A Graph-based Representation Framework for Trajectory Recovery via Spatiotemporal Interval-Informed Seq2Seq

Yaya Zhao[1] , **Kaiqi Zhao**[2] , **Zhiqian Chen**[3] , **Yuanyuan Zhang**[4] , **Yalei Du**[4] and **Xiaoling Lu**[1*]
[1]Center for Applied Statistics, School of Statistics, Renmin University of China
[2]The University of Auckland
[3]Mississippi State University
[4]Beijing Baixingkefu Network Technology Co., Ltd.
{zhaoyaya, xiaolinglu}@ruc.edu.cn, kaiqi.zhao@auckland.ac.nz, zchen@cse.msstate.edu,
zhang.huanzhiyuan@gmail.com, yaleidu@163.com

Abstract

The prevalent issue in urban trajectory data usage, notably in low-sample rate datasets, revolves around the accuracy of travel time estimations, traffic flow predictions, and trajectory similarity measurements. Conventional methods, often relying on simplistic mixes of static road networks and raw GPS data, fail to adequately integrate both network and trajectory dimensions. Addressing this, the innovative GRFTrajRec framework offers a graph-based solution for trajectory recovery. Its key feature is a trajectory-aware graph representation, enhancing the understanding of trajectory-road network interactions and facilitating the extraction of detailed embedding features for road segments. Additionally, GRFTrajRec's trajectory representation acutely captures spatiotemporal attributes of trajectory points. Central to this framework is a novel spatiotemporal interval-informed seq2seq model, integrating an attention-enhanced transformer and a feature differences-aware decoder. This model specifically excels in handling spatiotemporal intervals, crucial for restoring missing GPS points in low-sample datasets. Validated through extensive experiments on three large real-life trajectory datasets, GRFTrajRec has proven its efficacy in significantly boosting prediction accuracy and spatial consistency.

1 Introduction

Low-sampling rate datasets present a formidable obstacle when it comes to analyzing and utilizing trajectory data in urban settings. Such limitations often lead to significant inaccuracies and spatial disparities in vital tasks such as estimating travel times [Wang *et al.*, 2022; Zhang *et al.*, 2018], forecasting traffic flow [Li and Zhu, 2021; Lan *et al.*, 2022] and trajectory similarity measurement [Yao *et al.*, 2022; Han *et al.*, 2021]. Therefore, it becomes imperative to develop methods for enhancing the sampling rate by effectively recovering the missing points within a trajectory.

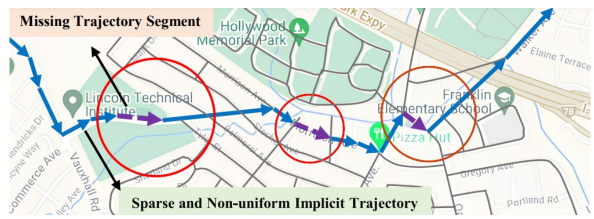


Figure 1: An example demonstrating the significance of attributing dynamic trajectory information to road network representation.

Recently, there has been a surge in deep learning-based models for trajectory recovery, such as MTrajRec [Ren *et al.*, 2021] and RNTrajRec [Chen *et al.*, 2023]. These methods adopt a sequence-to-sequence [Sutskever *et al.*, 2014] architecture, featuring an encoder model responsible for generating representations of the input trajectory and a decoder model tasked with recovering the trajectory point by point. Representation learning, a deep learning technique that automatically discovers meaningful patterns from raw data, has been widely used to model trajectory data [Jiang *et al.*, 2023] and road networks [Zhang and Zhao, 2021; Fu and Lee, 2020], in a wide range of downstream tasks [Li *et al.*, 2018; Yang *et al.*, 2021]. Furthermore, it plays a vital role in trajectory recovery. For instance, RNTrajRec utilizes road network representation learning and trajectory representation learning within the encoder to effectively capture both temporal and spatial features for each GPS point in the trajectory.

Nevertheless, it is important to note that all existing works still grapple with three significant limitations: **(1) Ignorance of dynamic trajectory-road network interactions:** Much of the existing research tends to either overlook the road network entirely or solely focus on static road networks, which contain only the fixed topology of the road network and do not account for the dynamic trajectory information traversing it. However, this trajectory-aware road network interplay plays a crucial role in the task of trajectory recovery. For instance, as illustrated in Figure 1, when provided with the road segments where the blue trajectory points are located, it becomes necessary to recover the missing road segments where the purple trajectory points circled in red are situated. In this context, road segments passing through or close to the blue trajectory gain higher consideration for selection compared

[*]The Corresponding Author

to other road segments farther away from the blue trajectory points. **(2) Neglect of extracting the overall and synergistic spatiotemporal trajectory representation:** Most studies solely rely on grid information or raw GPS points as input, resulting in extracted trajectory representations that lack comprehensiveness and synergy. **(3) Failure of considering the crucial spatiotemporal intervals when employing a seq2seq:** Many prior studies have relied on simple seq2seq models, often overlooking the crucial role of spatiotemporal intervals between trajectory points. However, it is essential to recognize that these intervals hold significance. For instance, smaller intervals between the first and second points within a trajectory imply similar embeddings, while larger intervals between the first and last points indicate weaker correlations. Failing to account for these spatiotemporal intervals between points can impede the effective extraction of comprehensive spatiotemporal contextual information among trajectory points, thereby hindering the recovery accuracy.

To overcome the three limitations inherent in existing trajectory recovery methods, we introduce a groundbreaking graph-based representation framework with a spatiotemporal interval-informed seq2seq model, known as GRFTrajRec. GRFTrajRec takes its first stride by harnessing trajectory-aware graph representation, enhancing the understanding of trajectory road network interactions and facilitating the extraction of detailed embedding features for road segments while considering dynamic trajectory information. Furthermore, it employs trajectory representation to capture overall and synergistic spatiotemporal features at each trajectory point. Finally, accounting for spatiotemporal intervals between points, GRFTrajRec utilizes a spatiotemporal interval-informed seq2seq model to effectively integrate and leverage both road and trajectory representations, thereby enabling the precise recovery of missing GPS points. In summary, our key contributions can be outlined as follows:

- We propose a novel framework, namely GRFTrajRec[1]. To the best of our knowledge, GRNTrajRec is the first attempt to use trajectory-road network interaction for the task of trajectory recovery.
- We introduce a trajectory-aware graph representation for extracting advanced road embedding, enabling a deep understanding of the interplay between trajectories and roads, considering dynamic trajectory information. Additionally, we leverage trajectory representation to capture the spatiotemporal features of each trajectory point.
- We propose a spatiotemporal interval-informed seq2seq model that combines an attention-enhanced transformer and a feature differences-aware decoder, all considering the spatiotemporal intervals between points. This model enhances the integration of road and trajectory representations, extracting comprehensive spatiotemporal contextual information between trajectory points.
- Extensive experimental results obtained from three real-life datasets conclusively and unequivocally demonstrate that GRFTrajRec outperforms all competitors in both prediction accuracy and spatial consistency.

[1]Source codes: https://github.com/zhaoyaya1234/GRFTrajRec.

2 Related Work

Road Network Representation Learning. Many existing works consider the road network as a directed graph. Models like Node2vec [Grover and Leskovec, 2016] and DeepWalk [Perozzi *et al.*, 2014] have been introduced to represent road segments as latent embeddings. With the rapid advancement of graph neural networks (GNNs), various graph convolutional networks, such as GCN [Kipf and Welling, 2017], GraphSage [Hamilton *et al.*, 2017], and GAT [Veličković *et al.*, 2018], have proven to be suitable for road network representation. Recent studies have focused on road network representation learning, notably SGMP [Zhang and Zhao, 2021] and Trembr [Fu and Lee, 2020]. SGMP offers a framework for spatial network representation, while Trembr's Road2Vec model learns embeddings for road segments by understanding the relationships among these segments.

Trajectory Representation Learning. In recent years, trajectory representation learning has garnered widespread attention, as evidenced in [Li *et al.*, 2023; Jarboui and Perchet, 2021]. Prominent models in this field include DeepTTE [Wang *et al.*, 2022], T2vec [Li *et al.*, 2018], ST2vec [Fang *et al.*, 2022], and Start [Jiang *et al.*, 2023]. DeepTTE employs LSTM modules to capture temporal dependencies and generate trajectory representations. T2vec introduces a pioneering deep learning model for trajectory similarity learning, utilizing BiLSTM [Graves and Graves, 2012] to model temporal dependencies. ST2vec [Fang *et al.*, 2022] focuses on encoding both spatial and temporal information within trajectories. Most recently, Start [Jiang *et al.*, 2023] presents a graph-based trajectory representation method, complemented by an innovative spatial network based on GAT [Veličković *et al.*, 2018].

Trajectory Recovery. Various studies, including [Xia *et al.*, 2022; Zhang *et al.*, 2022; Si *et al.*, 2023; Chen *et al.*, 2023], have proposed innovative solutions to address the trajectory recovery challenge. Notably, DHTR [Wang *et al.*, 2020] proposes a two-stage solution that first recovers a high-sample trajectory and then uses a map matching algorithm (i.e., HMM [Newson and Krumm, 2009]) to recover the actual GPS locations. Another significant advancement, MTrajRec, as detailed in [Ren *et al.*, 2021], employs a sequence-to-sequence model [Sutskever *et al.*, 2014]. It has outperformed two-stage techniques and is widely adopted in subsequent trajectory recovery research. However, most of these studies overlook the crucial road network structure aspect, which could enhance accuracy. RNTrajRec [Chen *et al.*, 2023] addresses this by introducing GridGNN for road segment feature learning and GPSFormer for detailed trajectory analysis. These components enable a multi-task decoder to efficiently reconstruct missing GPS points using encoder outputs.

3 Preliminaries

Definition 1 (Road Network)**.** *A Road Network is modelled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of road segments and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ captures the connectivity of these road segments.*

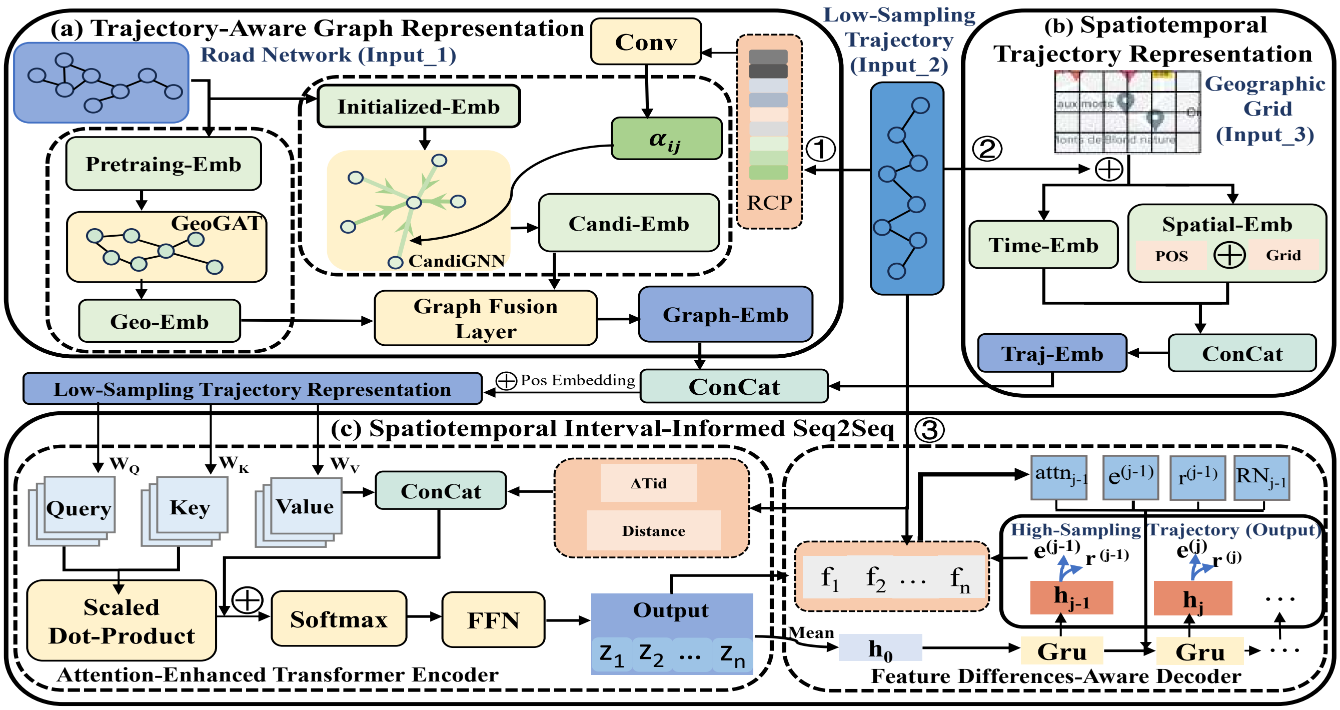


Figure 2: The framework of GRFTrajRec. (a) Trajectory-Aware Graph Representation: Enabling road segment representation enriched with missing point information through CandiGNN. (b) Spatiotemporal Trajectory Representation: Extracting nuanced and synergistic initial low-sampling trajectory representation information. (c) Spatiotemporal Interval-Informed Seq2Seq: Extracting the representation from (a) and (b), allowing for precise recovery of missing GPS points point by point by accounting for spatiotemporal intervals at different points.

Definition 2 (Trajectory)**.** *A trajectory* τ *can be defined as a sequence of GPS positions with timestamps, i.e.,* $\tau = (p_1, p_2, \cdots, p_n)$, *where* $p_i = \langle lat^{(i)}, lng^{(i)}, t^{(i)} \rangle, \forall i, 1 \leq i \leq n$, *which captures the latitude and longitude of the GPS position at timestamp* $t^{(i)}$.

Definition 3 (Road Segment Candidate Probability (RCP^{τ}))**.** *Given a raw trajectory* $\tau = (p_1, p_2, \cdots, p_n)$ *and a specified radius* R, *the road segment candidate frequency* N_j^{τ} *for road segment* $v_j \in \mathcal{V}$ *is defined as follows:* $N_j^{\tau} = \sum_{i=1}^{n} \mathrm{I}[v_j$ *is within a radius of* R *from* $p_i]$, *where* $I[\cdot]$ *is the identity function. The standardized* N_j^{τ} *is defined as the road segment candidate probability* RCP_j^{τ}, *with* $\sum_{j=1}^{|\mathcal{V}|} RCP_j^{\tau} = 1$.

Definition 4 (Map-matched Trajectory Point)**.** *Given a trajectory point* p_j *and a map-matching function* $M(\cdot)$, *the map-matched trajectory point* a_j *for* p_j *is defined as* $a_j = M(p_j) = \langle e^{(j)}, r^{(j)}, t^{(j)}, lat_{pre}^{(j)}, lng_{pre}^{(j)} \rangle$, *where* $e^{(j)}$ *is the matched road segment,* $r^{(j)}$ *is the moving ratio (representing the ratio of moving distance over the length of the road segment), and* $t^{(j)}$ *is the timestamp. The coordinates* $\langle lat_{pre}^{(j)}, lng_{pre}^{(j)} \rangle$ *are the map-matched latitude and longitude on* $e^{(j)}$.

Definition 5 (Map-matched ϵ-Sampling Rate Trajectory)**.** *A map-matched trajectory* $\tilde{\tau}$ *with* ϵ*-sampling rate is a sequence of map-matched trajectory points, i.e.,* $\tilde{\tau} = (a_1, a_2, \cdots, a_m)$, *where* $a_j = \langle e^{(j)}, r^{(j)}, t^{(j)}, lat_{pre}^{(j)}, lng_{pre}^{(j)} \rangle, \forall j, 1 \leq j \leq m$ *and* $a_{j+1}.t^{(j+1)} - a_j.t^{(j)} = \epsilon$. *For simplicity, we name* $\tilde{\tau}$ *as* ϵ*-MM trajectory ,with* ϵ *representing the sampling rate.*

Definition 6 (Trajectory Recovery)**.** *Given a low-sampling-rate trajectory* $\tau = (p_1, p_2, \cdots, p_n)$ *with corresponding map-matched road segment sequence* $\tau' = (M(p_1), M(p_2), ..., M(p_n))$ *and a target sampling rate* ϵ, *we aim to recover the real map-matched* ϵ *sampling-rate trajectory* $\tilde{\tau} = (a_1, a_2, \cdots, a_m)$. *This is to say, for each low-sampling-rate trajectory, we will infer its missing points and map match it onto the road network simultaneously.*

4 Methodology

In this section, we detail the components of the proposed framework GRFTrajRec for trajectory recovery, depicted in Figure 2. **(1)** We employ the Figure 2.(a) trajectory-aware graph representation and Figure 2.(b) spatiotemporal trajectory representation to derive an overall representation Z^{τ} for the low-sampling trajectory. **(2)** Using the attention-enhanced transformer encoder described in the Figure 2.(c) spatiotemporal interval-informed seq2seq model with Z^{τ}, we generate the initial GRU hidden cell $h_{gru}^{(0)}$. **(3)** Utilizing the feature differences-aware decoder in the Figure 2.(c) based on $h_{gru}^{(0)}$, we decode the high-sampling trajectory $\tilde{\tau}$ point by point.

4.1 Trajectory-Aware Graph Representation

In this subsection, we initially analyze the static road network topology to derive a geographical graph representation. Next, we introduce CandiGNN (Candidate Graph Neural Network) to acquire a candidate road network representation. This enables a more profound and nuanced comprehension of the intricate relationship between trajectories and roads by leveraging dynamic trajectory information. Lastly, we integrate the geographical representation with the dynamic candidate representation to obtain the ultimate trajectory-aware graph representation.

Geographical Representation: GeoGAT. The topology of the road network plays a crucial role in the formation of a trajectory. To capture the structural information, we first utilize Node2Vec [Grover and Leskovec, 2016] to obtain a geographical representation $\boldsymbol{g}_i$ for road segment v_i. Next, we feed $\boldsymbol{g}_i$ to a graph attention network (GAT [Veličković *et al.*, 2018]) step by step to obtain a smoothed road geographical representation $\tilde{\boldsymbol{g}}_i \in \mathbb{R}^{d_{hid}}$.

Candidate Representation: CandiGNN. While GeoGAT effectively captures the geographic characteristics of road segments, it has limitations in *extracting dynamic recovery information* from the original trajectory sequence. To address this limitation, we introduce road segment candidate probability (RCP) in Definition 3. This probability represents the likelihood that a road segment either passes through a trajectory point or is close to a trajectory point. The higher the RCP value for a road segment, the greater the probability that it will pass through a trajectory point, and subsequently, the greater the probability that it will be selected as the road segment where the missing points are located.

To extract the candidate representation of the graph effectively, we propose a novel graph neural network called CandiGNN. In its message-passing process, CandiGNN leverages candidate probability in measuring the correlations between candidate road segments. First, we take random vector $\boldsymbol{c}_i^{(0)} \in \mathbb{R}^{d_{in}}$ as the initialized candidate representation for node v_i. We use the difference in candidate probabilities to represent the information mobility between road segments, then the attention weight $\alpha_{ij}^{(l)}$ between nodes v_i and v_j in the l-th update are computed as:

$$\begin{aligned} \boldsymbol{D}_{ij}^{(l)} &= \left(RCP_j^\tau - RCP_i^\tau\right) * (\boldsymbol{c}_j^{(l)} - \boldsymbol{c}_i^{(l)}), \\ e_{ij}^{(l)} &= \mathrm{ReLU}\left(\boldsymbol{D}_{ij}^{(l)} \boldsymbol{W}_0^{(l)}\right) \boldsymbol{W}_1^{(l)}, \\ \alpha_{ij}^{(l)} &= \frac{\exp(\mathrm{LeakyReLU}(e_{ij}^{(l)}))}{\sum_{k \in \mathcal{N}_i} \exp(\mathrm{LeakyReLU}(e_{ik}^{(l)}))}, \end{aligned} \tag{1}$$

where $\boldsymbol{c}_i^{(l)}, \boldsymbol{c}_j^{(l)} \in \mathbb{R}^{d_{in}}$ represent the candidate representations for nodes v_i and v_j respectively during the l-th update of parameters, $\boldsymbol{W}_0^{(l)} \in \mathbb{R}^{d_{in} \times d_{in}}, \boldsymbol{W}_1^{(l)} \in \mathbb{R}^{d_{in} \times 1}$ are learnable parameters. The negative input slope of LeakyReLU is set to 0.2 [He *et al.*, 2015]. Then we obtain the road candidate representation $\tilde{\boldsymbol{c}}_i \in \mathbb{R}^{d_{hid}}$ for node v_i through combining the features of its neighbors using the attention weights α_{ij}^l as:

$$\begin{aligned} \boldsymbol{O}_i^{(l)} &= \boldsymbol{c}_i^{(l)} \boldsymbol{W}_2^{(l)} + \sum_{j \in N_i} \alpha_{ij}^{(l)} \boldsymbol{c}_j^{(l)} \boldsymbol{W}_3^{(l)}, \\ \tilde{\boldsymbol{c}}_i^{(l+1)} &= \left(\mathrm{ReLU}\left(\boldsymbol{O}_i^{(l)} \boldsymbol{W}_4^{(l)}\right)\right) \boldsymbol{W}_5^{(l)}, \end{aligned} \tag{2}$$

where $\boldsymbol{W}_2^{(l)}, \boldsymbol{W}_3^{(l)} \in \mathbb{R}^{d_{in} \times d_{mid}}, \boldsymbol{W}_4^{(l)} \in \mathbb{R}^{d_{mid} \times d_{mid}}$ and $\boldsymbol{W}_5^{(l)} \in \mathbb{R}^{d_{mid} \times d_{hid}}$ are the learnable parameters.

Graph Fusion Layer. We propose a co-attention fusion module to fuse the geographical representation and candidate representation to get the final trajectory-aware graph representation. Given a raw trajectory τ with corresponding map-matched road segment IDs src, for simplicity, its static geographical representation and dynamic candidate representation acquired by src are denoted by $\boldsymbol{G}^\tau = (\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_2, \ldots, \tilde{\boldsymbol{g}}_n)$ and $\boldsymbol{C}^\tau = (\tilde{\boldsymbol{c}}_1, \tilde{\boldsymbol{c}}_2, \ldots, \tilde{\boldsymbol{c}}_n)$, where $\boldsymbol{G}^\tau, \boldsymbol{C}^\tau \in \mathbb{R}^{n \times d_{hid}}$. We integrate $\boldsymbol{G}^\tau, \boldsymbol{C}^\tau$ to get final road segment representation. Specifically, we concatenate $\boldsymbol{G}^\tau$ and $\boldsymbol{C}^\tau$ to get $\boldsymbol{H}^\tau \in \mathbb{R}^{n \times 2 \times d_{hid}}$ and calculate the fusion attention:

$$\boldsymbol{H}^\tau \boldsymbol{W}_q^{(l)} \triangleq \boldsymbol{Q}_g^{(l)}, \quad \boldsymbol{H}^\tau \boldsymbol{W}_k^{(l)} \triangleq \boldsymbol{K}_g^{(l)}, \quad \boldsymbol{H}^\tau \boldsymbol{W}_v^{(l)} \triangleq \boldsymbol{V}_g^{(l)},$$

$$Attention(\boldsymbol{Q}_g^{(l)}, \boldsymbol{K}_g^{(l)}, \boldsymbol{V}_g^{(l)}) = Softmax\left(\frac{\boldsymbol{Q}_g^{(l)} {\boldsymbol{K}_g^{(l)}}^T}{\sqrt{d}}\right) \boldsymbol{V}_g^{(l)}, \tag{3}$$

where $\boldsymbol{W}_q^{(l)}, \boldsymbol{W}_k^{(l)}, \boldsymbol{W}_v^{(l)} \in \mathbb{R}^{d_{hid} \times d_{hid}}$ are learnable parameters. Denote the fusion result as $\tilde{\boldsymbol{H}}^\tau \in \mathbb{R}^{n \times 2 \times d_{hid}}$ and we obtain the final output $\boldsymbol{H}_{road}^\tau \in \mathbb{R}^{n \times d_{graph}}$, where $d_{graph} = 2d_{hid}$, using a feedforward network $FNN(\cdot)$ and batch normalization $Norm(\cdot)$ as does in [Vaswani *et al.*, 2017]:

$$\boldsymbol{H}_{road}^\tau = Reshape(Norm(FFN(\tilde{\boldsymbol{H}}^\tau) + \tilde{\boldsymbol{H}}^\tau)). \tag{4}$$

4.2 Spatiotemporal Trajectory Representation

In this subsection, building upon existing approaches relying on grid information or raw GPS points, integrating overall spatial sequences and relative positions in high-sampling trajectories, we present a spatiotemporal trajectory representation meticulously designed to capture overall and synergistic spatiotemporal information inherent in trajectory sequences.

Time-Aware Representation. To capture the temporal information of a trajectory τ, we first convert its timestamp sequence $(t_1, t_2, \cdots, t_n)$ into a sequence in seconds and normalize each timestamp to the range of $[0, 1]$ using min-max normalization. Denote $(t_{s_1}, t_{s_2}, \cdots, t_{s_n})$ as the standardized time sequence. Then, we obtain the time representation of the sequence $\boldsymbol{T}^\tau \in \mathbb{R}^{n \times d_{time}}$ via linear projection:

$$\boldsymbol{T}^\tau = (t_{s_1} \| \ldots \| t_{s_n}) \boldsymbol{W}_t^{(l)}, \tag{5}$$

where $\boldsymbol{W}_t^{(l)} \in \mathbb{R}^{1 \times d_{time}}$ are learnable parameters.

Spatial-Aware Representation. To enhance the extraction of spatial information from the raw sequence $\tau = (p_1, p_2, \cdots, p_n)$, we divide the entire road network area into grids identified by coordinates $\langle x_i, y_i \rangle$. In this way, each point p_i is located in a specific grid. We also define $tid_i = \left\lfloor \frac{t^{(i)} - t^{(0)}}{\epsilon} \right\rfloor$ as the index for the points in the target ϵ-MM trajectory $\tilde{\tau}$, where ϵ denotes the desired sampling rate and $t^{(i)}$ is the timestamp of i-th point in the low-sampling-rate trajectory τ. Consequently, each raw point p_i is represented as a triplet $\langle x_i, y_i, tid_i \rangle, \forall i, 1 \le i \le n$.

Initially, we retrieve the spatial grid representation $Grid_i \in \mathbb{R}^{1 \times d_{grid}}$ using the grid coordinates $\langle x_i, y_i \rangle$:

$$\begin{aligned} gid_i &= x_i + \max(x) \times (y_i - 1), \\ Grid_i &= Embedding(gid_i), \end{aligned} \tag{6}$$

where $\max(x)$ represents the maximum value among all x coordinates within the area, and $Embedding$ refers to the embedding technique in [Mikolov *et al.*, 2013]. Then we use tid_i to get the spatial positional encoding $POS_i \in \mathbb{R}^{1\times d_{pos}}$ using the following equation:

$$POS_i[2j] = \sin\left(tid_i/10000^{2j/d_{\text{pos}}}\right), \quad POS_i[2j+1] = \cos\left(tid_i/10000^{2j/d_{\text{pos}}}\right). \tag{7}$$

In conclusion, note $d_{sp} = d_{grid} + d_{pos} + 3$, the final spatial-aware representation $\boldsymbol{S}^\tau \in \mathbb{R}^{n\times d_{sp}}$, is obtained by:

$$\begin{aligned} \boldsymbol{S}_i^\tau &= Grid_i \| x_i \| y_i \| tid_i \| POS_i, \\ \boldsymbol{S}^\tau &= Concatenate(\boldsymbol{S}_i^\tau), \quad i = 1, 2, \cdots, n. \end{aligned} \tag{8}$$

We concatenate the spatial-aware representation $\boldsymbol{S}^\tau$ and temporal-aware representation $\boldsymbol{T}^\tau$ to obtain the spatiotemporal representation of trajectory τ, denoted by $\boldsymbol{H}_{traj}^\tau \in \mathbb{R}^{n\times d_{traj}}$, where $d_{traj} = d_{sp} + d_{time}$.

4.3 Spatiotemporal Interval-Informed Seq2Seq

In this subsection, we present a spatiotemporal interval-informed seq2seq model that combines an attention-enhanced transformer with a feature-aware decoder. This model leverages spatiotemporal intervals between points to enhance the extraction of contextual information between trajectory points, effectively integrating road and trajectory representations for improved performance.

Attention-Enhanced Transformer Encoder. For trajectory τ, we concatenate its graph representation $\boldsymbol{H}_{road}^\tau$, trajectory representation $\boldsymbol{H}_{traj}^\tau$, road features $RF \in \mathbb{R}^{n\times d_{RF}}$, road segment IDs $src \in \mathbb{R}^{n\times 1}$, and trajectory velocity $speed \in \mathbb{R}^{n\times 1}$, culminating in the formation of the full representation of the trajectory τ, i.e., $GST^\tau \in \mathbb{R}^{n\times(d_{graph}+d_{traj}+d_{RF}+2)}$. Then we apply a linear layer on GST^τ and add the position embedding [Vaswani *et al.*, 2017] to obtain the spatial-temporal input $\boldsymbol{Z}^\tau \in \mathbb{R}^{n\times d_{hid}}$ to the transformer encoder. Considering the time interval $\Delta tid = |tid_i - tid_j|$ and geographic distance interval $dis = \text{distance}(\langle lat^{(i)}, lng^{(i)}\rangle, \langle lat^{(j)}, lng^{(j)}\rangle)$ between samples in the trajectory, we propose an enhanced self-attention mechanism to obtain the contextual trajectory representation $\tilde{\boldsymbol{Z}}^\tau = \left(\tilde{\boldsymbol{Z}}_1^\tau, \tilde{\boldsymbol{Z}}_2^\tau, \ldots, \tilde{\boldsymbol{Z}}_n^\tau, \tilde{\boldsymbol{Z}}_i^\tau \in \mathbb{R}^{1\times d_{\text{hid}}}\right)$:

$$\begin{aligned} &\boldsymbol{Z}^\tau \boldsymbol{A}_Q^{(l)} \triangleq \boldsymbol{Q}^{(l)}, \quad \boldsymbol{Z}^\tau \boldsymbol{A}_K^{(l)} \triangleq \boldsymbol{K}^{(l)}, \quad \boldsymbol{Z}^\tau \boldsymbol{A}_v^{(l)} \triangleq \boldsymbol{V}^{(l)}, \\ &\tilde{\boldsymbol{Z}}^\tau = Attention(\boldsymbol{Q}^{(l)}, \boldsymbol{K}^{(l)}, \boldsymbol{V}^{(l)}) \\ &= Softmax\left(\frac{\boldsymbol{Q}^{(l)}\boldsymbol{K}^{(l)\top}}{\sqrt{d}} + \alpha f(\Delta\text{tid}) + \beta g(dis)\right)\boldsymbol{V}^{(l)}, \end{aligned}$$

where $\boldsymbol{A}_Q^{(l)}, \boldsymbol{A}_K^{(l)}, \boldsymbol{A}_V^{(l)} \in \mathbb{R}^{d_{hid}\times d'}$ are learnable parameters, $d' = d_{hid}/h_{num}$, h_{num} is the number of attention heads, α and β are hyperparameters, f and g are the corresponding probability transformation function.

Feature Differences-Aware Decoder. When decoding the j-th point a^j in $\tilde{\tau}$, the spatiotemporal interval between the $(j-1)$-th predicted map-matched trajectory point $a_{j-1} = \langle e^{(j-1)}, r^{(j-1)}, t^{(j-1)}, lat_{Pre}^{(j-1)}, lng_{Pre}^{(j-1)}\rangle$ and an input raw point $p_i = \langle lat^{(i)}, lng^{(i)}, t^{(i)}\rangle$ varies across all i in the range $1 \leq i \leq n$. To take this into account, we define a spatiotemporal feature difference vector $f_i \in \mathbb{R}^{1\times 3}$, which represents differences in sequence position, spatial location, and road segment candidate probability:

$$\begin{aligned} f_i = Concat(&\langle tid_i - (t-1)\rangle, \langle RCP_{M(p_i)}^\tau - RCP_{e^{(t-1)}}^\tau\rangle, \\ &distance(\langle lat^{(i)}, lng^{(i)}\rangle, \langle lat_{pre}^{(t-1)}, lng_{pre}^{(t-1)}\rangle)), \end{aligned} \tag{9}$$

where $RCP_{e^{(t-1)}}^\tau$ and $RCP_{M(p_i)}^\tau$ are the candidate probabilities for road segment $e^{(t-1)}$ and matched road segment $M(p_i)$ for p_i. We propose a differences-aware decoder that incorporates a GRU model [Cho *et al.*, 2014] with the feature difference vector f_i. In the GRU model, the hidden-state vector at timestamp j is denoted as $h_{gru}^{(j)}$, initially set to $h_{gru}^{(0)} = Mean(\tilde{\boldsymbol{Z}}^\tau) \in \mathbb{R}^{1\times d_{hid}}$. To capture the correlation between a_{j-1} and p_i, we introduce an attention mechanism $attn^{(j)} \in \mathbb{R}^{1\times d_{hid}}$ based on the feature difference f_i:

$$\begin{aligned} &u_i^{(j)} = \left(W_{tf}^{(l)T}\left(\tanh\left(h_{gru}^{(j-1)} \| \tilde{\boldsymbol{Z}}_i^\tau \| f_i\right) W_{attn}^{(l)}\right)\right), \\ &\alpha_i^{(j)} = \exp\left(\mu_i^{(j)}\right) / \sum_{i=1}^{n} \exp\left(\mu_i^{(j)}\right), \quad attn^{(j)} = \sum_{i=1}^{n} \alpha_i^{(j)} \tilde{\boldsymbol{Z}}_i^\tau, \end{aligned} \tag{10}$$

where $W_{tf}^{(l)} \in \mathbb{R}^{d_{hid}\times 1}$, $W_{attn}^{(l)} \in \mathbb{R}^{(2d_{hid}+3)\times d_{hid}}$ are learnable weights. The hidden-state vectors $h_{\text{gru}}^{(j)}$ is updated by:

$$h_{gru}^{(j)} = GRU\left(\left[p^{(j-1)} \middle\| r^{(j-1)} \middle\| RF^{(j-1)} \| attn^{(j)}\right]\right), \tag{11}$$

where $j \in [1, 2, \cdots, m]$, $p^{(j-1)} \in \mathbb{R}^{1\times d_{id}}$ is the embedding for the predicted road segment $e^{(j-1)}$, $r^{(j-1)} \in \mathbb{R}^{1\times 1}$ is moving ratio at the $(j-1)$-th timestamp, $RF^{(j-1)} \in \mathbb{R}^{1\times d_{RF}}$ is corresponding road features, $attn^{(j)}$ is the attention value.

4.4 Training

We employ a multi-task loss for training the model. Specifically, we consider predicting both the road segment ID and the moving ratio. In accordance with MTrajRec [Ren *et al.*, 2021], we employ the cross-entropy loss $\mathcal{L}_1(\theta)$ for the road segment ID prediction task and mean squared error (MSE) loss $\mathcal{L}_2(\theta)$ for the moving ratio prediction task:

$$\begin{aligned} &\mathcal{L}_1(\theta) = -\sum_{\tau,\tilde{\tau}} \sum_{j=1}^{m} \sum_{\ell=1}^{|\mathcal{V}|} 1\left\{a_j.e = e_\ell\right\} \log\left(P_\theta\left(\hat{a}_j.e = e_\ell \mid h_{\text{gru}}^{(j)}\right)\right) \\ &\mathcal{L}_2(\theta) = -\sum_{\tau',\tilde{\tau}} \sum_{j=1}^{m} \left(a_j \cdot r - R_\theta\left(\left[p^{(j)} \| h_{gru}^{(j)}\right]\right)\right)^2. \end{aligned} \tag{12}$$

where m is the length $\tilde{\tau}$, $|\mathcal{V}|$ is the size of road segments, $a_j.e$ is the ground truth of road segment ID, $\hat{a}_j$ is the prediction,$\mathcal{D}$ means the testing dataset consisting of low-sampling-rate trajectories and ϵ-MM trajectories, $a_j.r$ is the ground truth of the real moving ratio, $p^{(j)}$ represents the road segment embedding for the predicted road segment at the j-th timestamp,P_θ and R_θ represents the neural network for predicting road segments and the moving ratio. Overall, the final

Dataset	road network latitude range	road network longitude range	road segments	trajectories	sample ratio	Size of training area (km^2)
Porto	[41.121621,41.167815]	[-8.644531,-8.596830]	5121	67919	15	5.31*5.13
NanJing	[32.115211,32.215211]	[118.69454,118.84454]	3582	10267	15	11.49*16.13
YanCheng	[33.1696824,33.4401188]	[120.1070088,120.3560447]	7061	95420	15	27.72*29.99

Table 1: Dataset statistics.

Method	Porto (missing rate = 7/8)								YanCheng (missing rate = 7/8)							
	Accuracy ↑	Recall↑	Precision↑	F1 Score↑	MAE↓	RMSE↓	RN_MAE↓	RN_RMSE↓	Accuracy↑	Recall↑	Precision↑	F1 Score↑	MAE↓	RMSE↓	RN_MAE↓	RN_RMSE↓
HMM+linear	0.3045	0.4071	0.4195	0.4087	531.1619	1334.9815	649.3775	1407.1458	0.4119	0.4293	0.5745	0.4668	1040.6785	1916.1261	1354.5195	2223.4461
T2vecRec	0.5058	0.6047	0.7553	0.6716	81.2392	157.6399	794.2771	1223.3410	0.5591	0.5869	0.7307	0.6508	311.0731	997.0443	437.8637	1404.1595
ST2vecRec	0.5413	0.6423	0.7600	0.6961	66.0460	107.1436	85.3169	159.0711	0.5875	0.6107	0.7656	0.6793	194.7927	458.6835	289.1499	782.6099
TFTrajRec	0.5688	0.6602	0.7820	0.7159	59.0613	**93.4612**	671.4545	1120.5507	0.5928	0.6171	0.7578	0.6801	178.9251	382.5703	277.4743	802.5729
MTrajRec	0.5727	0.6625	0.7935	0.7221	60.2961	95.1038	667.9603	1117.0204	0.5931	0.6195	0.7648	0.6844	171.3210	**351.2522**	270.8849	771.2905
RNTrajRec	0.5584	0.6521	0.7737	0.7077	64.2907	103.0239	82.1912	152.7678	0.5813	0.6100	0.7663	0.6791	188.9081	405.8747	244.0889	436.1330
StartRec	0.5635	0.6554	0.7741	0.7098	60.5565	95.2924	77.4765	144.1988	0.5896	0.6122	0.7771	0.6847	178.4308	388.6332	235.6051	424.8397
GRFTrajRec	**0.6143**	**0.6993**	**0.8044**	**0.7481**	**55.7317**	99.6640	**70.1900**	**142.0160**	**0.6641**	**0.6596**	**0.8079**	**0.7261**	**153.1712**	396.9333	**195.7864**	**375.9991**

Method	Porto (missing rate = 15/16)								YanCheng (missing rate = 15/16)							
	Accuracy↑	Recall↑	Precision↑	F1 Score↑	MAE↓	RMSE↓	RN_MAE↓	RN_RMSE↓	Accuracy↑	Recall↑	Precision↑	F1 Score↑	MAE↓	RMSE↓	RN_MAE↓	RN_RMSE↓
HMM+linear	0.1857	0.2882	0.2983	0.2883	714.1581	1554.1832	879.1707	1641.2111	0.2924	0.3328	0.4608	0.3616	1611.9585	2469.8164	1924.9359	2733.0449
T2vecRec	0.4727	0.5816	0.7434	0.6526	90.4906	170.6969	853.8686	1271.1728	0.5366	0.5585	0.7422	0.6372	362.9485	1081.0936	876.3498	1283.9178
ST2vecRec	0.5110	0.6152	0.7464	0.6745	74.0351	119.8142	96.0490	176.2229	0.5746	0.5883	0.7775	0.6697	202.1533	415.6524	303.6573	831.5196
TFTrajRec	0.5322	0.6348	0.7614	0.6923	67.1900	109.0267	736.9071	1176.2382	0.5828	0.6094	0.7762	0.6826	182.2594	**354.7263**	779.6341	1205.4420
MTrajRec	0.5326	0.6241	**0.7804**	0.6935	70.4774	116.2203	742.8772	1181.4727	0.5890	**0.6097**	0.7797	0.6841	180.1026	359.1027	769.6789	1197.5423
RNTrajRec	0.5140	0.6163	0.7578	0.6797	73.7399	117.6378	94.7133	171.6858	0.5750	0.6011	0.7759	0.6772	198.6109	418.9837	254.3493	451.0068
StartRec	0.5263	0.6291	0.7497	0.6841	68.6396	109.1803	88.8423	162.4110	0.5785	0.6003	0.7779	0.6775	193.8829	389.5013	252.9964	439.1763
GRFTrajRec	**0.5614**	**0.6572**	0.7763	**0.7118**	**63.6366**	**108.3740**	**81.5393**	**158.1223**	**0.6090**	0.6080	**0.7847**	**0.6850**	**185.5892**	442.4354	**235.8697**	**433.7974**

Table 2: Performance evaluation for different methods in trajectory recovery (random missing).

objective function is a weighted sum of the two functions:

$$\mathcal{L}_t = \mathcal{L}_1(\theta) + \lambda \mathcal{L}_2(\theta). \tag{13}$$

5 Evaluation

5.1 Experimental Setup

Datasets. We validate the effectiveness of our model on three real-world trajectory datasets along with their corresponding road networks sourced from OpenStreetMap[2]. The Porto dataset[3] are openly available, while the YanCheng and NanJing datasets are provided by a company partner. Detailed statistics of these datasets are presented in Table 1. For each dataset, we divided the data into training, validation, and testing sets, following a 7:2:1 split ratio. To obtain low-sample trajectories, we set three distinct data missing types: random missing (where the missing trajectory points are scattered randomly), uniform missing (where the missing points are uniformly distributed across the entire trajectory), and block missing (where the missing points form a continuous sub-trajectory segment within the trajectory). Additionally, our experiments adopt two missing rates: 7/8 or 15/16 in line with the rates used in RNTrajRec [Chen *et al.*, 2023].

Baselines. To evaluate the effectiveness of GRFTrajRec, we implement in total seven baselines. **i)** HMM + Linear [Hoteit *et al.*, 2014] uses linear interpolation to obtain a high sample trajectory, and then uses HMM algorithm to obtain a map-matched ϵ-sampling rate trajectory. **ii)** t2vec [Li *et al.*, 2018] proposes a deep learning network for trajectory similarity learning with a BiLSTM [Graves and Graves, 2012] model. **iii)** ST2vec [Fang *et al.*, 2022] encodes spatial and temporal information of trajectories. **iv)** Transformer [Vaswani *et al.*, 2017] learns the representation with temporal dependency. **v)** MTrajRec [Ren *et al.*, 2021] solve the map-constrained trajectory recovery problem via seq2seq multi-task learning. **vi)** RNTrajRec [Chen *et al.*, 2023] is the state-of-the-art method for trajectory recovery task. **vii)** Start [Jiang *et al.*, 2023] is the state-of-the-art method for trajectory representation in spatial networks which is applicable for many trajectory-based downstream tasks.

Parameter Settings. We implemented all the baseline models on a machine equipped with an Intel(R) Xeon(R) 32-core CPU E5-2620 v4 @ 2.10GHz and a 12GB NVIDIA TITAN V GPU. All models were trained using the Adam optimizer for at most 50 epochs, with a batch size of 128 and a learning rate of 1e-3. For the Porto dataset, we configured the hidden-state size hyperparameter d to be 512. For the YanCheng and NanJing datasets, we set it to 128. We explore (alpha $\in$ {1,10,100,1000}, beta $\in$ {1,10,100,1000}, lambda $\in$ {10,20,30,40,50}). The optimal model parameters are Porto (100, 100, 10), YanCheng (1, 1, 10) and NanJing (1, 1, 10) while the worst model parameters are Porto (1, 1, 20), YanCheng (1000, 1000, 20) and NanJing (1000,1000,20).

Metrics. Following RNTrajRec [Chen *et al.*, 2023], we adopt the $accuracy$, $recall$, $precision$, $f1score$ of the road segments recovered and the RN_MAE,RN_RMSE of distance on road network, as well as MAE and $RMSE$ of the euclidean distance between predicted and actual points to evaluate the performances of different models.

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

[3]https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data

Method	Porto (missing rate = 7/8)								YanCheng (missing rate = 7/8)							
	Accuracy↑	Recall↑	Precision↑	F1 Score↑	MAE↓	RMSE↓	RN_MAE↓	RN_RMSE↓	Accuracy↑	Recall↑	Precision↑	F1 Score↑	MAE↓	RMSE↓	RN_MAE↓	RN_RMSE↓
GRF-ATF	0.6033	0.6904	0.8030	0.7425	58.6858	104.4425	73.0209	144.6467	0.6281	0.6321	0.7791	0.6979	180.6887	461.0387	225.0187	421.0963
GRF-FD	0.5726	0.6721	0.7929	0.7275	61.1597	99.7132	76.1045	141.7228	0.6193	0.6270	0.7892	0.6987	163.9825	370.1132	214.0067	393.5901
GRF-GR	0.5866	0.6752	0.7878	0.7272	58.1021	96.9891	75.0917	147.3024	0.6257	0.6345	0.7720	0.6964	171.0569	422.4033	221.6208	423.0005
GRF-CG+	0.5713	0.6700	0.7803	0.7209	59.5166	96.8129	76.0747	143.8126	0.6480	0.6425	0.7983	0.7118	**150.2455**	**364.1130**	196.8299	**371.7215**
GRF-TR	0.6047	0.6900	0.7961	0.7393	56.6704	**95.1261**	70.0992	**134.4768**	0.6457	0.6505	0.7965	0.7160	162.0133	389.6058	211.7423	397.5213
GRFTrajRec	**0.6143**	**0.6993**	**0.8044**	**0.7481**	**55.7317**	99.6640	**70.1900**	142.0160	**0.6641**	**0.6596**	**0.8079**	**0.7261**	153.1712	396.9333	**195.7864**	375.9991

Table 3: Performance evaluation of ablation experiments (random missing).

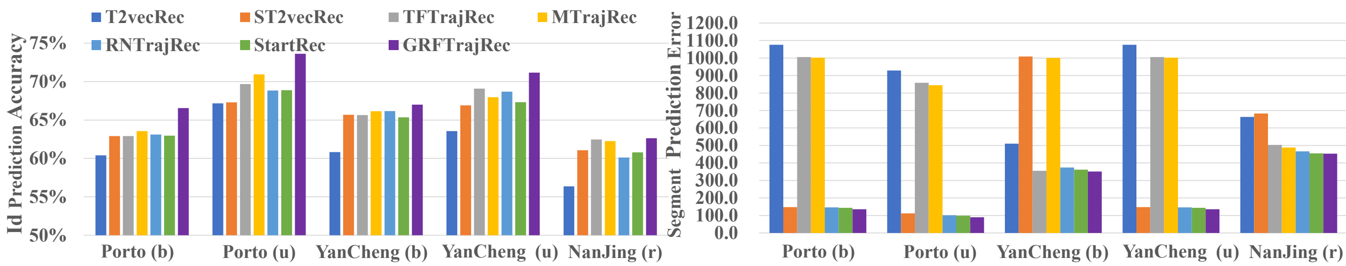


Figure 3: Further performance results for different methods (b: block missing, u: uniform missing, r: random missing, missing rate = 7/8).

5.2 Overall Performance

We conducted a comprehensive performance evaluation in the scenario of random missing data using the Porto and YanCheng datasets, as summarized in Table 2. The figures highlighted in bold signify the top-performing models. Notably, **GRFrajRec** consistently outperforms the baseline models across all datasets, demonstrating significant advantages across most metrics. Meanwhile, Linear+HMM consistently underperforms across all datasets. Notably, both TFrajRec and MTrajRec demonstrate superior performance on ID prediction metrics ($accuracy, f1score$), showcasing their ability to capture spatiotemporal information in low-sample trajectories. However, they fall short in segment prediction error metrics (RN_MAE,RN_RMSE) due to their lack of consideration for the road network's topological aspects. In contrast, RNTrajRec and StartRec incorporate the topological characteristics of the road network, resulting in improved performance in segment prediction error metrics. Remarkably, their ID prediction performance is on par with the previous two models. Our framework, **GRFrajRec**, emerges as the standout performer, excelling in both ID prediction metrics and road network segment prediction error metrics.

5.3 Further Experiments

We extend our experimentation to include the **Nanjing dataset** and introduce two additional missing types: uniform missing and block missing. This broader scope allows us to assess the generalizability and robustness of our model across various scenarios and datasets with differing distributions and complexities. The results, as depicted in Figure 3 alongside Table 2, consistently reaffirm the superior performance of **GRFTrajRec** across all missing types and datasets. This robust performance underscores the adaptability, versatility and reliability of GRFTrajRec in diverse real-world settings. These findings bolster confidence in the effectiveness of GRFTrajRec and highlight its potential for real-world applications across different contexts and real data distributions.

5.4 Ablation Study

To further verify the effectiveness of different modules in our model, we have devised five variants of **GRFTrajRec**: i) **GRF-ATF** removes the spatiotemporal interval consideration in the attention-enhanced transformer encoder and uses the traditional transformer. ii) **GRF-FD** removes the spatiotemporal interval consideration in the feature differences-aware decoder and uses the traditional GRU. iii) **GRF-GR** removes the trajectory-aware graph representation. iv) **GRF-CG+** removes CandiGNN but add the traditional GCN [Kipf and Welling, 2017] to the trajectory-aware graph representation. v) **GRF-TR** removes the spatiotemporal trajectory representation. The experimental results are detailed in Table 3. It's evident from the data that **GRFTrajRec** consistently surpasses all its variants across the majority of settings, highlighting the importance and effectiveness of these modules.

As discussed in subsection 4.1, the importance of trajectory-aware graph representation (GR) becomes evident in extracting advanced road embeddings. Removing GR from **GRFTrajRec** leads to a significant decline in overall performance. Notably, **GRFTrajRec** consistently outperforms **GRF-CG+**, only when replacing CandiGNN with the traditional GCN, highlighting the necessity of a dedicated graph neural network for trajectory recovery. Moreover, the omission of the spatiotemporal interval consideration in the feature differences-aware decoder results in a notable drop of over 0.4 across all datasets. This highlights the effectiveness of considering spatiotemporal interval in the decoder, reaffirming its importance in enhancing recovery accuracy. Across various metrics, **GRFTrajRec** consistently outperforms both **GRF-ATF** and **GRF-TR**, demonstrating the contributions of the attention-enhanced transformer and spatiotemporal representation learning in improving recovery accuracy. Overall, the trajectory-aware graph representation and the feature differences-aware decoder emerge as the two most crucial components within **GRFTrajRec**, underscoring their indispensable roles in achieving superior recovery performance.

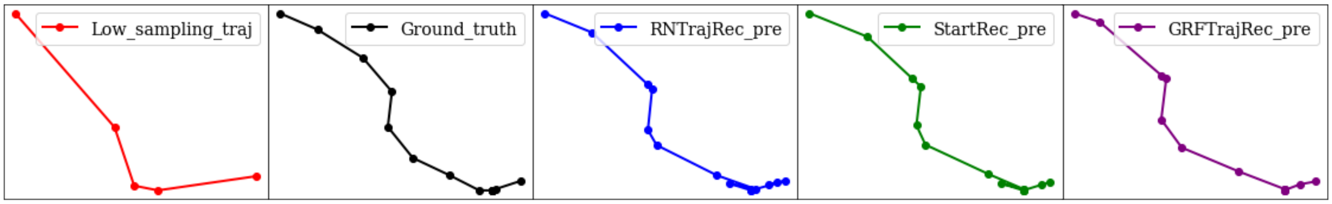


Figure 4: A case study for trajectory recovery on Porto dataset with missing rate 7/8 (horizontal coordinates: latitude, vertical coordinates: longitude). GRFTrajRec is our model, and StartRec is the current state-of-the-art model, and RNTrajRec is the other top-performing model.

5.5 Case Study

We conduct the visualization comparison experiment on the trajectory recovery task. The input is a low-sampling trajectory, while the ground truth is the actual trajectory at a higher sampling rate. We provide the trajectory recovery results of our model, GRFTrajRec, as well as two of the current best-performing models, RNTrajRec and StartRec.

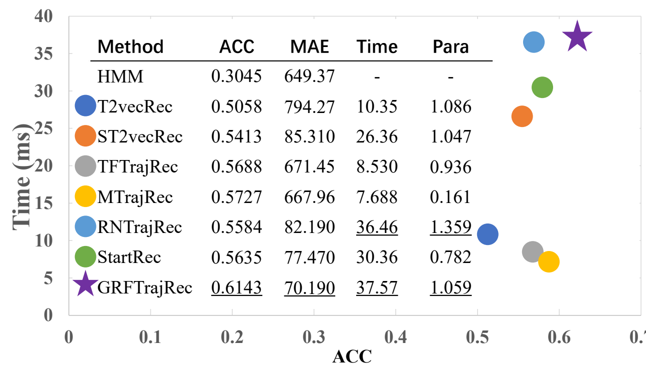
As illustrated in Figure 4, while all three models can recover the approximate high-sampling trajectory, our model's trajectory recovery closely aligns with the ground truth. In contrast, RNTrajRec and StartRec exhibit noticeable deviations from the ground truth, particularly at the trajectories' ends. Our model's precision in achieving seamless recovery from start to finish is bolstered by its holistic consideration of the interaction between road networks and trajectories, along with the spatiotemporal intervals between trajectory points. The dynamic interaction effectively incorporates missing information into the representation of low-sampling trajectories. Additionally, the spatiotemporal intervals between points facilitate a nuanced understanding of the specific spatiotemporal relationships among missing points and other data points. This comprehensive analysis enhances our ability to accurately pinpoint the spatiotemporal positions of each missing point, reducing overall recovery errors and notably enhancing trajectory recovery accuracy.

5.6 Efficiency Study

In addition to evaluating method effectiveness, we also assess efficiency from two perspectives: inference time for recovering a trajectory and the number of model parameters.

As shown in Figure 5, when compared to RNTrajRec and StartRec, which share similar inference times and model parameters, **GRFTrajRec** demonstrates superior performance in both accuracy (ACC) and mean absolute error (MAE) metrics. Many other models, despite having relatively fewer parameters or shorter runtimes, often compromise accuracy (ACC) or substantially increase mean absolute error (MAE).

Our model's longer inference time is due to the consideration of spatiotemporal intervals in seq2seq modelling, which significantly improves prediction accuracy (ACC) and reduces prediction error (MAE), as revealed in our ablation experiments in subsection 5.4. Thus, we believe that the little increase in the inference time is a worthwhile trade-off, considering it leads to enhanced ACC and reduced MAE.



Method	ACC	MAE	Time	Para
HMM	0.3045	649.37	-	-
T2vecRec	0.5058	794.27	10.35	1.086
ST2vecRec	0.5413	85.310	26.36	1.047
TFTrajRec	0.5688	671.45	8.530	0.936
MTrajRec	0.5727	667.96	7.688	0.161
RNTrajRec	0.5584	82.190	36.46	1.359
StartRec	0.5635	77.470	30.36	0.782
GRFTrajRec	0.6143	70.190	37.57	1.059

Figure 5: Efficiency analysis on Porto (random missing rate = 7/8).

5.7 Assessment of Trajectory Recovery Necessity

Assessing the necessity of trajectory recovery is paramount across various tasks, scenarios and different data densities, to effectively optimize predictive models based on trajectory data and ultimately save costs. For instance, by evaluating the impact of trajectory data of varying densities on model performance, we can determine when trajectory recovery is advantageous. We predict driver IDs using trajectory representation from the model in subsection 4.2. We employ 10,000 trajectories (sample one point per second), adjusting the missing rate to control trajectory data density. We observe a significant drop in model performance when the missing rate exceeds 0.75 (sample 0.25 points per second). **(1)** In scenarios with several points sampled per second below 0.25 due to data collection limitations, indicating excessively sparse sampling, trajectory recovery may enhance model performance. **(2)** Conversely, if the number of points sampled per second is above 0.25, and the data density is relatively high, model optimization takes precedence over trajectory recovery.

6 Discussion and Conclusion

In this paper, we introduce a new graph-based framework for trajectory recovery with spatiotemporal interval-informed seq2seq, GRFTrajRec. Experiments demonstrate that GRFTrajRec outperforms current methods by better integrating trajectory-road network interactions and considering spatiotemporal intervals. For future work, we aim to focus on advancing trajectory-network interaction analysis beyond only using road segment candidate probability modelling and explore calculating spatiotemporal intervals for selected points rather than all to speed up the computational speed.

Acknowledgments

This work is supported by the MOE Project of Key Research Institute of Humanities and Social Sciences (No.22JJD110001), the National Natural Science Foundation of China (No.72171229) and the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China. This work is also partly supported by the Marsden Fund Council from Government funding (MFP-UOA2123) administered by the Royal Society of New Zealand, and NSF IIS award #2153369.

References

[Chen *et al.*, 2023] Y. Chen, H. Zhang, W. Sun, and B. Zheng. Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 829–842, Los Alamitos, CA, USA, apr 2023. IEEE Computer Society.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Fang *et al.*, 2022] Ziquan Fang, Yuntao Du, Xinjun Zhu, Danlei Hu, Lu Chen, Yunjun Gao, and Christian S Jensen. Spatio-temporal trajectory similarity learning in road networks. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 347–356, 2022.

[Fu and Lee, 2020] Tao-Yang Fu and Wang-Chien Lee. Trembr: Exploring road networks for trajectory representation learning. *ACM Trans. Intell. Syst. Technol.*, 11(1), feb 2020.

[Graves and Graves, 2012] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Cornell University - arXiv,Cornell University - arXiv*, Jul 2016.

[Hamilton *et al.*, 2017] WilliamL. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Neural Information Processing Systems,Neural Information Processing Systems*, Jun 2017.

[Han *et al.*, 2021] Peng Han, Jin Wang, Di Yao, Shuo Shang, and Xiangliang Zhang. A graph-based approach for trajectory similarity computation in spatial networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 556–564, 2021.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[Hoteit *et al.*, 2014] Sahar Hoteit, Stefano Secci, Stanislav Sobolevsky, Carlo Ratti, and Guy Pujolle. Estimating human trajectories and hotspots through mobile phone data. *Computer Networks*, page 296–307, May 2014.

[Jarboui and Perchet, 2021] Firas Jarboui and Vianney Perchet. Trajectory representation learning for multi-task nmrdp planning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6786–6793, 2021.

[Jiang *et al.*, 2023] Jiawei Jiang, Dayan Pan, Houxing Ren, Xiaohan Jiang, Chao Li, and Jingyuan Wang. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *2023 IEEE 39th international conference on data engineering (ICDE)*, pages 843–855. IEEE, 2023.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[Lan *et al.*, 2022] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, pages 11906–11917. PMLR, 2022.

[Li and Zhu, 2021] Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4189–4196, 2021.

[Li *et al.*, 2018] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. Deep representation learning for trajectory similarity computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, Apr 2018.

[Li *et al.*, 2023] Jiajia Li, Mingshen Wang, Lei Li, Kexuan Xin, Wen Hua, and Xiaofang Zhou. Trajectory representation learning based on road network partition for similarity computation. In *International Conference on Database Systems for Advanced Applications*, pages 396–413. Springer, 2023.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, GregS. Corrado, and J.Michael Dean. Efficient estimation of word representations in vector space. *International Conference on Learning Representations,International Conference on Learning Representations*, Jan 2013.

[Newson and Krumm, 2009] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Nov 2009.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[Ren *et al.*, 2021] Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, Aug 2021.

[Si *et al.*, 2023] Junjun Si, Jin Yang, Yang Xiang, Hanqiu Wang, Li Li, Rongqing Zhang, Bo Tu, and Xiangqun Chen. Trajbert: Bert-based trajectory recovery with spatial-temporal refinement for implicit sparse trajectories. *IEEE Transactions on Mobile Computing*, pages 1–12, 2023.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2020] Jingyuan Wang, Ning Wu, Xinxi Lu, Xin Zhao, and Kai Feng. Deep trajectory recovery with fine-grained calibration using kalman filter. *IEEE Transactions on Knowledge and Data Engineering*, page 1–1, Jan 2020.

[Wang *et al.*, 2022] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. When will you arrive? estimating travel time based on deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Jun 2022.

[Xia *et al.*, 2022] Tong Xia, Yunhan Qi, Jie Feng, Fengli Xu, Funing Sun, Diansheng Guo, and Yong Li. Attnmove: History enhanced trajectory recovery via attentional network. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 4494–4502, Sep 2022.

[Yang *et al.*, 2021] Peilun Yang, Hanchen Wang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. T3s: Effective representation learning for trajectory similarity computation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Apr 2021.

[Yao *et al.*, 2022] Di Yao, Haonan Hu, Lun Du, Gao Cong, Shi Han, and Jingping Bi. Trajgat: A graph-based long-term dependency modeling approach for trajectory similarity computation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2275–2285. ACM, 2022.

[Zhang and Zhao, 2021] Zheng Zhang and Liang Zhao. Representation learning on spatial networks. *Advances in Neural Information Processing Systems*, 34:2303–2318, 2021.

[Zhang *et al.*, 2018] Hanyuan Zhang, Hao Wu, Weiwei Sun, and Baihua Zheng. Deeptravel: A neural network based travel time estimation model with auxiliary supervision. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 3655–3661. AAAI Press, 2018.

[Zhang *et al.*, 2022] Xinrui Zhang, Xingyuan Liang, Hai Wang, Shuai Wang, and Tian He. Patr: Periodicity-aware trajectory recovery for express system via seq2seq model. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 486–491, 2022.