MoveSim	0.3169	0.2091	✗	4.3434	0.0929	0.1015	✗	1.3911	0.1464	0.0946	✗	1.5704
TSG	0.4498	0.1471	✗	0.8636	0.1769	0.3037	✗	0.5676	0.3464	0.0952	✗	0.8720
TrajGen	0.2750	0.1553	✗	3.5120	0.2305	0.2287	✗	1.3774	0.2895	0.0652	✗	1.7050
DiffTraj	0.0033	0.0078	✗	0.6483	0.0070	0.0066	✗	0.6005	0.0040	0.0384	✗	0.6196
STEGA	0.0090	0.0331	0.2858	0.7473	0.0128	0.0877	0.2239	0.6231	0.0155	0.1376	0.3468	0.5984
TS-TrajGen	0.0172	0.0059	0.2580	0.9618	0.0050	0.0052	0.2023	0.7153	0.0143	0.0062	0.2931	0.7605
Markov*	0.0034	0.0037	✗	0.6086	**0.0041**	0.0092	✗	0.6410	0.0049	0.0037	✗	0.6161
SeqGAN*	0.0029	0.0032	✗	0.6099	0.0055	0.0039	✗	0.6903	0.0055	0.0043	✗	0.6605
MoveSim*	0.0651	0.0099	✗	1.3311	0.0309	0.0108	✗	0.9110	0.0292	0.0074	✗	0.7708
STEGA*	0.0086	0.0054	0.2747	0.6923	0.0528	0.0353	0.1819	1.0897	0.0112	0.0082	0.3820	0.7216
Ours w/o RNE	0.0024	0.0026	0.0274	0.5694	0.0050	0.0038	0.0242	0.5993	0.0037	0.0036	0.0499	0.5403
Ours w/o TrajE	0.0027	0.0029	0.0268	0.5650	0.0051	0.0041	0.0219	0.5956	0.0040	0.0039	0.0487	0.5381
Ours w/o Nav	0.0029	0.0030	0.0259	0.5704	0.0055	0.0043	0.0237	0.6263	0.0045	0.0040	0.0359	0.5661
Ours	**0.0024**	**0.0025**	**0.0245**	**0.5503**	0.0045	**0.0033**	**0.0197**	**0.5746**	**0.0033**	**0.0034**	**0.0249**	**0.5351**

Table 1: Average performance of 5 random seeds (0 to 4) on three real-world trajectory datasets in terms of global and local level metrics. The method names followed by an asterisk (*) indicate the corresponding search versions. The best one is denoted by **boldface** and the second-best is denoted by underline. Unsupported metrics are denoted by ✗. ↓ denotes lower is better.

trajectory generation. Among them, the removal of the Navigator has the most significant impact on model performance, underscoring the importance of incorporating destination guidance in trajectory generation. Moreover, the significant drop in the Duration metric after removing the Road Network Encoder highlights the critical role of road network representation in accurately predicting travel time. Lastly, the removal of the Trajectory Encoder results in a decline across all performance metrics, indicating that generating reliable trajectories requires not only destination information but also historical trajectory data.

Visualization Analysis. To intuitively compare the similarity between real and generated trajectories, we visualize the distribution of metrics including *Distance*, *Radius* and *Duration* of the trajectories, as shown in Fig. 2. Specifically, for the *Distance* and *Radius* metrics, the generated data not only captures the peak values but also aligns well with the long-tail distributions of the real data. For the *Duration* metric, the synthetic data successfully replicates the multimodal characteristics observed in the real data, further confirming the reliability of the synthetic data.

We also visualize both the real trajectories and the generated trajectories to facilitate a more intuitive comparison. Fig. 3 presents a heatmap illustrating the distribution of real trajectories alongside those generated by the top three methods in Beijing. Since Dijkstra's algorithm directly uses the shortest path between OD pairs to generate trajectories, the frequency of road segment access is relatively uniform. In addition, DiffTraj fails to fully consider the topological structure of the road network, resulting in a significant discrepancy from actual data. In contrast, our method nearly matches the original trajectories perfectly, indicating a marked improvement over other methods.

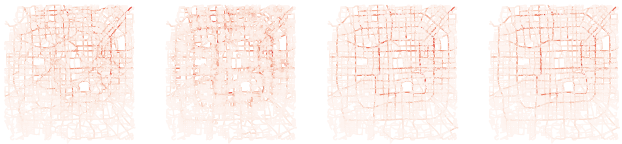

Figure 3: Visualization of the trajectories in Beijing (a larger view for Beijing, as well as for the other two cities, can be found in Appendix C.1).

4.3 Utility of Generated Data

Since the generated trajectories are ultimately used to analyze human mobility patterns, their utility determines whether the data generation method is feasible. Here, we evaluate the utility of the generated trajectories through a well-known location prediction task. We train three advanced prediction models: DeepMove (Feng et al. 2018), Flashback (Yang et al. 2020), and LSTPM (Sun et al. 2020), using both real and generated data, and compare their performance. As shown in Table 2, DeepMove and LSTPM perform comparably with synthetic and real data, while FlashBack shows slight deviations due to its reliance on timestamp information, indicating room for improvement. Nevertheless, these results highlight the potential of generated trajectories as viable substitutes for real data (please refer to Appendix C.2 for the results of other baselines).

Datasets	Methods	Acc@5	MRR
Beijing	DeepMove	0.776 / 0.804	0.697 / 0.728
	Flashback	0.749 / 0.782	0.676 / 0.706
	LSTPM	0.761 / 0.795	0.694 / 0.713
Porto	DeepMove	0.888 / 0.929	0.758 / 0.780
	Flashback	0.812 / 0.895	0.698 / 0.761
	LSTPM	0.860 / 0.914	0.741 / 0.778
San Francisco	DeepMove	0.797 / 0.847	0.673 / 0.698
	Flashback	0.746 / 0.815	0.625 / 0.685
	LSTPM	0.774 / 0.816	0.667 / 0.680

Table 2: Comparison of data utility based on location prediction task, results are expressed as (generated / real).

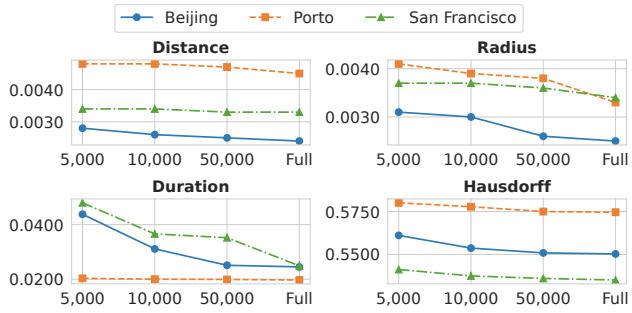

Figure 4: HOSER's performance with varying amounts of training data across three trajectory datasets. "Full" denotes the complete dataset, with sizes of 629,380, 481,359, and 205,116 for Beijing, Porto, and San Francisco, respectively.

4.4 Few-Shot and Zero-Shot Learning Tests

Considering the scarcity of real-world trajectory data, the few-shot and zero-shot capabilities of the trajectory generation model are crucial. Therefore, we evaluate the few-shot and zero-shot capabilities of HOSER.

For few-shot learning, we randomly sample 5,000, 10,000, and 50,000 trajectories for training and compare the performance of the generated trajectories. As shown in Fig. 4, our model's precise representation of road networks and incorporation of human mobility patterns enables strong performance even with limited data, which improves as the size of the training dataset increases.

For zero-shot learning, among the baselines, Dijkstra, TS-TrajGen, and DiffTraj perform well in general trajectory generation tasks. However, as TS-TrajGen lacks support for zero-shot learning, we compare HOSER specifically with Dijkstra and DiffTraj in this context. As shown in Table 3, HOSER excels in zero-shot learning tasks due to its holistic semantic modeling of human mobility patterns, which effectively captures and leverages the universality of policies employed in human mobility, enhancing its generalizability.

5 Related Work

Trajectory Generation. Trajectory synthesis methods fall into two categories: model-based and model-free. Model-based methods (Song et al. 2010; Jiang et al. 2016) assume interpretable mobility patterns but often oversimplify real-world complexity. Model-free methods are further classified into grid-based, coordinate point-based, and road segment-based approaches. Grid-based methods generate matrix trajectory data by dividing the map into grids (Ouyang et al. 2018; Cao and Li 2021). Coordinate point-based methods map GPS points to high-dimensional spaces via linear transformations and apply generative models (Kingma 2014; Goodfellow et al. 2014; Ho, Jain, and Abbeel 2020; Liu et al. 2024), including VAE (Huang et al. 2019), GAN (Wang et al. 2021), and diffusion-based models (Zhu et al. 2023b,a, 2024). Road segment-based methods (Feng et al. 2020; Cao and Li 2021; Jiang et al. 2023; Wang et al. 2024a) embed road segments as tokens. However, existing methods struggle to balance different aspects of human mobility patterns.

Methods	Global (↓)			Local (↓)
	Distance	Radius	Duration	Hausdorff
Dijkstra	0.0177	0.0099	✗	0.6011
DiffTraj	0.0633	0.2521	✗	0.8023
HOSER	**0.0052**	**0.0053**	**0.0223**	**0.5843**

Table 3: Results of zero-shot learning. DiffTraj and HOSER are trained in Beijing and generated in Porto, while Dijkstra is generated directly in Porto.

Road Network Representation Learning. Road networks are crucial for intelligent transportation tasks like spatial query processing (Huang et al. 2021; Zhao et al. 2022; Chang et al. 2023), travel time estimation (Yuan, Li, and Bao 2022), and traffic forecasting (Guo et al. 2021). Early studies (Jepsen et al. 2018; Jepsen, Jensen, and Nielsen 2019; Wang et al. 2019, 2020; Wu et al. 2020) leverage GNNs (Kipf and Welling 2017; Veličković et al. 2018; Zheng et al. 2022, 2023) for road network representation learning. Recent work (Chen et al. 2021; Mao et al. 2022; Schestakov, Heinemeyer, and Demidova 2023; Zhang and Long 2023; Chen et al. 2024b) enhances road representations by integrating trajectory data. Nonetheless, applying these methods to trajectory generation remains challenging, demanding specialized integration models.

6 Conclusion

This paper introduces HOSER, a novel trajectory generation framework enhanced with holistic semantic representation, which incorporates multi-level road network encoding, multi-granularity trajectory representation, and destination guidance modeling. Extensive experiments demonstrate that our method surpasses *state-of-the-art* baselines in global and local similarity metrics. The synthetic trajectories are effective for downstream tasks, demonstrating their potential as real-data substitutes. Additionally, HOSER performs well in few-shot and zero-shot learning. In the future, we will investigate the division of dense spatio-temporal points along a trajectory into coarse-grained activity sequences and fine-grained road segment sequences, facilitating the semantic representations of trajectories at varying scales.

Acknowledgments

This work is supported by the Zhejiang Province "JianBingLingYan+X" Research and Development Plan (2024C01114), Zhejiang Province High-Level Talents Special Support Program "Leading Talent of Technological Innovation of Ten-Thousands Talents Program" (No.2022R52046), the Fundamental Research Funds for the Central Universities (No.226-2024-00058), and the Scientific Research Fund of Zhejiang Provincial Education Department (Grant No.Y202457035). Also, we thank Bayou Tech (Hong Kong) Limited for providing the data used in this paper free of charge.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Bao, J.; He, T.; Ruan, S.; Li, Y.; and Zheng, Y. 2017. Planning bike lanes based on sharing-bikes' trajectories. In *SIGKDD*.

Brody, S.; Alon, U.; and Yahav, E. 2022. How attentive are graph attention networks? In *ICLR*.

Cao, C.; and Li, M. 2021. Generating mobility trajectories with retained data utility. In *SIGKDD*.

Chang, Y.; Qi, J.; Liang, Y.; and Tanin, E. 2023. Contrastive trajectory similarity learning with dual-feature attention. In *ICDE*.

Chen, W.; Liang, Y.; Zhu, Y.; Chang, Y.; Luo, K.; Wen, H.; Li, L.; Yu, Y.; Wen, Q.; Chen, C.; et al. 2024a. Deep learning for trajectory data management and mining: A survey and beyond. *arXiv preprint arXiv:2403.14151*.

Chen, Y.; Li, X.; Cong, G.; Bao, Z.; and Long, C. 2024b. Semantic-enhanced representation learning for road networks with temporal dynamics. *arXiv preprint arXiv:2403.11495*.

Chen, Y.; Li, X.; Cong, G.; Bao, Z.; Long, C.; Liu, Y.; Chandran, A. K.; and Ellison, R. 2021. Robust road network representation learning: When traffic patterns meet traveling semantics. In *CIKM*.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numer. Math.*, 1: 269–271.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW*.

Feng, J.; Yang, Z.; Xu, F.; Yu, H.; Wang, M.; and Li, Y. 2020. Learning to simulate human mobility. In *SIGKDD*.

Gambs, S.; Killijian, M.-O.; and del Prado Cortez, M. N. 2012. Next place prediction using mobility Markov chains. In *Proceedings of the first workshop on measurement, privacy, and mobility*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*.

Guo, S.; Lin, Y.; Wan, H.; Li, X.; and Cong, G. 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Trans. Knowl. Data Eng.*, 34(11): 5415–5428.

Han, P.; Wang, J.; Yao, D.; Shang, S.; and Zhang, X. 2021. A graph-based approach for trajectory similarity computation in spatial networks. In *SIGKDD*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

Huang, D.; Song, X.; Fan, Z.; Jiang, R.; Shibasaki, R.; Zhang, Y.; Wang, H.; and Kato, Y. 2019. A variational autoencoder based generative model of urban human mobility. In *MIPR*.

Huang, S.; Wang, Y.; Zhao, T.; and Li, G. 2021. A learning-based method for computing shortest path distances on road networks. In *ICDE*.

Jepsen, T. S.; Jensen, C. S.; and Nielsen, T. D. 2019. Graph convolutional networks for road networks. In *SIGSPATIAL*.

Jepsen, T. S.; Jensen, C. S.; Nielsen, T. D.; and Torp, K. 2018. On network embedding for machine learning on road networks: A case study on the danish road network. In *Big Data*.

Jiang, S.; Yang, Y.; Gupta, S.; Veneziano, D.; Athavale, S.; and González, M. C. 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proc. Natl. Acad. Sci. U.S.A.*, 113(37): E5370–E5378.

Jiang, W.; Zhao, W. X.; Wang, J.; and Jiang, J. 2023. Continuous trajectory generation based on two-stage GAN. In *AAAI*.

Kingma, D. P. 2014. Auto-encoding variational bayes. In *ICLR*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Lestyán, S.; Ács, G.; and Biczók, G. 2022. In search of lost utility: Private location data. In *PETS*.

Li, Y.; Bao, J.; Li, Y.; Wu, Y.; Gong, Z.; and Zheng, Y. 2016. Mining the Most Influential k-Location Set from Massive Trajectories. In *SIGSPATIAL*.

Liu, S.; Song, J.; Zhou, Y.; Yu, N.; Chen, K.; Feng, Z.; and Song, M. 2024. Interaction pattern disentangling for multi-agent reinforcement learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12): 8157–8172.

Mao, Z.; Li, Z.; Li, D.; Bai, L.; and Zhao, R. 2022. Jointly contrastive representation learning on road network and trajectory. In *CIKM*.

Ouyang, K.; Shokri, R.; Rosenblum, D. S.; and Yang, W. 2018. A non-parametric generative model for human trajectories. In *IJCAI*.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

Reich, T.; Budka, M.; Robbins, D.; and Hulbert, D. 2019. Survey of ETA prediction methods in public transport networks. *arXiv preprint arXiv:1904.05037*.

Sanders, P.; and Schulz, C. 2013. Think locally, act globally: Highly balanced graph partitioning. In *International Symposium on Experimental Algorithms*.

Schestakov, S.; Heinemeyer, P.; and Demidova, E. 2023. Road network representation learning with vehicle trajectories. In *PAKDD*.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-attention with relative position representations. In *NAACL*.

Song, C.; Koren, T.; Wang, P.; and Barabási, A.-L. 2010. Modelling the scaling properties of human mobility. *Nat. Phys.*, 6(10): 818–823.

Sun, K.; Qian, T.; Chen, T.; Liang, Y.; Nguyen, Q. V. H.; and Yin, H. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *AAAI*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Wang, D.; Wu, L.; Zhang, D.; Zhou, J.; Sun, L.; and Fu, Y. 2023. Human-instructed deep hierarchical generative learning for automated urban planning. In *AAAI*.

Wang, M.-x.; Lee, W.-C.; Fu, T.-y.; and Yu, G. 2019. Learning embeddings of intersections on road networks. In *SIGSPATIAL*.

Wang, M.-X.; Lee, W.-C.; Fu, T.-Y.; and Yu, G. 2020. On representation learning for road networks. *ACM Trans. Intell. Syst. Technol.*, 12(1): 1–27.

Wang, X.; Liu, X.; Lu, Z.; and Yang, H. 2021. Large scale GPS trajectory generation using map based on two stage GAN. *J. Data Sci.*, 19(1): 126–141.

Wang, Y.; Cao, J.; Huang, W.; Liu, Z.; Zheng, T.; and Song, M. 2024a. Spatiotemporal gated traffic trajectory simulation with semantic-aware graph learning. *Inf. Fusion*, 108: 102404.

Wang, Y.; Zheng, T.; Liang, Y.; Liu, S.; and Song, M. 2024b. Cola: Cross-city mobility transformer for human trajectory simulation. In *WWW*.

Wang, Y.; Zheng, T.; Liu, S.; Feng, Z.; Chen, K.; Hao, Y.; and Song, M. 2024c. Spatiotemporal-augmented graph neural networks for human mobility simulation. *IEEE Trans. Knowl. Data Eng.*, 36(11): 7074–7086.

Wen, H.; Lin, Y.; Wu, L.; Mao, X.; Cai, T.; Hou, Y.; Guo, S.; Liang, Y.; Jin, G.; Zhao, Y.; Zimmermann, R.; Ye, J.; and Wan, H. 2024. A survey on service route and time prediction in instant delivery: Taxonomy, progress, and prospects. *IEEE Trans. Knowl. Data Eng.*, 36(12): 7516–7535.

Wu, N.; Zhao, X. W.; Wang, J.; and Pan, D. 2020. Learning effective road network representation with hierarchical graph neural networks. In *SIGKDD*.

Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; and Achan, K. 2020. Inductive representation learning on temporal graphs. In *ICLR*.

Yang, D.; Fankhauser, B.; Rosso, P.; and Cudre-Mauroux, P. 2020. Location prediction over sparse user mobility traces using RNNs: Flashback in hidden states! In *IJCAI*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Yuan, H.; Li, G.; and Bao, Z. 2022. Route travel time estimation on a road network revisited: Heterogeneity, proximity, periodicity and dynamicity. In *VLDB*.

Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; and Huang, Y. 2010. T-drive: Driving directions based on taxi trajectories. In *SIGSPATIAL*.

Zhang, L.; and Long, C. 2023. Road network representation learning: A dual graph-based approach. *ACM Trans. Knowl. Discov. Data*, 17(9): 1–25.

Zhao, T.; Huang, S.; Wang, Y.; Chai, C.; and Li, G. 2022. RNE: Computing shortest paths using road network embedding. *VLDB J.*, 31(3): 507–528.

Zheng, T.; Feng, Z.; Zhang, T.; Hao, Y.; Song, M.; Wang, X.; Wang, X.; Zhao, J.; and Chen, C. 2022. Transition propagation graph neural networks for temporal networks. *IEEE Trans. Neural Networks Learn. Syst.*, 35(4): 4567–4579.

Zheng, T.; Wang, X.; Feng, Z.; Song, J.; Hao, Y.; Song, M.; Wang, X.; Wang, X.; and Chen, C. 2023. Temporal aggregation and propagation graph neural networks for dynamic representation. *IEEE Trans. Knowl. Data Eng.*, 35(10): 10151–10165.

Zheng, Y. 2015. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.*, 6(3): 1–41.

Zhu, Y.; Ye, Y.; Wu, Y.; Zhao, X.; and Yu, J. 2023a. SynMob: creating high-fidelity synthetic GPS trajectory dataset for urban mobility analysis. In *NeurIPS*.

Zhu, Y.; Ye, Y.; Zhang, S.; Zhao, X.; and Yu, J. 2023b. DiffTraj: generating GPS trajectory with diffusion probabilistic model. In *NeurIPS*.

Zhu, Y.; Yu, J. J.; Zhao, X.; Liu, Q.; Ye, Y.; Chen, W.; Zhang, Z.; Wei, X.; and Liang, Y. 2024. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *SIGKDD*.