Bridging Traffic State and Trajectory for Dynamic Road Network and Trajectory Representation Learning

Chengkai Han[1], **Jingyuan Wang**[1,2,3*], **Yongyao Wang**[1], **Xie Yu**[1], **Hao Lin**[2,3], **Chao Li**[1,4], **Junjie Wu**[2,3]

[1] School of Computer Science and Engineering, Beihang University, Beijing, China
[2] MIIT Key Laboratory of Data Intelligence and Management, Beihang University, Beijing, China
[3] School of Economics and Management, Beihang University, Beijing, China
[4] Shenzhen Institute of Beihang University, Shenzhen, China

Abstract

Effective urban traffic management is vital for sustainable city development, relying on intelligent systems with machine learning tasks such as traffic flow prediction and travel time estimation. Traditional approaches usually focus on static road network and trajectory representation learning, and overlook the dynamic nature of traffic states and trajectories, which is crucial for downstream tasks. To address this gap, we propose TRACK, a novel framework to bridge traffic state and trajectory data for dynamic road network and trajectory representation learning. TRACK leverages graph attention networks (GAT) to encode static and spatial road segment features, and introduces a transformer-based model for trajectory representation learning. By incorporating transition probabilities from trajectory data into GAT attention weights, TRACK captures dynamic spatial features of road segments. Meanwhile, TRACK designs a traffic transformer encoder to capture the spatial-temporal dynamics of road segments from traffic state data. To further enhance dynamic representations, TRACK proposes a co-attentional transformer encoder and a trajectory-traffic state matching task. Extensive experiments on real-life urban traffic datasets demonstrate the superiority of TRACK over state-of-the-art baselines. Case studies confirm TRACK's ability to capture spatial-temporal dynamics effectively.

Code — https://github.com/NickHan-cs/TRACK

Introduction

Intelligent urban traffic management, such as traffic flow prediction (Bai et al. 2020; Wang et al. 2022; Ji et al. 2022a; Liu et al. 2024), travel time estimation (Wang et al. 2018; Wu et al. 2019a) and trajectory analysis (Chen et al. 2017; Ding et al. 2018; Chen et al. 2019), plays a crucial role in ensuring efficient city functioning and promoting sustainable development (Wang et al. 2021; Jiang et al. 2023b). In the realm of intelligent urban traffic management, traffic state data and trajectory data are two core components that can encapsulate the macroscopic and microscopic characteristics of cities, respectively, and their representation learning, *i.e.,* learning generic low-dimensional road segment and trajectory vectors, serve as two fundamental pillars for various urban traffic tasks (Chen et al. 2018; Wang et al. 2019a; Jiang et al. 2023d).

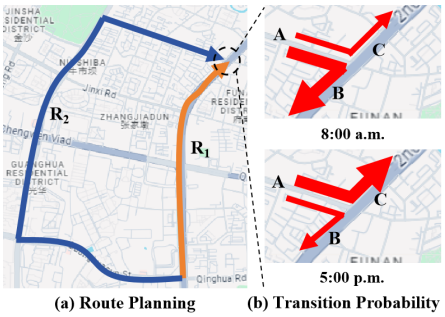

Figure 1: An example of mutual influences between traffic state data and trajectory data.

Recently, many efforts have been devoted to modeling traffic state data (Jiang et al. 2023a; Zou et al. 2024; Yi et al. 2024; Ji et al. 2022b, 2023, 2020) and trajectory data (Yang et al. 2021; Fu and Lee 2020; Jiang et al. 2023c). However, existing methods typically model these two types of data independently, lacking approaches capable of jointly modeling them. In urban transportation scenarios, traffic state data describes the dynamic macroscopic characteristics of groups on the road network, while trajectory data reflects the dynamic movement attributes of individuals on the road network. There are spatio-temporal correlations and mutual influences between traffic state data and trajectory data.

Firstly, traffic states influence individuals' choices of the trajectory route. For example, as shown in Figure 1(a), people tend to choose the shortest route R_1 during non-peak hours, but it might be a better choice to detour with a more time-saving route R_2 during peak hours (*i.e.,* when the average traffic speed of R_1 is low). Traffic states naturally affect the travel time of each road segment in a trajectory. Secondly, individual transitions on the road network are the direct cause of changes in traffic states. As shown in Figure (b), the traffic flow on segment C consists of the traffic flow entering from segment A and that entering from segment B. Therefore, the transition probability from segment A to segment C varies over different time periods, directly impacting the traffic states on segment C. Therefore, jointly modeling traffic state data and trajectory data can enrich the spatio-

*Corresponding author: jywang@buaa.edu.cn

temporal information learned by the model, capturing the dynamic nature of traffic states and trajectories.

To achieve this, we propose to bridge two prominent types of dynamic data, *i.e.,* TRAffic state and trajectory, for dynamiC road networK and trajectory representation learning (TRACK). Specifically, TRACK encodes the static and spatial features of road segments with graph attention networks (GAT) to learn road segment representations, followed by a trajectory transformer encoder with the masked trajectory prediction task and the contrastive trajectory learning task to learn trajectory representations. To capture the dynamic spatial features of road segments, we incorporate the transition probabilities computed from the trajectory data into the attention weights of GAT. Meanwhile, to capture the spatial-temporal dynamics of road segments from traffic state data, we learn a traffic transformer encoder with the mask state prediction task and the next state prediction task. More importantly, to capture the interactions of traffic state and trajectory data in characterizing a road segment's dynamic features, we model the information exchange between different data views by designing a co-attentional transformer encoder with a novel gravitivity-based attention mechanism and the trajectory-traffic state matching task. Finally, we pre-train the whole model with a joint self-supervised learning loss.

We conduct extensive experiments on two real-life urban traffic datasets and compare our proposed TRACK method with several state-of-the-art baselines. Evaluated with two downstream tasks, TRACK achieves consistently superior performances over baselines. Case studies also validate that TRACK can capture the spatial-temporal dynamics of road segments and trajectories through learned dynamic representations.

Preliminaries

In this section, we introduce the mathematical notations used throughout the paper and formally define our research problem.

Basic Elements of Urban Traffic

We start by introducing the basic spatial-temporal units of urban traffic, *i.e.,* a road segment and a time slice.

Definition 1 (**Road Segment**). *A road segment $v \in \mathcal{V}$ is the minimum spatial unit in urban traffic scenarios where $\mathcal{V}$ is the set of road segments.*

Definition 2 (**Time Slice**). *A time slice t is the minimum time unit (*e.g., *an hour) in urban traffic scenarios.*

For convenience, we might call *segment* for short in unambiguous cases. Next, we characterize the **spatial**, **static**, **dynamic** features of road segments and the **trajectory** with the following concepts.

Definition 3 (**Road Network**). *A road network is characterized as a graph $\mathcal{G} = (\mathcal{V}, \boldsymbol{A})$, where $\mathcal{V} = \{v_1, \cdots, v_N\}$ is a node set of N road segments and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is an adjacency matrix to capture the link information between N road segments. The road network entails the spatial features of road segments.*

Definition 4 (**Static Feature of Road Segment**). *The static feature $\boldsymbol{f}_v \in \mathbb{R}^{C_1}$ for a road segment v is a feature vector with which v is generally associated and does not change over time after it was built. C_1 is the dimension of the feature vector. For example, $C_1 = 5$ if the features include longitude, latitude, segment type, length, and speed limit.*

Definition 5 (**Traffic State Sequence**). *A traffic state sequence $\mathcal{S}_t \in \mathbb{R}^{T \times N \times C_2}$ is composed of T consecutive historical traffic states before the time slice t, where $\mathcal{S}_t = (\boldsymbol{TS}_{t-T}, \cdots, \boldsymbol{TS}_{t-1})$. $\boldsymbol{TS}_t$ denotes a traffic state at the time slice t. A traffic state $\boldsymbol{TS}_t \in \mathbb{R}^{N \times C_2}$ is the statistics (*e.g., *flow, density, average speed) on N road segments within the time slice t, where C_2 is the dimension of the statistics. The traffic state sequence involves the dynamic features of road segments that will change over time.*

Definition 6 (**Trajectory**). *A trajectory $\mathcal{T} = [\langle v_i, t_i \rangle]_{i=1}^{m}$ is a sequence of spatial-temporal points that record the movement behavior of a car or person in the scope of the road network $\mathcal{G}$, where m is the total number of spatial-temporal points, $v_i \in \mathcal{V}$ denotes the segment for the i-th visit, and $t_i \in \mathbb{R}$ denotes the corresponding visit timestamp.*

We use $\boldsymbol{F}_{\mathcal{V}} \in \mathbb{R}^{N \times C_1}$ to denote the static feature matrix of N road segments in the road network $\mathcal{G}$. We assume that there is a set of k_D trajectories within the time slice t, denoted as a trajectory set $\mathcal{D}_t = \{\mathcal{T}_j\}_{j=1}^{k_D}$, indicating that the departure timestamp of each trajectory $\mathcal{T}_j \in \mathcal{D}_t$ lies in the time slice t.

Problem Formulation

We formulate two representation learning tasks in urban traffic scenarios, *i.e., Dynamic Road Network Representation Learning (DRNRL)* and *Trajectory Representation Learning (TRL).*

Definition 7 (**Dynamic Road Network Representation Learning**). *Given the road network $\mathcal{G}$, the historical traffic state sequence $\mathcal{S}_t$ and the trajectory set $\mathcal{D}_t$ at the time slice t, DRNRL aims to derive a generic d_s-dimensional representation $\boldsymbol{h}_{v,t} \in \mathbb{R}^{d_s}$ at the time slice t for each road segment $v \in \mathcal{V}$ on the road network.*

Definition 8 (**Trajectory Representation Learning**). *Given the road network $\mathcal{G}$, the historical traffic state sequence $\mathcal{S}_t$ and the trajectory set $\mathcal{D}_t$ at the time slice t, TRL aims to derive a generic d_t-dimensional representation $\boldsymbol{l}_{\mathcal{T}} \in \mathbb{R}^{d_t}$ for each trajectory $\mathcal{T} \in \mathcal{D}_t$.*

Traditional representation learning methods on road network (Wu et al. 2020a) usually focus on learning a road segment's representation that does not change over time. However, in great contrast, *DRNRL* aims to learn dynamic representations of road segments by considering the dynamic features derived from traffic state and trajectory data. We interchangeably use the terms *road network representation* and *road segment representation* hereinafter. We assume that the numbers of latent dimensions for *DRNRL* and *TRL* are set to the same value d in our problem, *i.e.,* $d = d_s = d_t$. The learned road segment representation can be applied to various segment-related downstream tasks such as traffic state prediction and on-demand service prediction. The

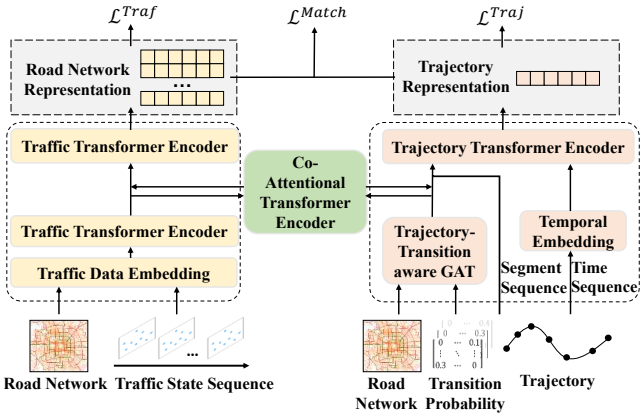

Figure 2: The overall architecture of the TRACK model.

learned trajectory representation can be applied to various trajectory-related downstream tasks such as travel time estimation and anomalous trajectory detection.

Methodology

In this section, we present the proposed *TRACK* model. Our core idea is to incorporate dynamic information from traffic state and trajectory data into road network and trajectory representation learning, and model the information exchange between multi-view data to enhance the dynamic representations. The overall architecture of the proposed model is shown in Figure 2.

Basic Pipeline of TRL

In this part, we introduce a basic pipeline for TRL, which first encodes the segments appearing in a trajectory into low-dimensional vectors and then combines them with the timestamp representations to derive the trajectory's final representation vector. We train it with the Masked Trajectory Prediction (MTP) task and the Contrastive Trajectory Learning (CTL) task.

Encoding Road Segment's Static and Spatial Features. We aim to project each segment v in the road network $\mathcal{G}$ into a low-dimensional representation vector $\boldsymbol{h}_v^{\mathcal{G}} \in \mathbb{R}^d$. The static features of each segment, *e.g.,* length and speed limits, contain rich semantics of the segment. Moreover, the connectivity of the segments in the road network, *i.e.,* the local network structure, also entails the spatial semantics of a segment. To this end, it is natural to learn the representation vector of a segment from both its static features and the local network structure. We adopt a GNN method, *i.e.,* a multi-layer Graph Attention Network (GAT)(Velickovic et al. 2018), to model each segment's static and spatial features as follows:

$$\boldsymbol{H}^{Traj} = \mathrm{GAT}(\boldsymbol{F}_{\mathcal{V}}, \boldsymbol{A}), \tag{1}$$

where $\mathrm{GAT}(\cdot,\cdot)$ denotes the implementation of a standardized GAT or a GAT optimized with sparse matrix operations, and $\boldsymbol{H}^{Traj} = [\boldsymbol{h}_v^{Traj}]_{v=1}^N \in \mathbb{R}^{N\times d}$ is a matrix form of N segments's representations in $\mathcal{G}$.

Encoding Timestamp Information. We introduce a temporal embedding layer to transform raw timestamps in trajectories into low-dimensional representation vectors. Specifically, it contains $\boldsymbol{t}_i^{weekly}, \boldsymbol{t}_i^{daily}, \boldsymbol{t}_i^{pos}, \boldsymbol{t}_i^{interval} \in \mathbb{R}^d$, which represent weekly periodic patterns, daily periodic patterns, position information, time interval information, respectively.

Encoding the Whole Trajectory. The whole trajectory can be divided into a segment sequence and a time sequence. Therefore, for the i-th visit, we first feed the segment v_i into the GAT to obtain the segment representation $\boldsymbol{h}_{v_i}^{Traj}$. Meanwhile, we feed the time sequence into the temporal embedding layer to obtain $\boldsymbol{t}_i^{weekly}, \boldsymbol{t}_i^{daily}, \boldsymbol{t}_i^{pos}$ and $\boldsymbol{t}_i^{interval}$. Then, we derive the overall representation $\boldsymbol{l}_i \in \mathbb{R}^d$ for the i-th visit in a trajectory by as follows:

$$\boldsymbol{l}_i = \boldsymbol{h}_{v_i}^{Traj} + \boldsymbol{t}_i^{weekly} + \boldsymbol{t}_i^{daily} + \boldsymbol{t}_i^{pos} + \boldsymbol{t}_i^{interval}. \tag{2}$$

In order to capture the long-range dependencies of visits in a trajectory and identify the global semantics of the trajectory, we further feed the representation sequence $[\boldsymbol{l}_i]_{i=1}^m$ into a transformer encoder (Vaswani et al. 2017) to obtain the final trajectory representation $\boldsymbol{l}_{\mathcal{T}} \in \mathbb{R}^d$ which can be mathematically defined as follows:

$$\boldsymbol{l}_{\mathcal{T}} = \mathrm{TrajTrans}(\boldsymbol{ph}, \boldsymbol{l}_1, \cdots, \boldsymbol{l}_m)[0], \tag{3}$$

where the function $\mathrm{TrajTrans}(\cdot)$ denotes a standard transformer encoder or a transformer-like encoder and $\boldsymbol{ph}$ is the embedding vector of the placeholder.

Masked Trajectory Prediction. The general idea of this task is to mask the consecutive subsegments of the trajectory and their corresponding timestamps, and to use linear layers to predict the masked values, *i.e.,* $\hat{y}^S \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{V}|}$ and $\hat{y}^T \in \mathbb{R}^{|\mathcal{T}|}$, respectively. The loss functions can be defined as follows:

$$\mathcal{L}_S^{Traj} = -\frac{1}{|\mathcal{M}_S|}\sum_{v_i\in\mathcal{M}_S}\log\frac{\exp(\hat{y}_{v_i}^S)}{\sum_{v_j\in\mathcal{V}}\exp(\hat{y}_{v_j}^S)}, \tag{4}$$

$$\mathcal{L}_T^{Traj} = \frac{1}{|\mathcal{M}_T|}\sum_{t_i\in\mathcal{M}_T}|\hat{y}_{v_i}^T - t_i|, \tag{5}$$

where $\mathcal{M}_S$ and $\mathcal{M}_T$ denote the sets of masked road segments and masked timestamps, respectively.

Contrastive Trajectory Learning. The general idea of this task is to adopt a contrastive learning strategy to generate multiple samples of a trajectory from different views and bring semantically similar samples closer in the representation space while dispersing dissimilar samples. The loss function can be defined as follows:

$$\mathcal{L}_{con}^{Traj}(\mathcal{T}_i,\mathcal{T}_j) = -\log\frac{\exp(\mathrm{sim}(\boldsymbol{l}_{\mathcal{T}_i},\boldsymbol{l}_{\mathcal{T}_j})/\tau)}{\sum_{k=1}^{2B}\mathbf{1}_{[k\neq i]}\exp(\mathrm{sim}(\boldsymbol{l}_{\mathcal{T}_i},\boldsymbol{l}_{\mathcal{T}_k})/\tau)}, \tag{6}$$

where B is the batch size, $(\mathcal{T}_i,\mathcal{T}_j)$ is a positive pair in the batch, $\mathbf{1}_{[k\neq i]} \in \{0,1\}$ is an indicator function that is equal to 1 if condition $k \neq i$ is satisfied and τ denotes a temperature parameter. The overall loss of this task for the batch, *i.e.,* $\mathcal{L}_{con}^{Traj}$, is computed by averaging the losses of all positive pairs in the batch.

Modeling Road Segments' Dynamic Features

In this part, we model the dynamic features of road segments that can change over time, including the spatial-temporal dynamics in trajectories and traffic state sequences.

Encoding Road Segment's Dynamic Spatial Features with Trajectory Data. Trajectory data entails some dynamic spatial semantics of segments. For example, a large transition probability between two segments revealed from the trajectory data may indicate that the two segments are nearby in the semantic space. Therefore, we replace the standard GAT in Section *Encoding Road Segment's Static and Spatial Features* with a Trajectory Transition-aware GAT. Specifically, based on the trajectories, we introduce a time-aware transition probability $p_{i,j,t}$ for the time slice t to capture the transition patterns between two segments v_i and v_j. $p_{i,j,t}$ considers the historical trajectories that occur periodically within the time slice t. To incorporate $p_{i,j,t}$ into the GAT, we compute a normalized attention weight $\alpha_{i,j,t}$ between v_i and v_j as follows:

$$\alpha_{i,j,t} = \frac{\exp(\text{LeakyReLU}(e_{i,j,t}))}{\sum_{k \in \mathcal{N}_{v_i}} \exp(\text{LeakyReLU}(e_{i,k,t}))}, \quad (7)$$

$$e_{i,j,t} = (\boldsymbol{h}'_{v_i,t}\boldsymbol{W}_1 + \boldsymbol{h}'_{v_j,t}\boldsymbol{W}_2 + p_{i,j,t}\boldsymbol{W}_3)\boldsymbol{W}_4^\top, \quad (8)$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d \times d'}$ and $\boldsymbol{W}_3, \boldsymbol{W}_4 \in \mathbb{R}^{1 \times d'}$ are learnable weight parameters, $\mathcal{N}_{v_i}$ is the neighborhood set of road segment v_i.

Traffic Data Embedding. The Traffic Data Embedding layer is designed to convert the traffic state sequence at time slice t, *i.e.,* $\mathcal{S}_t \in \mathbb{R}^{T \times N \times C_2}$, into an embedding tensor, *i.e.,* $\mathcal{X}_t^{Traf} \in \mathbb{R}^{T \times N \times d_x}$. Specifically, we first project $\mathcal{S}_t$ directly into a d_x-dimensional representation tensor, *i.e.,* $\mathcal{X}_t^{raw} = FC(\mathcal{S}_t) \in \mathbb{R}^{T \times N \times d_x}$, through a fully-connected feed-forward network $FC(\cdot) : \mathbb{R}^{T \times N \times C_2} \rightarrow \mathbb{R}^{T \times N \times d_x}$. Next, we employ three temporal embeddings, *i.e.,* $\boldsymbol{X}_t^{weekly}, \boldsymbol{X}_t^{daily}, \boldsymbol{X}_t^{pos} \in \mathbb{R}^{T \times d_x}$, to extract the weekly periodic patterns, daily periodic patterns and position information for all T time slices, respectively. To further model the spatial information of the road network, we also employ a multi-layer GAT to generate an embedding matrix $\boldsymbol{X}^{\mathcal{G}} \in \mathbb{R}^{N \times d_x}$ of the road network. The final embedding tensor $\mathcal{X}_t^{Traf}$ can then be computed as follows:

$$\mathcal{X}_t^{Traf} = \mathcal{X}_t^{raw} + \boldsymbol{X}_t^{weekly} + \boldsymbol{X}_t^{daily} + \boldsymbol{X}_t^{pos} + \boldsymbol{X}^{\mathcal{G}}. \quad (9)$$

Traffic Transformer Encoder. We further feed $\mathcal{X}_t^{Traf}$ into a traffic transformer encoder to model dynamic spatial-temporal dependencies hidden in traffic state sequences. The traffic transformer encoder is composed of multiple traffic transformer encoder layers. In each encoder layer, we first feed $\mathcal{X}_t^{Traf}$ into a spatial encoder and a temporal encoder, respectively. The spatial encoder takes a geographical and semantic neighbor-aware GAT layer to capture the dynamic spatial dependencies of traffic states in each time slice, whereas the temporal encoder takes a temporal self-attention layer to capture the dynamic temporal patterns for different road segments in the traffic state data. Next, we concatenate the output embedding of the spatial and temporal encoders to form a fusion representation, which is further fed into other components of a transformer encoder, *e.g.,* Add & Norm layer and Feed Forward layer. At the last of the traffic transformer encoder, we use a convolutional layer to transform the output embedding tensor into a matrix $\boldsymbol{H}_t^{Traf} \in \mathbb{R}^{N \times d}$, which is the final representation for N road segments at the time slice t. We summarize the computation of $\boldsymbol{H}_t^{Traf}$ as follows:

$$\boldsymbol{H}_t^{Traf} = \text{TrafTrans}(\mathcal{X}_t^{Traf}), \quad (10)$$

where the function $\text{TrafTrans}(\cdot) : \mathbb{R}^{T \times N \times d_x} \rightarrow \mathbb{R}^{N \times d}$ denotes the whole Traffic Transformer Encoder. Actually, other feasible spatio-temporal encoders can also replace this encoder.

Pre-training Traffic Data Embedding and Traffic Transformer Encoder via Mask State Prediction and Next State Prediction. We design two self-supervised tasks to learn generic segment representations. Specifically, we randomly mask a sequence of historical traffic states for each segment and use the generated intermediate representations to predict the masked traffic states, while using the generated segment representations of the next time slice to predict the traffic state of the next time slice. Ultimately, the loss $\mathcal{L}^{Traf}$ is obtained by the weighted sum of the losses from two tasks.

Modeling Multi-view Information Exchange

We propose a co-attentional transformer encoder to tackle the multi-view information exchange between different data modality, followed by a trajectory-traffic state matching task to maximize the consistency between segment representations and trajectory representations.

Co-Attentional Transformer Encoder. In each view, the key idea for the co-attentional transformer encoder is to replace the transformer encoder's multi-head self-attention module with a GAT layer, as shown in Figure 3, which aggregates the neighborhood nodes' features from the other view to form the node's feature representation in the target view. Moreover, in the GAT layer, inspired by Newton's law of universal gravitation (Simini et al. 2021), we assume that the influence between two segments decreases with the distance between them and design a novel way to compute the attention weight.

Specifically, we take the view of trajectory data to describe the mechanism of the co-attentional transformer encoder, and the co-attentional transformer encoder for the other view works similarly. We first define a K-minute reachable neighborhood set for segment v_i as $\mathcal{R}_{v_i}$, which is the set of segments that can reach v_i in K minutes. We use $\boldsymbol{h}_{v_i}^{Traj} \in \mathbb{R}^d$ to denote the representation vector of segment v_i at the time slice t produced by Trajectory Transition-aware GAT. To model the influence of information from the other view over $\boldsymbol{h}_{v_i}^{Traj}$, we employ a GAT layer, which defines a normalized attention weight $\alpha_{i,j}^{Traj}$ between v_i and

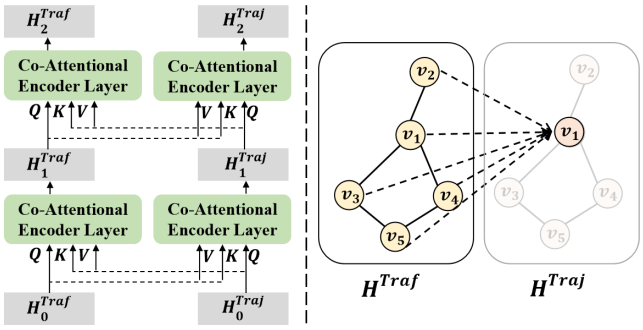

Figure 3: Framework of the Co-Attentional Transformer Encoder and the GAT Operation from the View of Trajectory Data.

$v_j \in \mathcal{R}_{v_i}$ as follows:

$$\alpha_{i,j}^{Traj} = \frac{\exp(\text{LeakyReLU}(e_{i,j}^{Traj}))}{\sum_{k \in \mathcal{R}_{v_i}} \exp(\text{LeakyReLU}(e_{i,k}^{Traj}))}, \quad (11)$$

$$e_{i,j}^{Traj} = (\boldsymbol{h}_{v_i}^{Traj}\boldsymbol{W}_5 + \boldsymbol{h}_{v_j}^{Traf}\boldsymbol{W}_6 + \text{deter}(v_i, v_j)\boldsymbol{W}_7)\boldsymbol{W}_8^\top, \quad (12)$$

where $\boldsymbol{W}_5, \boldsymbol{W}_6 \in \mathbb{R}^{d \times d'}$ and $\boldsymbol{W}_7, \boldsymbol{W}_8 \in \mathbb{R}^{1 \times d'}$ are learnable parameters, the function $\text{deter}(\cdot)$ denotes a geographical distance deterrence function such as Negative Power Function, and $\boldsymbol{h}_{v_j}^{Traf}$ denotes the representation vector of segment $v_j \in \mathbb{R}^d$ at the time slice t produced by the Traffic Transformer Encoder in the other view. The GAT layer uses the normalized attention weight $\alpha_{i,j}^{Traj}$ to aggregate the features of neighboring segments in the other view to obtain the feature representation of the targeted segment. The rest of the transformer block proceeds as before.

Trajectory-Traffic State Matching. We design a contrastive learning task, *i.e., Trajectory-Traffic State Matching*, which maximizes the agreement of a trajectory's representation vector and the corresponding road segment sequence's representation vector generated from traffic state data. The core of this task is that the trajectories of the current time slice correspond to the traffic state of the current time slice, rather than traffic states of other time slices.

Specifically, for a trajectory $\mathcal{T} = [\langle v_i, t_i \rangle]_{i=1}^m$, its representation $\boldsymbol{l}_\mathcal{T}$ can be extracted by the TRL process. In the meanwhile, a representation matrix $\boldsymbol{RS}_\mathcal{T}^{Traf} \in \mathbb{R}^{m \times d}$ for the trajectory's corresponding segment sequence $[v_i]_{i=1}^m$ can be extracted based on the traffic state data as follows:

$$\boldsymbol{RS}_\mathcal{T}^{Traf} = [\boldsymbol{h}_{v_i,t}^{Traf}]_{i=1}^m, \quad (13)$$

where $\boldsymbol{h}_{v_i,t}^{Traf} \in \mathbb{R}^d$ denotes the representation vector for road segment v_i at the timestamp t_1 (within the time slice t), and is derived based on $\boldsymbol{H}_t^{Traf}$ as described in Equation (10). Then, we can obtain the final representation vector $\boldsymbol{h}_{\mathcal{T},t}^{Traf} \in \mathbb{R}^d$ by average pooling over the first dimension of the matrix $\boldsymbol{RS}_\mathcal{T}^{Traf}$.

In the contrastive learning process, We define the spatio-temporal data of B_t time slices Θ as a training batch. For each trajectory $\mathcal{T} \in \mathcal{D}_t$ within the time slice $t \in \Theta$, we regard $(\boldsymbol{l}_\mathcal{T}, \boldsymbol{h}_\mathcal{T}^{Traf})$ as a positive pair. To construct negative pairs, we obtain $\boldsymbol{h}_{\mathcal{T},t'}^{Traf} \in \mathbb{R}^d$ from other time slices $t' \in \Theta$. Then, a negative pair can be constructed as $(\boldsymbol{l}_\mathcal{T}, \boldsymbol{h}_{\mathcal{T},t'}^{Traf})$. Formally, the loss function of the positive pair $(\boldsymbol{l}_\mathcal{T}, \boldsymbol{h}_{\mathcal{T},t}^{Traf})$ for the contrastive learning based on NT-Xent is defined as:

$$d_{\mathcal{T},t} = \exp(\text{sim}(\boldsymbol{l}_\mathcal{T}, \boldsymbol{h}_{\mathcal{T},t}^{Traf})/\tau), \quad (14)$$

$$\mathcal{L}^{Match}(\boldsymbol{l}_\mathcal{T}, \boldsymbol{h}_{\mathcal{T},t}^{Traf}) = -\log \frac{d_{\mathcal{T},t}}{d_{\mathcal{T},t} + \sum_{t' \in \Theta} d_{\mathcal{T},t'}}, \quad (15)$$

where τ denotes a temperature parameter. The total loss of contrastive learning for the batch, *i.e.,* $\mathcal{L}^{Match}$, is calculated by averaging the losses of all positive pairs in the batch.

Joint Pre-training for the Whole Model

To facilitate learning spatial-temporal patterns across multi-source data, we pre-train all modules of the proposed model in a joint manner, which defines the following loss function:

$$\mathcal{L} = \lambda^{Traj}\mathcal{L}^{Traj} + \lambda^{Traf}\mathcal{L}^{Traf} + \lambda^{Match}\mathcal{L}^{Match}, \quad (16)$$

where $\lambda^{Traj}, \lambda^{Traf}, \lambda^{Match}$ are three hyper-parameters that control the influence of each individual loss function over the proposed model, respectively.

Dataset	#Road Segment	#Edge	#Trajectory
Xi'an	5,168	12,643	834,560
Chengdu	6,153	15,783	1,262,406

Table 1: Statistics of the Two Datasets.

Experiments

Experimental Setup

We utilize two real-world datasets from two cities in China, *i.e.,* Xi'an and Chengdu, collected from the DiDi GAIA project in October and November 2018. The statistics for the two datasets are presented in Table 1. The duration of each time slice is 30 minutes. We chronologically split the traffic state data into a training, validation, and testing set with a ratio of 6:2:2. All experiments are repeated 10 times and the average results are reported according to Student's t-test at the 0.01 significance level. The number of dimension d in the representation learning is searched over $\{32, 64, 128\}$. The numbers of traffic transformer encoder layers, trajectory transformer encoder layers and co-attentional transformer encoder layers are 3, 6 and 2 respectively. More details are available at the code repository.

We evaluate the learned segment representations and trajectory representations on two downstream tasks, respectively, *i.e.,* Multi-Step Traffic State Prediction (MSTSP) and Travel Time Estimation (TTE). For the MSTSP task, we compare TRACK with 7 traffic state prediction methods, including DCRNN (Li et al. 2018b), GWNET (Wu et al. 2019b), MTGNN (Wu et al. 2020b), TrGNN (Li et al. 2021), STGODE (Fang et al. 2021), ST-Norm (Deng et al. 2021)

Multi-Step Traffic State Prediction							Travel Time Estimation						
Dataset	Xi'an			Chengdu			Dataset	Xi'an			Chengdu		
Models	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	Models	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE
DCRNN	1.288	16.38	2.491	1.554	18.21	2.860	traj2vec	1.667	23.44	2.465	1.296	22.16	1.915
GWNET	1.297	15.58	2.334	1.610	18.13	2.707	t2vec	1.663	23.38	2.463	1.296	22.21	1.923
MTGNN	1.222	14.90	2.161	1.470	16.76	2.495	Trembr	1.668	23.76	2.470	1.315	22.63	1.948
TrGNN	1.255	15.89	2.415	1.559	17.68	2.763	PIM	1.692	24.65	2.487	1.322	23.67	1.920
STGODE	1.391	17.34	2.297	1.628	18.76	2.602	Toast	1.720	22.78	2.582	1.331	21.96	1.976
ST-Norm	1.270	15.64	2.276	1.485	17.00	2.593	JCLRNT	1.799	24.91	2.576	1.370	23.95	1.987
SSTBAN	1.175	14.11	2.193	1.454	16.90	2.491	START	1.522	22.26	2.169	1.182	20.02	1.749
TRACK	**1.094**	**13.32**	**2.141**	**1.363**	**15.42**	**2.471**	TRACK	**1.426**	**20.74**	**1.988**	**1.143**	**19.43**	**1.612**

Table 2: Performance Comparsion.

and SSTBAN (Guo et al. 2023). For the TTE task, we compare TRACK with 7 trajectory representation learning methods, including traj2vec (Yao et al. 2017), t2vec (Li et al. 2018a), Trembr (Fu and Lee 2020), PIM (Yang et al. 2021), Toast (Chen et al. 2021), JCLRNT (Mao et al. 2022) and START (Jiang et al. 2023c).

Performance Comparison

Table 2 reports the results for two downstream tasks. The bold results are the best, and the underlined results are the second best. It can be seen that our proposed TRACK model outperforms all baselines on both datasets. This demonstrates the effectiveness of TRACK in learning effective road network representations and trajectory representations.

For the MSTSP task, MTGNN and SSTBAN achieve competitive performance compared to other baselines. This is because MTGNN proposes an adaptive graph generation module to reflect realistic spatial correlations while SSTBAN employs a self-supervised learner and designs a spatial botteleneck attention mechanism to capture global spatial dynamics. In contrast, TRACK achieves better performance because it further incorporates transition patterns of segments based on trajectory data in modeling the spatial-temporal dynamics. For the TTE task, START consistently outperforms other baselines for all metrics and datasets. One important reason is that it captures the temporal dynamics of trajectories and therefore enables the trajectories passing through the same route at different time slices may have different representations. However, START only encodes the timestamp information of trajectories for the periodic patterns of urban traffic, while TRACK also learns the spatial-temporal dynamics in the short-term traffic states.

Ablation Study

To further investigate the effects of different components in TRACK, we perform ablation studies with the following variants. (1) *w/o Traf*: this variant eliminates modeling of the spatial-temporal dynamics in traffic state sequences; (2) *w/o Traj*: this variant eliminates the process of trajectory representation learning and modeling of dynamic spatial features in trajectories; (3) *w/o Match*: this variant eliminates the trajectory-traffic state matching task; (4) *w/o TMask*: this variant removes the loss $\mathcal{L}_T^{Traj}$ of the MTP task, which means no masking prediction for timestamps; (5) *w/o SMask*: this variant removes the loss $\mathcal{L}_S^{Traj}$ of the MTP task, which means no masking prediction for segments; (6) *w/o Contra*: this variant removes the loss $\mathcal{L}_{Con}^{Traj}$ of the CTL task.

Figure 4 shows the performance of these variants on the MSTSP and TTE tasks of the Xi'an dataset. The following observations can be made. Firstly, the variants *w/o Traj* and *w/o Traf* are consistently inferior to TRACK on both tasks. This demonstrates that traffic state and trajectory data can serve as side information of each other to enhance the representation learning process of road networks and trajectories. Specifically, trajectory representations can perceive dynamic traffic states on the road network, while segment representations can perceive dynamic dependencies between upstream and downstream traffic flows. Secondly, TRACK performs better than *w/o Match*. This indicates that the Trajectory-Traffic State Matching task can indeed help to enhance the representation learning process of the whole model. Third, the performance of *w/o TMask*, *w/o SMask*, and *w/o Contra* on both tasks is inferior to TRACK. This indicates that both MTP and CTL tasks contribute to more accurate representations of dynamic road networks and trajectories, which in turn impacts their performance on two downstream tasks.

Case Study

In this section, we conduct two case studies to visualize and analyze the dynamic segment representations and trajectory representations, which enhance TRACK's interpretability.

Case 1: Study on Dynamic Road Segment Representations. One advantage of the proposed model is to learn dynamic road segment representations. We take segments A, B and C in Figure 5(a) as an example to investigate this type of representation. The incoming traffic flow of segment A is composed of the outgoing traffic flow of segments B and C. We take October 11th as the observation period and set the window of the time slice as 30 minutes. Then we adopt t-SNE (Van der Maaten and Hinton 2008) to visualize the learned segment representations in Figure 5(b), where each point corresponds to the representation of a segment within a time slice. We can see that the learned segment representation indeed changes over time. Moreover, we use *SimRatio(A,B)* to measure the ratio of the similarity between segments A and B and the similarity between segments A and C. We then plot *SimRatio(A,B)* and the transition probabilities from segment B to A, denoted as *TransProb(B→A)*, over

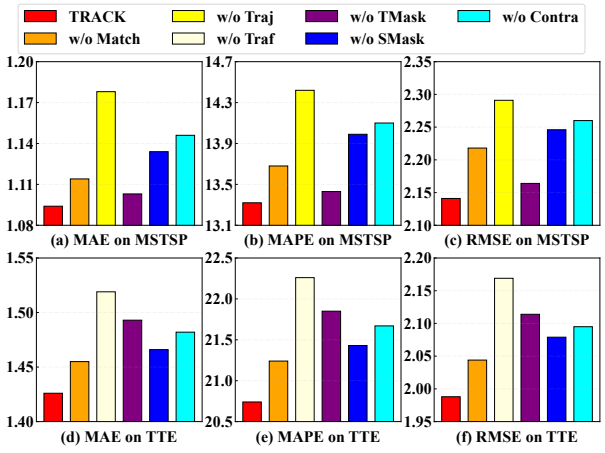

Figure 4: Ablation Study on the Xi'an Dataset.

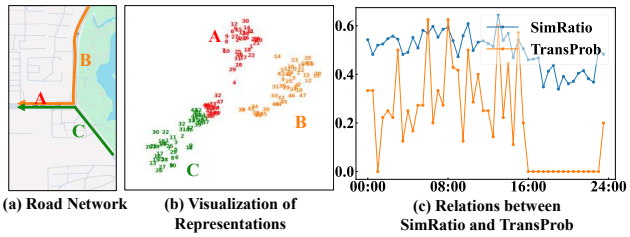

Figure 5: Case Study of Dynamic Segment Representations.

time in Figure 5(c). It can be observed that the transition patterns indeed change over time. More importantly, we can see that from 16:00 to 23:00 there is a significant decrease in *SimRatio(A,B)*, which is due to the fact that there were no trajectories from B to A in that period. This suggests that the learned segment representations can indeed capture the dynamic transition patterns between road segments.

Case 2: Study on Dynamic Trajectory Representations. Another advantage of the proposed model is that with the information exchange of different views, the learned trajectory representations can also be influenced by the dynamics of traffic states. We take the three routes R_1, R_2, and R_3 from segment 131 to segment 2768 on November 1st shown in Figure 6(a) as an example. In Figure 6(b), we visualize trajectory representations of the three routes at different time slices, which shows that the learned trajectory representations of the same route are indeed dynamic over time. Moreover, compared to R_3, R_1 and R_2 are closer in the trajectory representation space. This can be explained by the road network's spatial semantics because R_2 has a shorter detouring path than R_3 and is more similar to R_1. Meanwhile, the trajectory representation of R_1 at the 20th time slice (*i.e.,* departing at 10:00) is more similar to those of R_2 than those of R_1 at other time slices. This can be explained by that there was a sharp decrease in traffic speed on segment 96 at 10:00 as shown in Figure 6(c), which makes the semantics of R_1 at that time slice resembles the semantics of detouring in R_1. This indicates that trajectory representation generated by TRACK can indeed perceive the dynamic traffic states of the road segments visited in the trajectory.

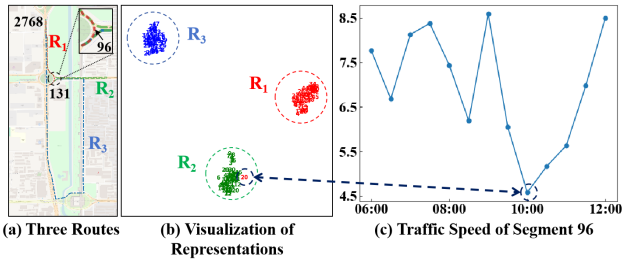

Figure 6: Case Study of Trajectory Representations.

Related Work

Road Network Representation Learning aims to transform the road network into a general low-dimensional representation matrix. Since graph representation learning methods can model the topological structure of the road network (Perozzi, Al-Rfou, and Skiena 2014; Tang et al. 2015; Grover and Leskovec 2016; Hamilton, Ying, and Leskovec 2017), some existing methods (Wang et al. 2019b; Chen et al. 2022; Chang et al. 2023) take the spatial correlations of road segments into account by using graph representation learning methods. A more recent method (Wu et al. 2020a; Yu et al. 2024) takes the transition patterns into account when modeling the road networks. In summary, our proposed method is the first attempt to jointly model the dynamics of road segments based on trajectory and traffic state data.

Trajectory Representation Learning aims to transform the trajectory into a general low-dimensional representation vector. Early TRL studies (Yao et al. 2017; Li et al. 2018a) obtain trajectory representations through the reconstruction task. Recent TRL methods (Zhu et al. 2022; Yang et al. 2021; Liu et al. 2022; Mao et al. 2022; Yang et al. 2022; Chen et al. 2021; Lin et al. 2023; Jiang et al. 2023c) primarily first obtain the road network representations and then derive trajectory representations through sequential models with self-supervised tasks. The trajectory representation is assumed to be static in most methods, with only a few encoding temporal information in trajectories. For example, Trembr (Fu and Lee 2020) and START (Jiang et al. 2023c) reconstructs timestamps during the decoding process. However, they do not consider the impact of dynamic traffic states on trajectory representations, which is one of our main contributions.

Conclusion

We propose TRACK, a novel dynamic road network and trajectory representation learning framework, to jointly model traffic state data and trajectory data. Extensive experiments on two real-world datasets showcase the performance of TRACK and provide interpretations of dynamic road segment and trajectory representations. TRACK offers a promising way for improving traffic-related tasks, contributing to more efficient and sustainable urban management.

Acknowledgments

Prof. Jingyuan Wang's work was partially supported by the National Natural Science Foundation of China (No. 72171013, 72222022), and the Special Fund for Health Development Research of Beijing (2024-2G-30121). Prof. Junjie Wu's work was partially supported by the National Natural Science Foundation of China (72242101, 72031001), and the Outstanding Young Scientist Program of Beijing Universities (JWZQ20240201002). Dr. Hao Lin's work was partially supported by the National Natural Science Foundation of China (72301017) and the Shenzhen Science and Technology Program (CJGJZD20230724093201004).

References

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *NeurIPS*.

Chang, Y.; Tanin, E.; Cao, X.; and Qi, J. 2023. Spatial Structure-Aware Road Network Embedding via Graph Contrastive Learning. In *EDBT*, 144–156. OpenProceedings.org.

Chen, K.; Chu, G.; Lei, K.; Shi, Y.; and Deng, M. 2022. A Multiview Representation Learning Framework for Large-Scale Urban Road Networks. *Applied Sciences*, 12(13): 6301.

Chen, L.; Gao, Y.; Fang, Z.; Miao, X.; Jensen, C. S.; and Guo, C. 2019. Real-time distributed co-movement pattern detection on streaming trajectories. *Proceedings of the VLDB Endowment*, 12(10): 1208–1220.

Chen, L.; Gao, Y.; Li, X.; Jensen, C. S.; and Chen, G. 2017. Efficient Metric Indexing for Similarity Search and Similarity Joins. *IEEE Transactions on Knowledge and Data Engineering*, 29(3): 556–571.

Chen, L.; Zhong, Q.; Xiao, X.; Gao, Y.; Jin, P.; and Jensen, C. S. 2018. Price-and-time-aware dynamic ridesharing. In *2018 IEEE 34th international conference on data engineering (ICDE)*, 1061–1072. IEEE.

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*, 211–220. ACM.

Deng, J.; Chen, X.; Jiang, R.; Song, X.; and Tsang, I. W. 2021. ST-Norm: Spatial and Temporal Normalization for Multi-variate Time Series Forecasting. In *KDD*, 269–278. ACM.

Ding, X.; Chen, L.; Gao, Y.; Jensen, C. S.; and Bao, H. 2018. UlTraMan: A unified platform for big trajectory data management and analytics. *Proceedings of the VLDB Endowment*, 11(7): 787–799.

Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *KDD*, 364–373. ACM.

Fu, T.-Y.; and Lee, W.-C. 2020. Trembr: Exploring road networks for trajectory representation learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1): 1–25.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD*, 855–864. ACM.

Guo, S.; Lin, Y.; Gong, L.; Wang, C.; Zhou, Z.; Shen, Z.; Huang, Y.; and Wan, H. 2023. Self-Supervised Spatial-Temporal Bottleneck Attentive Network for Efficient Long-term Traffic Forecasting. In *ICDE*, 1585–1596. IEEE.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*, 1024–1034.

Ji, J.; Wang, J.; Huang, C.; Wu, J.; Xu, B.; Wu, Z.; Zhang, J.; and Zheng, Y. 2023. Spatio-Temporal Self-Supervised Learning for Traffic Flow Prediction. In *AAAI*, 4356–4364. AAAI Press.

Ji, J.; Wang, J.; Jiang, Z.; Jiang, J.; and Zhang, H. 2022a. STDEN: Towards Physics-Guided Neural Networks for Traffic Flow Prediction. In *AAAI*, 4048–4056. AAAI Press.

Ji, J.; Wang, J.; Jiang, Z.; Ma, J.; and Zhang, H. 2020. Interpretable spatiotemporal deep learning model for traffic flow prediction based on potential energy fields. In *2020 IEEE international conference on data mining (ICDM)*, 1076–1081. IEEE.

Ji, J.; Wang, J.; Wu, J.; Han, B.; Zhang, J.; and Zheng, Y. 2022b. Precision CityShield against hazardous chemicals threats via location mining and self-supervised learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3072–3080.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023a. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. In *AAAI*, 4365–4373. AAAI Press.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023b. Unified Data Management and Comprehensive Performance Evaluation for Urban Spatial-Temporal Prediction [Experiment, Analysis & Benchmark]. *CoRR*, abs/2308.12899.

Jiang, J.; Pan, D.; Ren, H.; Jiang, X.; Li, C.; and Wang, J. 2023c. Self-supervised Trajectory Representation Learning with Temporal Regularities and Travel Semantics. In *ICDE*, 843–855. IEEE.

Jiang, W.; Zhao, W. X.; Wang, J.; and Jiang, J. 2023d. Continuous Trajectory Generation Based on Two-Stage GAN. In *AAAI*, 4374–4382. AAAI Press.

Li, M.; Tong, P.; Li, M.; Jin, Z.; Huang, J.; and Hua, X. 2021. Traffic Flow Prediction with Vehicle Trajectories. In *AAAI*, 294–302. AAAI Press.

Li, X.; Zhao, K.; Cong, G.; Jensen, C. S.; and Wei, W. 2018a. Deep Representation Learning for Trajectory Similarity Computation. In *ICDE*, 617–628. IEEE Computer Society.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018b. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR (Poster)*. OpenReview.net.

Lin, Y.; Wan, H.; Guo, S.; Hu, J.; Jensen, C. S.; and Lin, Y. 2023. Pre-Training General Trajectory Embeddings With Maximum Multi-View Entropy Coding. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, X.; Tan, X.; Guo, Y.; Chen, Y.; and Zhang, Z. 2022. CSTRM: Contrastive Self-Supervised Trajectory Representation Model for trajectory similarity computation. *Comput. Commun.*, 185: 159–167.

Liu, Z.; Wang, J.; Li, Z.; and He, Y. 2024. Full Bayesian Significance Testing for Neural Networks in Traffic Forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*.

Mao, Z.; Li, Z.; Li, D.; Bai, L.; and Zhao, R. 2022. Jointly Contrastive Representation Learning on Road Network and Trajectory. In *CIKM*, 1501–1510. ACM.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: online learning of social representations. In *KDD*, 701–710. ACM.

Simini, F.; Barlacchi, G.; Luca, M.; and Pappalardo, L. 2021. A Deep Gravity model for mobility flows generation. *Nature communications*, 12(1): 6576.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. LINE: Large-scale Information Network Embedding. In *WWW*, 1067–1077. ACM.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR (Poster)*. OpenReview.net.

Wang, D.; Zhang, J.; Cao, W.; Li, J.; and Zheng, Y. 2018. When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks. In *AAAI*, 2500–2507. AAAI Press.

Wang, J.; Ji, J.; Jiang, Z.; and Sun, L. 2022. Traffic flow prediction based on spatiotemporal potential energy fields. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 9073–9087.

Wang, J.; Jiang, J.; Jiang, W.; Li, C.; and Zhao, W. X. 2021. LibCity: An Open Library for Traffic Prediction. In *SIGSPATIAL/GIS*, 145–148. ACM.

Wang, J.; Wu, N.; Zhao, W. X.; Peng, F.; and Lin, X. 2019a. Empowering A* search algorithms with neural networks for personalized route recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 539–547.

Wang, M.; Lee, W.; Fu, T.; and Yu, G. 2019b. Learning Embeddings of Intersections on Road Networks. In *SIGSPATIAL/GIS*, 309–318. ACM.

Wu, N.; Wang, J.; Zhao, W. X.; and Jin, Y. 2019a. Learning to Effectively Estimate the Travel Time for Fastest Route Recommendation. In *CIKM*, 1923–1932. ACM.

Wu, N.; Zhao, W. X.; Wang, J.; and Pan, D. 2020a. Learning Effective Road Network Representation with Hierarchical Graph Neural Networks. In *KDD*, 6–14. ACM.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020b. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *KDD*, 753–763. ACM.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019b. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *IJCAI*, 1907–1913. ijcai.org.

Yang, S. B.; Guo, C.; Hu, J.; Tang, J.; and Yang, B. 2021. Unsupervised Path Representation Learning with Curriculum Negative Sampling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, 3286–3292. ijcai.org.

Yang, S. B.; Guo, C.; Hu, J.; Yang, B.; Tang, J.; and Jensen, C. S. 2022. Weakly-supervised Temporal Path Representation Learning with Contrastive Curriculum Learning. In *ICDE*, 2873–2885. IEEE.

Yao, D.; Zhang, C.; Zhu, Z.; Huang, J.; and Bi, J. 2017. Trajectory clustering via deep representation learning. In *2017 international joint conference on neural networks (IJCNN)*, 3880–3887. IEEE.

Yi, Z.; Zhou, Z.; Huang, Q.; Chen, Y.; Yu, L.; Wang, X.; and Wang, Y. 2024. Get Rid of Task Isolation: A Continuous Multi-task Spatio-Temporal Learning Framework. *CoRR*, abs/2410.10524.

Yu, X.; Wang, J.; Yang, Y.; Huang, Q.; and Qu, K. 2024. BIGCity: A Universal Spatiotemporal Model for Unified Trajectory and Traffic State Data Analysis. *arXiv preprint arXiv:2412.00953*.

Zhu, G.; Sang, Y.; Chen, W.; and Zhao, L. 2022. When Self-attention and Topological Structure Make a Difference: Trajectory Modeling in Road Networks. In *APWeb/WAIM (3)*, volume 13423 of *Lecture Notes in Computer Science*, 381–396. Springer.

Zou, D.; Wang, S.; Li, X.; Peng, H.; Wang, Y.; Liu, C.; Sheng, K.; and Zhang, B. 2024. MultiSPANS: A Multi-range Spatial-Temporal Transformer Network for Traffic Forecast via Structural Entropy Optimization. In *WSDM*, 1032–1041. ACM.