8-2022

Interpreting trajectories from multiple views: A hierarchical self-attention network for estimating the time of arrival

Zebin CHEN
South China University of Technology

Xiaolin XIAO

Yue-Jiao GONG

Jun FANG
Didi Chuxing

Nan MA

See next page for additional authors

Citation

CHEN, Zebin; XIAO, Xiaolin; GONG, Yue-Jiao; FANG, Jun; MA, Nan; CHAI, Hua; and CAO, Zhiguang. Interpreting trajectories from multiple views: A hierarchical self-attention network for estimating the time of arrival. (2022). *KDD '22: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, August 14-18*. 2771-2779.

Available at: https://ink.library.smu.edu.sg/sis_research/8136

Author

Zebin CHEN, Xiaolin XIAO, Yue-Jiao GONG, Jun FANG, Nan MA, Hua CHAI, and Zhiguang CAO

Interpreting Trajectories from Multiple Views: A Hierarchical Self-Attention Network for Estimating the Time of Arrival

Zebin Chen
Xiaolin Xiao[†]
Yue-Jiao Gong[†]
cszebinc0409@mail.scut.edu.cn
shellyxiaolin@gmail.com
gongyuejiao@gmail.com
South China University of Technology
Guangzhou, China

Jun Fang
Nan Ma
Hua Chai
fangjun@didiglobal.com
mandymanan@didiglobal.com
chaihua@didiglobal.com
Didi Chuxing
Beijing, China

Zhiguang Cao
zhiguangcao@outlook.com
Singapore Institute of Manufacturing Technology
Singapore

ABSTRACT

Estimating the time of arrival is a crucial task in intelligent transportation systems. Although considerable efforts have been made to solve this problem, most of them decompose a trajectory into several segments and then compute the travel time by integrating the attributes from all segments. The segment view, though being able to depict the local traffic conditions straightforwardly, is insufficient to embody the intrinsic structure of trajectories on the road network. To overcome the limitation, this study proposes multi-view trajectory representation that comprehensively interprets a trajectory from the segment-, link-, and intersection-views. To fulfill the purpose, we design a hierarchical self-attention network (HierETA) that accurately models the local traffic conditions and the underlying trajectory structure. Specifically, a segment encoder is developed to capture the spatio-temporal dependencies at a fine granularity, within which an adaptive self-attention module is designed to boost performance. Further, a joint link-intersection encoder is developed to characterize the natural trajectory structure consisting of alternatively arranged links and intersections. Afterward, a hierarchy-aware attention decoder is designed to realize a tradeoff between the multi-view spatio-temporal features. The hierarchical encoders and the attentive decoder are simultaneously learned to achieve an overall optimality. Experiments on two large-scale practical datasets show the superiority of HierETA over the state-of-the-arts.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; • **Computing methodologies → Neural networks**.

KEYWORDS

Estimating the Time of Arrival; Self-Attention Network; Hierarchical Representation Learning

ACM Reference Format:
Zebin Chen, Xiaolin Xiao, Yue-Jiao Gong, Jun Fang, Nan Ma, Hua Chai, and Zhiguang Cao. 2022. Interpreting Trajectories from Multiple Views: A Hierarchical Self-Attention Network for Estimating the Time of Arrival. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539051

1 INTRODUCTION

The task of estimating the time of arrival (ETA) for a trip plays an important role in intelligent transportation systems. It is widely used in fields of route planning, navigation, online ride-hailing, and congestion control, etc [4, 11, 20, 24, 29, 32, 45, 50]. Nevertheless, the accurate estimation of travel time remains a challenge as a diverse spectrum of events can significantly affect travel demand such as the spatial and temporal dependencies of the trajectories, the weather condition, and rush hours or peak-off hours [3, 6, 8, 14, 19, 25, 27, 33, 35, 42]. To accurately model the local traffic factors, traditional ETA algorithms mainly employ the divide-and-conquer strategy by representing a trajectory as a segment sequence and then summing up the local predictions [5]. The local prediction errors, however, may rapidly accumulate using the segment-view representation. Recently, deep learning methods alleviate this problem by explicitly modeling the spatio-temporal characteristics of trajectories on the road network in an integrated manner [7, 38, 42].

Although many progresses have been reported, most deep learning based models empirically adopt a single-view representation, i.e., the segment view, and thus suffer the representation deficiency. As shown in Figure 1, the segment-view representation is artificially produced to capture the fined-grained local traffic conditions, which is however not comprehensive in characterizing the natural structure of the road network. More specifically, a trajectory naturally consists of alternatively arranged links and intersections. The links usually preserve static road characteristics, such as pavement type, road width and road functional level [18], while the intersections could provide valued information such as the waiting time, the number of traffic lights, and the historical traffic volume. The link-, intersection-, and segment-views work together to jointly depict the underlying characteristics of the trajectory via a hierarchical

[†] Corresponding Authors.

KDD '22, August 14–18, 2022, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539051

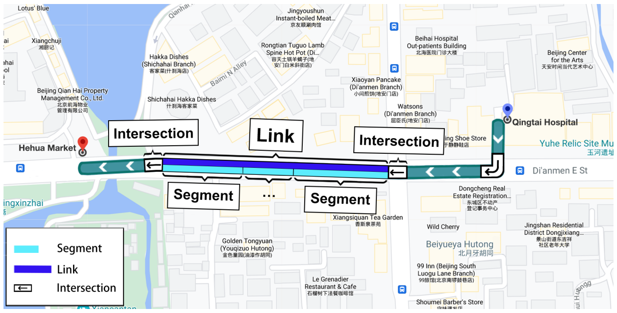


Figure 1: A trajectory in the ETA application is commonly represented by a segment sequence. In this work, we propose to interpret trajectories jointly from the segment-, link-, and intersection-views. The three views work together to provide a comprehensive exploration of trajectories.

structure. That is, the link- and intersection-views characterize the trajectory attributes from a coarse perspective; a link can be further decomposed into several segments, and hence the segment-view representation models the spatial dependencies at a fine granularity.

It is a nontrivial task to integrate the above three views effectively and efficiently. On the one hand, an intuitive way is to roughly approximate the link-view traffic factors using the segment-view representation directly instead of developing a higher-level representation specifically [9, 26, 42]. However, without explicitly modeling the link-view characteristics, existing studies can hardly model the coherent consistency across segments within the same links. On the other hand, in real scenarios, the numbers and attribute types of segments and intersections are naturally inconsistent. As such, it is hard to jointly model these two views in a single-module network, and the intersection information is therefore largely ignored [7, 18, 42] or oversimplified [21, 23, 40]. This may lead to serious representation deficiency since the travel time is strictly influenced by the intersection conditions in practice. For example, the waiting time may accumulate quickly on urban roads where intersections are densely distributed [47].

To fix the above-mentioned problems, it is highly motivated to simultaneously utilize the segment-, link-, and intersection-views in an organic manner for efficiently interpreting the trajectories. We thus propose a **<u>Hier</u>**archical Self-Attention Network for **<u>E</u>**stimating the **<u>T</u>**ime of **<u>A</u>**rrival (**HierETA**). Specifically, the segment-view representation enables the flexibility in modeling the local traffic conditions. Then, as some inherent trajectory attributes are preserved within a link, we use the link-view representation to share these common attributes. Meanwhile, the intersections are used to elegantly model the indirect factors, e.g., the waiting time caused by complex factors. In contrast to simply integrating the link and intersection features by pending them on the traditional segment-view representation, HierETA exploits the hierarchical relationship among the three views to portray the underlying road structure. The main contributions of HierETA are summarized as follows:

- As far as we know, this is the first work that explicitly represents trajectories on the road network using multi-view representation learning. The proposed hierarchical self-attention network organizes the segment-, link-, and intersection-views efficiently according to their natural relationships.
- We design an adaptive self-attention network to jointly leverage the global and local patterns for spatio-temporal dependency modeling within the multi-view representation framework.
- A hierarchy-aware attention decoder is devised to estimate the travel time using the learned context features from different granularities, being capable of balancing the segment, link, and intersection features.
- We evaluate the performance of HierETA on two large-scale datasets from Didi Chuxing with over ten million trajectories. Experimental results demonstrate the superiority of HierETA over the state-of-the-arts.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 outlines the preliminary concepts and formulates the ETA problem. Section 4 details the structure of the proposed HierETA. The experimental results and analysis are presented in Section 5. Finally, Section 6 concludes the paper.

2 RELATED WORKS

2.1 Trajectory Data Mining

Trajectory data is a fundamental component of the intelligent transportation systems, and is commonly seen in many traffic-related applications, such as traffic flow prediction [5, 20, 25], travel time estimation [38, 42], trajectory recovery and inference [15, 34, 36, 44]. Generally, the trajectory data shows rich spatial-moving patterns and explicitly provides structural constraints and traffic semantics of road networks [51]. Recently, the graph-structured data has been extensively investigated as well [3, 7, 19, 50]. For instance, GMAN [50] employs a graph multi-attention structure to extract the spatial and temporal relationships. However, graph representation learning generally suffers from the negative impact from irrelevant spatial neighboring regions, resulting in error propagation especially when the involved area grows larger [28]. This way, graph modeling is limited to process only narrow neighboring regions and falls short on developing large-scale urban-wise systems [7].

2.2 Estimating the Time of Arrival

The task of estimating the time of arrival has been studied for decades. Existing ETA models chiefly fall into two categories: the origin-destination methods and route-based methods.

The origin-destination methods make predictions based on existing trips with similar origins and destinations. As examples, TEMP [41] calculates the weighted average of neighboring trips to estimate the travel time of a query trip; T-drive [48] captures the dynamic traffic patterns by a time-dependent land-mark graph, and then estimates the distribution of travel time between two landmarks by clustering; the work in [1] introduces a dynamic Bayesian network to model traffic congestion state of various road segments and searches for optimal concatenation of these segments to predict the travel time. However, algorithms in this category usually ignore the informative route attributes and hence are inadequate to model the complex spatio-temporal features.

The route-based methods further comprise the segment aggregation methods and deep learning methods. Among them, the former separately consider the travel time of segments and intersections. To be specific, SMA [21] models the correlation across road segments with a spatial-moving average structure, while the model in [43] uses the support vector regression to predict the travel time. The relationship between road segments is considered in [46] and a spatio-temporal Hidden Markov model is introduced to obtain the correlations among adjacent segments. Though intuitive, these methods may encounter the serious error accumulation problem without modeling the spatio-temporal correlation of the segments.

The deep learning methods have received particular attention and achieved considerable improvements in recent years. They generally focus on modeling the spatial and temporal dependencies of the road network so as to improve the modeling accuracy. For instance, DeepTTE [40] transforms trajectories into raw GPS sequences and utilizes geo-convolutional network and LSTM to learn the spatio-temporal dependencies; WDR [42] and its variant [38] adopt a wide-deep-recurrent network to capture the contextual information of route attributes; ConSTGAT [7] and CompactETA [9] explore the joint relations of spatial and temporal information using the graph attention network; the work in [23] proposes a CNN to integrate the trajectory data with the information of morphological layout images, providing rich information on the surrounding environments; DeepGTT [26] uses a deep generative model for learning the distribution of travel time; HetETA [18] learns the representation of spatio-temporal information using a multi-relational network; TTPNet [37] extracts the travel speed and representation of road network from historical trajectories based on tensor decomposition and graph embedding.

However, none of existing methods take into account the multi-view representation of the trajectory, and hence suffer the representation deficiency.

2.3 Self-Attention Network

The recent development of the self-attention network [39] has established the state-of-the-art benchmarks and attracted lots of interests, owning to its high efficiency in modeling long-term dependencies and the ability in parallel computation. Self-attention deals with the inner-dependencies within a sequence, and thus is able to learn the sequential patterns and internal correlations. Specifically, an input sequence $H = \{H_1, ..., H_n\}$ is treated as a bag-of-word tokens. Formally, H is first projected into three matrices: queries $Q \in \mathbb{R}^{n\times d}$, keys $K \in \mathbb{R}^{n\times d}$, and values $V \in \mathbb{R}^{n\times d}$ by

$$Q, K, V = HW_Q, HW_K, HW_V, \tag{1}$$

where $\{W_Q, W_K, W_V\} \in \mathbb{R}^{d\times d}$ are trainable parameters. Then, the output states are calculated by attending to any two tokens as

$$\hat{H} = \text{softmax}(\frac{QK^T}{\sqrt{d}})V, \tag{2}$$

where $\sqrt{d}$ is the scaling factor.

In this work, we design an adaptive self-attention network to explicitly capture the spatio-temporal dependencies of the trajectory using multi-view sequences. With carefully designed network structures, our model is able to adaptively capture the dependencies over the segment-, link-, and intersection-views.

3 PROBLEM DEFINITIONS

In this section, we first introduce the multi-view representation of trajectories and then formally formulate the ETA task.

Multi-View Trajectory Representation. In this study, a trajectory T is characterized by three views: the segment view $T_S = \{s_{ij}\}_{i=1,j=1}^{m,n}$, the link view $T_L = \{l_i\}_{i=1}^{m}$, and the intersection view $T_I = \{c_i\}_{i=1}^{m}$, where n is the number of segments within a link and m denotes the number of links and intersections. Each view is packed with a set of view-specific attributes, such as the length, width of the segment view, and the historical traffic volume of the intersection view. Besides, a trajectory also contains some external factors such as the day of week, the start timeslot, the driving style and the total travel distance. The external factors are also involved in our task.

Estimating the Time of Arrival. In this study, the goal of ETA is to estimate the duration time of a query trajectory T_q by jointly modeling the spatio-temporal dependencies of the road network. We assume that the query trajectory is specified by the user or generated by the route planing applications.

4 MODEL ARCHITECTURE

In this section, we describe the architecture of HierETA. As shown in Figure 2, we introduce a hierarchical self-attention network for multi-view trajectory representation. The learned multi-view trajectory features work together to comprehensively model the underlying structure of trajectories on the road network for travel time estimation. More specifically, an attribute feature extractor is designed as a pre-requisite component for subsequent modules. Then, we use a segment encoder to describe the local traffic conditions at a fine-scale, while the joint link-intersection encoder captures the trajectory attributes from a coarse perspective. Finally, a hierarchy-aware attention decoder is introduced to generate context features for travel time estimation. We will elaborate the modules of HierETA as follows.

4.1 Attribute Feature Extractor

We first introduce an attribute feature extractor to learn the multi-view contextual features, i.e., the segment, link, and intersection attributes. The attributes in all views are either continuous or categorical. We apply the Z-score method to the continuous attributes for data normalization and utilize the embedding strategy [10] to learn the features of categorical ones, and then concatenate the associated vectors as the view-specific feature embedding. Here, the features of the segment-, link- and intersection-views are represented as $\{x_{ij}^s\}_{i=1,j=1}^{m,n}$, $\{x_i^l\}_{i=1}^m$ and $\{x_i^c\}_{i=1}^m$, respectively. Physically, the segment-view feature models the spatial dependencies at a fine granularity, while the link- and intersection-views characterize the trajectory attributes from a coarse perspective. Besides, the external impact factors are shared across the whole trajectory and are represented as x^r.

In the following subsections, attributes from three views are fully exploited for trajectory representation.

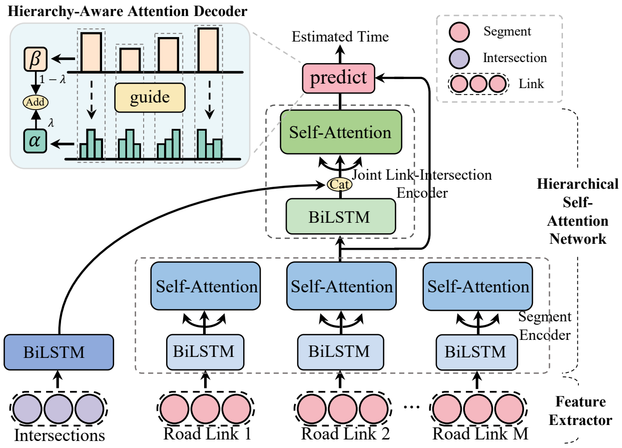


Figure 2: The framework of HierETA. The proposed model is divided into three submodules: an attribute feature extractor, a hierarchical self-attention network comprising a segment encoder and a joint link-intersection encoder, and a hierarchy-aware attention decoder.

4.2 Hierarchical Self-Attention Network for Multi-View Trajectory Representation

To effectively interpret the underlying structure of the trajectories from multiple views, we design a hierarchical self-attention network comprising a segment encoder and a joint link-intersection encoder.

Segment Encoder. Given the attribute feature x^s_{ij} of segment s_{ij}, we first stick x^r to x^s_{ij} so as to take the external impact factors into consideration. Then, we employ a BiLSTM [17] to sequentially encode the j-th concatenated feature vectors[1] $[x^s_j|x^r]$ and is represented as H^s_j by concatenating the forward and backward hidden states.

As previously mentioned, the accurate estimation of travel time requires the information from complex spatio-temporal dependencies of the road network. To enhance the capability in modeling the spatial dependencies, we design an adaptive self-attention network to capture the dependencies over different segments by jointly leveraging the global structural and local semantic of traffic patterns. Given a matrix whose rows correspond to a sequence of segment-view LSTM states, denoted by $H^s = [H^s_1, \dots, H^s_n] \in \mathbb{R}^{n \times d_s}$, we first project it into three matrices: queries Q, keys K, and values V according to Eq. (1). The representations of segments are learned by explicitly attending to all segments within the same link. Thus, the segment encoder naturally follows a global structural pattern over the entire link. Specifically, we define $GP_j \in \mathbb{R}^n$ to represent the similarity between the j-th segment and all segments in the same link as

$$GP_j = \frac{Q_j K^T}{\sqrt{d_s}}, \tag{3}$$

where $\sqrt{d_s}$ is the scaling factor.

However, the global structural pattern may endure problems in practical tasks. For example, the congestion propagation after traffic accidents may significantly affect the road conditions and cause similar congestion levels to the adjacent segments. To accurately evaluate the effect from neighboring segments, we then introduce a local semantic pattern LP_j as

$$LP_j(k) = \begin{cases} GP_j(k), & |j-k| \leq \omega \\ -\infty, & otherwise \end{cases}, \tag{4}$$

where ω is the single-sided window scale of neighboring segments, and thus the receptive field of the local segment pattern is $\varphi = 2\omega+1$; $LP_j(k)$ and $GP_j(k)$ represent the local and global similarities between the j-th and k-th segments. Conceptually, the strength of local pattern is set to the same as the global pattern when the k-th segment locates within the receptive field of the j-th segment, and is ignored otherwise. Using this strategy, both the upstream and downstream spatial dependencies can be captured.

By jointly considering the global and local patterns, the segment encoder is able to deal with the complex real-world traffic conditions within each link. To leverage both the global structural and local semantic patterns, we adopt a gating mechanism that adaptively aggregates them as

$$F^s_j = (1 - z_j) \odot Att(GP_j) + z_j \odot Att(LP_j), \tag{5}$$

where $\odot$ represents the element-wise product; $Att(GP_j)$ and $Att(LP_j)$ respectively represent the global and local segment attentions; $Att(\cdot)$ denotes the operation of softmax$(\cdot)V$ according to Eq. (2). The gating scalar z_j, conditioned on H^s_j, $Att(GP_j)$ and $Att(LP_j)$, is learned via

$$z_j = \sigma(W_h H^s_j + W_g Att(GP_j) + W_l Att(LP_j) + b_z), \tag{6}$$

where W_h, W_g, W_l and b_z are learnable parameters; $\sigma(\cdot)$ denotes the sigmoid activation. Then we employ a residual connection [16] and a layer normalization [2] to produce the segment-view context features $\{h^s_j\}^n_{j=1}$. By efficiently exploiting the global structural and local semantic patterns, the learned context features can well exhibit the rich information from the segment-view.

Note that the goal of segment encoder is to capture the spatio-temporal dependencies of segments in the same link and it is reasonable to expect that all links are endowed with a similar dependency structure. Therefore, the learnable parameters of the segment encoder are shared across all links and trained in parallel, which not only dramatically reduces the amount of parameters but also improves the computation efficiency of our HierETA model.

Joint Link-Intersection Encoder. Although the segment-view representation is widely used in existing works, it fails to model the consistency shared within the same link and hence the link-view consistency is discarded. Our hierarchical network solves this problem by introducing a coarse-scale representation as the complement of the segment-view representation. Besides, as links and intersections appear alternatively, we introduce a joint link-intersection encoder to seamlessly integrate these two views.

Let the learned context features of segments from the i-th link be $\{h^s_{ij}\}^n_{j=1}$, we apply the vanilla attention to represent the link-view feature as $x^l_i = \sum^n_{j=1} \gamma_{ij} h^s_{ij}$. Physically, it compresses the segment-view feature into a compact representation by exploring

[1] In the following of this subsection, we omit the link-view subscript i for concise description.

the importance of different segments. The weight coefficient γ_{ij} is calculated as $\gamma_{ij} = \text{softmax}_j(W_\gamma h^s_{ij} + b_\gamma)$, where W_γ and b_γ are the linear transformation matrix and bias term respectively.

Let the features of links and intersections be $\{x^l_i\}^m_{i=1}$ and $\{x^c_i\}^m_{i=1}$ respectively. Following the strategy in designing the segment encoder, we first employ two BiLSTMs to respectively encode the links and intersections, and represent the i-th hidden states of BiLSTMs as H^l_i and H^c_i respectively. Afterward, we combine these two features as $\hat{H}^l_i = [H^l_i | H^c_i]$. The integrated representation $\hat{H}^l_i$ therefore naturally reveals the intrinsic structure of trajectory on the road network where the link and intersection appear alternatively. To capture the spatio-temporal dependencies across different links and intersections, the joint link-intersection encoder also includes a self-attention layer, a residual connection and a layer normalization to obtain the joint context features $\{h^l_i\}^m_{i=1}$. Note that we eliminate the local pattern in the joint link-intersection encoder, as the traffic impacts between adjacent links are much weaker and sparser, and hence the model may be prone to overfitting.

Thanks to the novel hierarchical self-attention network, our HierETA model is able to simultaneously obtain the segment-view context feature that captures the local traffic conditions and the joint link-intersection context feature that preserves the common road attributes. The learned context features from three views work together to comprehensively model the underlying structure of trajectories on the road network such that the trajectories are well explored for travel time estimation.

4.3 Hierarchy-Aware Attention Decoder

In real scenarios, the spatio-temporal dependencies across different sub-routes are highly dynamic and correlated, and the uncertainty of travel time estimation is closely related to the critical components. For example, if a trajectory contains busy intersections or crowded segments, it is justified that more attention should be paid to these components. As such, treating the features of all sub-routes equally is unfair in practice. We hence propose a hierarchy-aware attention decoder to jointly leverage the multi-view context features. Formally, the final representation of a route $\mathcal{R}$ is defined as a combination of the context features as

$$\mathcal{R} = (1-\lambda)\sum_{i=1}^{m}\sum_{j=1}^{n}\alpha_{ij}h^s_{ij} + \lambda\sum_{i=1}^{m}\beta_i h^l_i, \tag{7}$$

where h^s_{ij} and h^l_i respectively represent the context feature of segment s_{ij} and joint link-intersection l_i; α_{ij} and β_i denote the scalar weights of the corresponding attentions; λ is the weighting parameter that linearly balances the fine- and coarse-scale features. Moreover, the driving conditions of a link during a limited period are often consistent owning to the relatively unchanged static road attributes. Thus, separately processing each segment without considering the link-view correlation is problematic as it lacks the feedback from the link-view consistency. To solve this issue, we design an attention guidance that adopts the link-view consistency to further adjust the segment-view attention. First, the scalar weight β_i of the link-view attention is set to

$$\beta_i = \underset{i}{\text{softmax}}(f^l(h^l_i, x^r)), \tag{8}$$

where $f^l(\cdot)$ incorporates both the link-view spatio-temporal features h^l_i and external impact factors x^r. It is formulated as

$$f^l(h^l_i, x^r) = v^T\tanh(w_1 h^l_i + w_2 x^r + b), \tag{9}$$

where v, w_1, w_2 and b are the trainable parameters. Then, the scalar weight α_{ij} of the segment-view attention guided by link-view consistency is updated as

$$\alpha_{ij} = \underset{(i,j)}{\text{softmax}}(\beta_i f^s(h^s_{ij}, x^r)). \tag{10}$$

Conceptually, employing the hierarchy-aware attention decoder, we can adaptively select the most relevant features from different representation granularities. Finally, a fully-connected layer is applied to the representation $\mathcal{R}$ for producing the travel time prediction $\hat{Y} \in \mathbb{R}^{N\times 1}$, where N denotes the total number of routes.

During training, HierETA is trained end-to-end via back propagation by minimizing the mean absolute error between the predicted value $\hat{Y}$ and the ground truth Y as

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum\nolimits_{k=1}^{N}\left|Y_k - \hat{Y}_k\right|, \tag{11}$$

where Θ denotes all learnable parameters in HierETA.

5 EXPERIMENTS

In this section, we examine the performance of HierETA on two large-scale datasets from Didi Chuxing. The sensitivity of key parameters and the ablation study are also provided for comprehensive understanding. The source codes are made publicly available.[2]

5.1 Datasets

We perform experiments on two real-world datasets collected in Beijing from Aug. 1st to 27th in 2020 and in Guangzhou from Jun. 1st to 30th in 2021. All GPS trajectories are mapped into the road network by utilizing the hidden markov map matching algorithm [30] to get the route attributes. In both experiments, we transform the trajectory data into segment sequences. The segments within two adjacent intersections are further grouped as a link. In our study, we aggregate the traffic data into 5-minutes intervals, which means there are 288 timeslots for one day. To avoid the artifacts caused by the abnormal cases in raw data, we remove the data with very short travel time (< 60s), extremely high travel speed (> 120km/h), and the number of segments in each link and the number of links in a route are restricted to 3-50 and 3-30 to fit the general cases. We extract the trips occupied by passengers as valid trajectories. Table 1 summarizes the description and statistics of the two datasets. Figure 3 depicts the travel time distributions on probability density functions (PDFs) and cumulative distribution functions (CDFs) of these two datasets.

5.2 Competitors

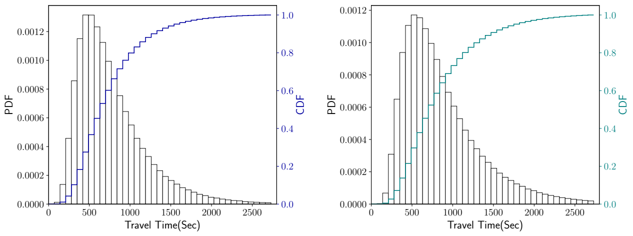
To demonstrate the effectiveness of HierETA, we compare it with the following competitors:

- **Route-ETA** simply sums up the historical average travel time at each road segment and the delay time at each intersection for the overall travel time prediction of a query trajectory.

[2] https://github.com/YuejiaoGong/HierETA

Table 1: The description and statistics of the datasets.

Dataset	Beijing	Guangzhou
training set	8.1-8.19 (2020)	6.1-6.22 (2021)
validation set	8.20 (2020)	6.23 (2021)
test set	8.21-8.27 (2020)	6.24-6.30 (2021)
number of trips	8, 170, 000	2, 570, 000
# unique drivers	200, 000	76, 000
# timeslots per day	288	288
travel time mean	892.98s	768.61s
travel distance mean	5.50km	5.16km
# avg. links per trip	9.04	8.35
# avg. segments per trip	96.10	94.94



(a) Beijing

(b) Guangzhou

Figure 3: The distributions of travel time (seconds) on Beijing and Guangzhou datasets.

- **MlpTTE** applies a 5-layer perceptron with ReLU activation [13] to estimate the travel time. We set the input of MlpTTE to be the same with that of HierETA and embed the discrete features into the same dimensions as HierETA. Note that MlpTTE cannot handle variable-length sequences directly. Here, we uniformly sample each trajectory to a fixed length of 128 segments.
- **DeepTravel** [49] captures different dynamics like driving state features, short-term and long-term traffic features for estimating the travel time of different grids, and introduces a dual interval loss, working as an auxiliary supervision, to fully leverage the temporal information of the trajectory data.
- **DeepTTE** [40] transforms trajectories into raw GPS sequences and utilizes geo-convolutional networks and recurrent neural networks to learn the complex spatio-temporal dependencies. It applies a multi-task learning component to estimate the travel time of the entire path and each local path simultaneously.
- **WDR** [42] formulates ETA as a regression problem and introduces a wide-deep-recurrent architecture to respectively handle high dimensional sparse features, the real value features and the temporal road segment features. In the experiments, we set the dimensions of hidden states in the recurrent and deep modules to 128.
- **DeepGTT** [26] develops a deep generative model for travel time distribution learning. It employs spatial smoothness embeddings and amortization to deal with the data sparsity dilemma. And a convolutional neural network based representation learning is utilized to capture real time traffic conditions.
- **ConSTGAT** [7] is a spatial-temporal graph neural network that adopts the graph attention mechanism to exploit the joint dependencies of spatial and temporal dynamics and applies convolutions to capture the contextual information of trajectories.
- **CoDriver ETA** [38] addresses the driver data sparsity problem by transferring knowledge from dense drivers to sparse ones under a multi-task learning framework. It adopts a triplet loss to measure the similarity between different drivers' driving preference.
- **TTPNet** [37] exploits a fast non-negative tensor decomposition algorithm to restore the missing travel speed and extracts the long-term ans short-term travel speed features via a CNN-RNN model. Then it portrays the representation of road network from historical trajectories based on graph embedding.

5.3 Experimental Settings

We initialize the weight parameters of HierETA via Xavier [12] and set the bias to zero. HierETA is trained over 50 epochs until convergence with a batch size of 256. The Adam optimizer [22] is utilized with a fixed learning rate of 1e-4 and a weight decay of 1e-5 as a regularization term to prevent over-fitting. For each competing method, we use the training set to train the model, select the model with the best MAPE on the validation set, and evaluate the performance using the test set. We repeat each experiment for five times except the statistics-based approach Route-ETA and report the mean and the standard deviation of different runs. The categorical external factors, i.e., *weekID*, *timeID*, *driverID*, are embedded into 3-, 5-, 16-dimensional spaces. The embedding dimensions of segment-view attributes, *segID* and *laneNum*, are 16 and 2. The intersection-view attribute *crossID* is projected into a 15-dimensional space. The segment number n and the link number m are set to 50 and 30 respectively. In the segment encoder, the dimension of the hidden states in bidirectional LSTM is 128. And the sizes of LSTM hidden states for encoding links and intersections are 192 and 64, respectively. For both segment- and joint link-intersection self-attentions, the parameter sizes are fixed to 256. All experiments are implemented in Python using Pytorch toolbox [31] with a NVIDIA RTX 3090Ti GPU with 24GB RAM. The platform runs on Ubuntu 16.04 OS with Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz.

We utilize four metrics for performance evaluation, including mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), and satisfaction rate (SR), similar to existing approaches [23]. Specifically, SR refers to the fraction of trips with error rates less than 10% and a higher SR indicates better performance and customer satisfaction. Their definitions are as follows:
Mean Absolute Error (MAE),

$$MAE = \frac{1}{N}\sum_{k=1}^{N}|y_k - \hat{y}_k|$$

Table 2: Overall performance comparison of HierETA and the competitors on Beijing and Guangzhou datasets. All the results are better with smaller values except the SR metric.

Dataset	Beijing				Guangzhou			
Metrics	*MAE(s)*	*RMSE(s)*	*MAPE(%)*	*SR(%)*	*MAE(s)*	*RMSE(s)*	*MAPE(%)*	*SR(%)*
Route-ETA	159.85	254.39	17.815	37.575	142.74	210.70	17.611	33.101
MlpTTE	134.48±1.67	228.52±6.54	14.678±0.19	44.248±1.03	126.33±0.98	212.62±5.68	15.765±0.74	42.244±1.46
DeepTravel	123.28±1.09	189.09±0.35	14.442±3.99	44.322±0.94	115.20±0.76	181.10±3.97	14.498±0.20	42.937±0.55
DeepTTE	112.77±0.87	172.83±2.12	12.816±0.04	47.644±0.35	102.50±0.60	163.77±1.40	13.688±0.21	47.639±0.27
WDR	107.46±1.13	165.06±1.78	12.549±0.16	49.786±0.48	98.55±0.96	162.80±2.26	12.526±0.19	50.009±0.32
DeepGTT	118.33±1.04	184.69±1.15	13.612±0.42	46.734±0.65	107.68±0.85	170.08±3.27	14.120±0.21	45.908±0.51
ConSTGAT	110.43±0.76	169.92±1.05	12.703±0.14	49.106±0.07	102.52±0.94	165.88±1.77	13.050±0.16	48.063±0.36
CoDriver ETA	106.62±0.74	167.06±2.71	12.125±0.07	50.624±0.08	97.78±0.97	160.12±2.69	12.511±0.13	49.974±0.43
TTPNet	104.91±0.67	163.25±1.41	12.004±0.11	51.524±0.36	97.96±0.70	156.92±1.76	12.802±0.08	49.688±0.37
HierETA	**99.61±0.66**	**153.62±1.20**	**11.673±0.12**	**53.153±0.22**	**94.62±0.58**	**149.64±2.17**	**12.275±0.10**	**51.339±0.17**

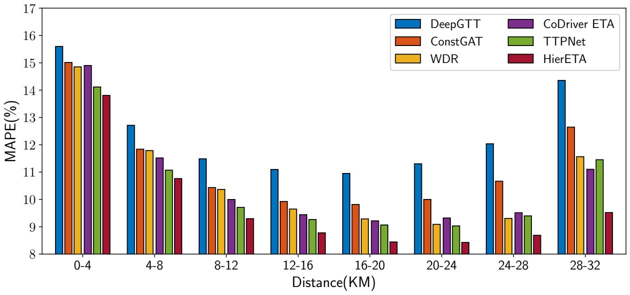


Figure 4: Performance of HierETA and its competitors on the MAPE metric for trips with varying origin-destination distances.

Root Mean Squared Error (RMSE),

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2}$$

Mean Absolute Percentage Error (MAPE),

$$MAPE = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{y_k - \hat{y}_k}{y_k}\right| \times 100\%$$

and Satisfaction Rate (SR),

$$SR = \frac{1}{N}\sum_{k=1}^{N}\left(\left|\frac{y_k - \hat{y}_k}{y_k}\right| \leq 10\%\right) \times 100\%$$

where y_k and $\hat{y}_k$ denote the ground truth and the predicted value, respectively.

5.4 Experimental Results and Analysis

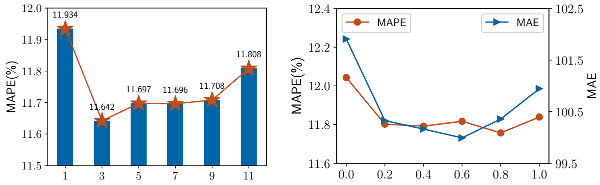
Table 2 presents the comparison between HierETA and its competitors. The naive baselines Route-ETA simply utilizes the historical traffic speed, and MlpTTE possesses a multilayer perceptron that fails to characterize the valued spatio-temporal information. As such, these two methods produce unsatisfactory results. DeepTravel performs better as it considers the contextual information, while only applying BiLSTM is incapable of modeling the complex spatial dependency of traffic data. Other competitors show advanced performance. For example, ConstGAT considers the graph structures of the road network to exploit the joint relations of spatio-temporal information. And TTPNet restores the valued historical traffic speed to alleviate the intractable data sparsity problem and utilizes graph embedding to represent the road network structure. Compared to the best-performing competitors, our model HierETA shows significantly better performance on all evaluation metrics. On the Beijing dataset, HierETA outperforms the best competitor TTPNet by reducing the MAE from 104.91 to 99.61 seconds on MAE. On the Guangzhou dataset, our model also surpasses TTPNet by 3.34 seconds on MAE and improves the satisfaction rate from 49.69% to 51.34%, achieving 3.32% relative improvement.

We also visualize the prediction error of HierETA with five best-performing competitors on the Beijing dataset by considering trips with varying distances. To this end, we group the trips in test set into subgroups by their lengths in 5km step, and study the performance of different models on these subgroups. As shown in Figure 4, HierETA obtains significant advantages in all scenarios, especially in the case of long trajectories, more obvious performance improvements have been achieved. That is, interpreting the trajectory from multiple views effectively portrays the hierarchical structure of road network and eases the error propagation for estimating the travel time.

In general, HierETA is effective for explicitly learning both structural and semantic traffic characteristics using the hierarchical self-attention network and multi-view trajectory representation. By hierarchically modeling the multi-view trajectory features, our model is able to learn long-range spatio-temporal dependencies from different granularities. Compared with the single-view segment representation, our hierarchical structure is more interpretable and capable of modeling the underlying road network structure.

5.5 Model Analysis

We also analysize the key model parameters and conduct ablation study to provide a comprehensive understanding of HierETA.



(a) window sizes φ

(b) coefficients λ

Figure 5: Ablation study of HierETA w.r.t different (a) window sizes φ of local attention and (b) coefficients λ. We set the default values as $\varphi = 3$ and $\lambda = 0.4$, respectively.

Table 3: Ablation results of HierETA and its variants on Beijing and Guangzhou datasets.

Dataset	Beijing		Guangzhou	
Metrics	*MAE(s)*	*MAPE(%)*	*MAE(s)*	*MAPE(%)*
HierETA	**99.61**	**11.673**	**94.62**	**12.275**
w/o lcoal@e	100.64	11.909	95.08	12.455
w/o global@e	102.51	11.716	96.72	12.734
w/o guide@d	101.90	12.043	97.40	12.447
w/o cross info.	101.69	11.992	97.03	12.545
w/o hier.	103.55	12.243	98.90	13.041

5.5.1 *Parameter Analysis.*

To examine the performance of HierETA over different settings of parameters, we first analyze the impact of the local segment attention under different window sizes φ on the Beijing dataset. As plotted in Figure 5(a), compared to the setting that no adjacent segment is explored, i.e., $\varphi = 1$, HierETA generally achieves better performance by introducing neighboring segments, i.e., $\varphi > 1$, and the best result is obtained when $\varphi = 3$. That is, the local segment attention shows the best empirical performance when only the nearest segments are considered. The performance gradually decreases with the increase of window sizes φ. This observation may come from the fact that, when the attention scope is enriched, the correlation between adjacent segments slightly decreases while the modeling uncertainty increases.

We also evaluate the importance of leveraging the multi-view features by varying λ from 0 to 1, the results are recorded in Figure 5(b). We find that high error rates are observed when only the segment- or link-intersection attention is applied (λ equals to 0 or 1 respectively). And HierETA achieves consistently satisfactory results when λ locates within $[0.2, 0.8]$, clearly revealing the robustness when jointly leveraging the multi-view context features.

5.5.2 *Ablation Study.*

We provide the ablation tests to examine the effectiveness of key modules in HierETA. Five variants are considered by individually eliminating the local (w/o local@e) or global (w/o global@e) attentions in the segment encoder, the attention guidance in the hierarchy-aware attention decoder (w/o guide@d), the intersection information (w/o cross info.), and the hierarchical structure (w/o hier.).

- **w/o lcoal@e**: The local attention in encoder is removed to verify the effectiveness for modeling the semantic traffic condition.
- **w/o global@e**: This model does not consider the global attention to verify the necessity of extracting the structural traffic pattern.
- **w/o guide@d**: The attention guidance assisting in decoder is removed, lacking the feedback from the link-view consistency as in Eq. (10).
- **w/o cross info.**: Only segment-view and link-view representations of trajectory are applied for modeling the spatio-temporal dependency.
- **w/o hier.**: In this case, we eliminate the hierarchical structure by removing the joint link-intersection encoder.

Table 3 presents the experimental results with HierETA as a comparison. We find that HierETA consistently outperforms all variants, indicating the importance of these modules. The result shows that HierETA performs better than both variants that eliminating local and global attentions, which is contributed to the introduction of the global structural and local semantic patterns. The performance of variant without attention guidance is also inferior to that of HierETA, as it is incapable of balancing the segment, link, and intersection features simultaneously. Note that significant declines are witnessed without the hierarchical structure. Specifically, after introducing the hierarchical representation, HierETA decreases the MAE from 103.55 to 99.61 and 98.90 to 94.62 on Beijing and Guangzhou datasets, with a relative improvement of 3.8% and 4.3% respectively. This validates the great benefits of explicitly representing trajectory on the road network using multi-view representation learning and hierarchically organizing the segment-, link-, and intersection-views for the ETA task.

6 CONCLUSION

In this work, we propose to comprehensively interpret trajectories from the segment-, link-, and intersection-views and accordingly design a novel hierarchical self-attention network HierETA to learn the underlying structure of trajectories on the road network. Taking the complex real-world traffic conditions into consideration, HierETA adaptively integrates the global and local patterns for spatio-temporal dependency modeling within the multi-view representation framework. In addition, we devise a hierarchy-aware attention decoder that adaptively balances the importance of multi-view context features. Experiments on two large-scale real-world datasets from Didi Chuxing show that HierETA achieves the state-of-the-art performance. Moreover, the novel multi-view trajectory representation has shown great promise in practice, and hence opens up new opportunities in developing advanced trajectory data mining models.

ACKNOWLEDGMENTS

This work was supported in part by the DiDi GAIA Research Fund, in part by the Guangdong Natural Science Funds for Distinguished Young Scholars under Grant 2022B1515020049, in part by the National Natural Science Foundation of China under Grant 62006080, in part by the Guangdong Regional Joint Fund for Basic and Applied Research under Grant 2021B1515120078, and in part by the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Avinash Achar, Venkatesh Sarangan, Rohith Regikumar, and Anand Sivasubramaniam. 2018. Predicting vehicular travel times by modeling heterogeneous influences between arterial roads. In *Proc. Am. Assoc. Artif. Intell.* 2063–2070.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[3] Xu Chen, Junshan Wang, and Kunqing Xie. 2021. TrafficStream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. In *Proc. Int. Joint Conf. Artif. Intell.* 3620–3626.

[4] Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. 2020. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 3074–3082.

[5] Corrado De Fabritiis, Roberto Ragona, and Gaetano Valenti. 2008. Traffic estimation and prediction based on real time floating car data. *IEEE Trans. Intell. Transp. Syst.* (2008), 197–203.

[6] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. 2021. SSML: Self-supervised meta-learner for en route travel time estimation at Baidu Maps. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 2840–2848.

[7] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. 2020. ConSTGAT: Contextual spatial-temporal graph attention network for travel time estimation at Baidu maps. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 2697–2705.

[8] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 364–373.

[9] Kun Fu, Fanlin Meng, Jieping Ye, and Zheng Wang. 2020. CompactETA: A fast inference system for travel time prediction. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 3337–3345.

[10] Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proc. Adv. Neural Inf. Process. Syst.* 1019–1027.

[11] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proc. Am. Assoc. Artif. Intell.* 3656–3663.

[12] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proc. Int. Conf. Artif. Intell. Stat.* 249–256.

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proc. Int. Conf. Artif. Intell. Stat.* JMLR Workshop and Conference Proceedings, 315–323.

[14] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. 2021. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 547–555.

[15] Peng Han, Jin Wang, Di Yao, Shuo Shang, and Xiangliang Zhang. 2021. A graph-based approach for trajectory similarity computation in spatial networks. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 556–564.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 770–778.

[17] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* (1997), 1735–1780.

[18] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. 2020. HetETA: Heterogeneous information network embedding for estimating time of arrival. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 2444–2454.

[19] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, and Weiyang Kong. 2020. LSGCN: Long short-term traffic prediction with graph convolutional networks. In *Proc. Int. Joint Conf. Artif. Intell.* 2355–2361.

[20] Bo Hui, Da Yan, Haiquan Chen, and Wei-Shinn Ku. 2021. TrajNet: A trajectory-based deep learning model for traffic prediction. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 716–724.

[21] Erik Jenelius and Haris N Koutsopoulos. 2013. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transp. Res. B: Methodol.* (2013), 64–81.

[22] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[23] Wuwei Lan, Yanyan Xu, and Bin Zhao. 2019. Travel time estimation without road networks: An urban morphological layout representation approach. In *Proc. Int. Joint Conf. Artif. Intell.* 1772–1778.

[24] Ke Li, Lisi Chen, Shuo Shang, Panos Kalnis, and Bin Yao. 2021. Traffic congestion alleviation over dynamic road networks: Continuous optimal route combination for trip query streams. In *Proc. Int. Joint Conf. Artif. Intell.* 3656–3662.

[25] Mingqian Li, Panrong Tong, Mo Li, Zhongming Jin, Jianqiang Huang, and Xian-Sheng Hua. 2021. Traffic flow prediction with vehicle trajectories. In *Proc. Am. Assoc. Artif. Intell.* 294–302.

[26] Xiucheng Li, Gao Cong, Aixin Sun, and Yun Cheng. 2019. Learning travel time distributions with deep generative model. In *WWW.* 1017–1027.

[27] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 1695–1704.

[28] Haoxing Lin, Rufan Bai, Weijia Jia, Xinyu Yang, and Yongjian You. 2020. Preserving dynamic attention for long-term spatial-temporal prediction. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 36–46.

[29] Hao Liu, Ying Li, Yanjie Fu, Huaibo Mei, Jingbo Zhou, Xu Ma, and Hui Xiong. 2020. Polestar: An intelligent, efficient and national-wide public transportation routing engine. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 2321–2329.

[30] Paul Newson and John Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.* 336–343.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proc. Adv. Neural Inf. Process. Syst.* 8026–8037.

[32] Huiling Qin, Xianyuan Zhan, Yuanxun Li, Xiaodu Yang, and Yu Zheng. 2021. Network-wide traffic states imputation using self-interested coalitional learning. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 1370–1378.

[33] Mahmood Rahmani, Erik Jenelius, and Haris N Koutsopoulos. 2013. Route travel time estimation using low-frequency floating car data. In *IEEE Intell. Transp. Syst. Conf.* 2292–2297.

[34] Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. 2021. MTrajRec: Map-constrained trajectory recovery via seq2seq multi-task learning. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 1410–1419.

[35] Stefano Giovanni Rizzo, Giovanna Vantini, and Sanjay Chawla. 2019. Time critic policy gradient methods for traffic signal control in complex and congested scenarios. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 1654–1664.

[36] Sijie Ruan, Cheng Long, Jie Bao, Chunyang Li, Zisheng Yu, Ruiyuan Li, Yuxuan Liang, Tianfu He, and Yu Zheng. 2020. Learning to generate maps from trajectories. In *Proc. Am. Assoc. Artif. Intell.* 890–897.

[37] Yibin Shen, Cheqing Jin, and Jiaxun Hua. 2020. TTPNet: A neural network for travel time prediction based on tensor decomposition and graph embedding. *IEEE Trans. Knowl. Data Eng.* (2020).

[38] Yiwen Sun, Kun Fu, Zheng Wang, Donghua Zhou, Kailun Wu, Jieping Ye, and Changshui Zhang. 2020. CoDriver ETA: Combine driver information in estimated time of arrival by driving style learning auxiliary task. *IEEE Trans. Intell. Transp. Syst.* (2020).

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. Adv. Neural Inf. Process. Syst.* 5998–6008.

[40] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. 2018. When will you arrive? estimating travel time based on deep neural networks. In *Proc. Am. Assoc. Artif. Intell.* 2500–2507.

[41] Hongjian Wang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. 2016. A simple baseline for travel time estimation using large-scale trip data. In *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.* 1–4.

[42] Zheng Wang, Kun Fu, and Jieping Ye. 2018. Learning to estimate the travel time. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 858–866.

[43] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. 2004. Travel time prediction with support vector regression. *IEEE Trans. Intell. Transp. Syst.* (2004), 276–281.

[44] Tong Xia, Yunhan Qi, Jie Feng, Fengli Xu, Funing Sun, Diansheng Guo, and Yong Li. 2021. AttnMove: History enhanced trajectory recovery via attentional network. In *Proc. Am. Assoc. Artif. Intell.* 4494–4502.

[45] Zhe Xu, Zhixin Li, Qingwen Guan, Dingshui Zhang, and Jieping Ye. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 905–913.

[46] Bin Yang, Chenjuan Guo, and Christian S Jensen. 2013. Travel cost inference from sparse, spatio temporally correlated time series using markov models. *Proc. VLDB Endow.* (2013), 769–780.

[47] Zhengxu Yu, Shuxian Liang, Long Wei, Zhongming Jin, Jianqiang Huang, Deng Cai, Xiaofei He, and Xian-Sheng Hua. 2021. MaCAR: Urban traffic light control via active multi-agent communication and action rectification. In *Proc. Int. Joint Conf. Artif. Intell.* 2491–2497.

[48] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. T-drive: Enhancing driving directions with taxi drivers’ intelligence. *IEEE Trans. Knowl. Data Eng.* (2011), 220–232.

[49] Hanyuan Zhang, Hao Wu, Weiwei Sun, and Baihua Zheng. 2018. DeepTravel: A neural network based travel time estimation model with auxiliary supervision. In *Proc. Int. Joint Conf. Artif. Intell.* 3655–3661.

[50] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A graph multi-attention network for traffic prediction. In *Proc. Am. Assoc. Artif. Intell.* 1234–1241.

[51] Yu Zheng. 2015. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.* (2015), 1–41.