Alleviating Data Sparsity Problems in Estimated Time of Arrival via Auxiliary Metric Learning

Yiwen Sun, *Graduate Student Member, IEEE*, Wenzheng Hu, Donghua Zhou, *Fellow, IEEE*, Baichuan Mo, Kun Fu, *Member, IEEE*, Zhengping Che, *Member, IEEE*, Zheng Wang, *Member, IEEE*, Shenhao Wang, Jinhua Zhao, Jieping Ye, Jian Tang, and Changshui Zhang, *Fellow, IEEE*

***Abstract*— With millions of people using ride-hailing platforms for daily travel, estimated time of arrival (ETA) has become a significant problem in intelligent transportation systems and attracted considerable attention recently. Deep learning-based ETA methods have achieved promising results using massive spatial-temporal data. However, we find that the prediction accuracy is not satisfactory in practical applications due to the prevalent data sparsity problems. Instead of focusing on the average prediction performance as many other methods, this study aims to alleviate the data sparsity problems in ETA to enhance user experience. In general, the data sparsity problems arise from two aspects. The first is the road network, where many links are only traversed by few floating cars. The second aspect is drivers, where many drivers' trajectories are too scarce (e.g., with only 3 trip records). To alleviate the sparsity in road network, we propose a Road Network Metric Learning framework for ETA (*RNML-ETA*), where an auxiliary metric learning task is used to improve the link-embedding, especially for links with insufficient data. A novel triangle loss is proposed to improve metric learning effectiveness for links. Experiments on massive real-world data show that *RNML-ETA* outperforms competing methods by promoting the cold links with limited data. Furthermore, we propose a novel unified framework to Alleviate Data Sparsity problems in ETA (*ADS-ETA*) by extending *RNML-ETA* with an additional auxiliary task for driver ID embedding. Results with extensive experiments demonstrate that *ADS-ETA* can effectively alleviate the data sparsity problems caused by road network and driver sparsity.**

***Index Terms*— Estimated time of arrival, data sparsity problem, metric learning, multi-task learning.**

Manuscript received 10 March 2021; revised 10 December 2021 and 10 June 2022; accepted 20 July 2022. This work was supported by the National Natural Science Foundation of China under Grant 62033008 and Grant 62061136001. The Associate Editor for this article was J. W. Choi. *(Corresponding author: Changshui Zhang.)*

Yiwen Sun and Changshui Zhang are with the Institute for Artificial Intelligence, Tsinghua University (THUAI), State Key Lab of Intelligent Technologies and Systems, Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: syw17@mails.tsinghua.edu.cn; zcs@mail.tsinghua.edu.cn).

Wenzheng Hu is with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 100084, China, and also with Kuaishou Technology Company Ltd., Beijing 100084, China (e-mail: huwenzheng@kuaishou.com).

Donghua Zhou is with the College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China, and also with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: zdh@mail.tsinghua.edu.cn).

Baichuan Mo is with the Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: baichuan@mit.edu).

Kun Fu and Zheng Wang are with Beike AI Tech, Beijing 100089, China (e-mail: fukun009@ke.com; wangzheng087@ke.com).

Zhengping Che and Jian Tang are with the Midea Group, Beijing 100102, China (e-mail: chezp@midea.com; tangjian22@midea.com).

Shenhao Wang is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA, and also with the Department of Urban and Regional Planning, University of Florida, Gainesville, FL 32611 USA (e-mail: shenhao@mit.edu).

Jinhua Zhao is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jinhua@mit.edu).

Jieping Ye is with Beike AI Tech, Beijing 100089, China, and also with the Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: yejieping@ke.com).

Digital Object Identifier 10.1109/TITS.2022.3200445

I. Introduction

INTELLIGENT transportation systems (ITS) are devoted to explore better transportation options for human beings and better relationships among users, vehicles and transportation infrastructures [1], [2], [3]. Estimated time of arrival (ETA) is one of the most fundamental and challenging problems in ITS [1], [2], [3]. ETA means predicting the travel time from an origin location to a destination location along a given route. A real case of ETA is illustrated in the left part of Fig. 1. ETA models enable transportation systems to efficiently schedule vehicles and reduce urban traffic congestion [4]. In recent years, with the rapid growth of ride-hailing companies such as Uber and DiDi, ETA has attracted more attention. For example, route planning, navigation, carpooling, vehicle dispatching, and scheduling in ride-hailing platforms rely heavily on the ETA system [5], [6]. Therefore, an accurate and efficient ETA system is vital for improving the platforms' operating performance and enhancing users' experience.

Existing ETA methods can be roughly divided into two categories. The first is the additive method. Generally, these methods split a route into several links and explicitly predict the travel time for each road segment and each intersection by adopting different methods [7], [8], [9], [10]. The total travel time of a route is then calculated by assembling different ingredients' travel time. These methods are intuitive and interpretable, but the accumulation of local errors may easily result in inaccurate predictions. The second is the global method that formulates ETA as a regression problem and estimates the overall travel time directly. The methods based on deep learning [5], [6], [11], [12] can effectively capture the

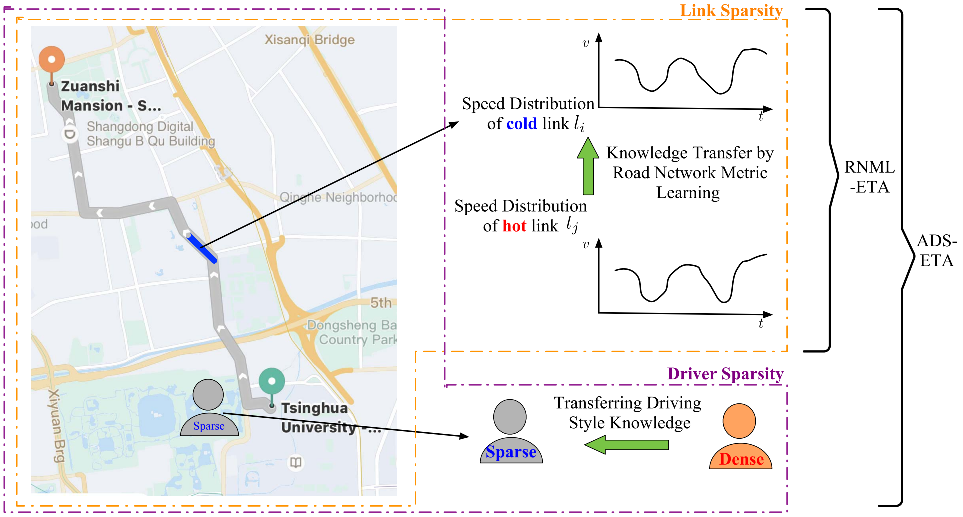

Link Sparsity
Xisanqi Bridge
Zuanshi Mansion - S...
Shangdong Digital Shangu B Qu Building
Qinghe Neighborhoo
5th
Dongsheng Ba Country Park
Xiyuan Brg
Tsinghua University - ...
Sparse
Speed Distribution of cold link l_i
v
t
Knowledge Transfer by Road Network Metric Learning
Speed Distribution of hot link l_j
RNML -ETA
ADS-ETA
Driver Sparsity
Transferring Driving Style Knowledge
Sparse
Dense

Fig. 1. Conceptual demonstration of *RNML-ETA* and *ADS-ETA*. The left part shows a real-world example in which the ETA system predicts the travel time along the route starting from the green pin to the orange pin. The route consists of a sequence of links. To alleviate the road network data sparsity problem, *RNML-ETA* transfers the knowledge of hot links to the cold links by metric learning. The links' similarity is measured using their speed distribution. When, unfortunately, the driver data is also sparse, *ADS-ETA* proposes to also transfer the knowledge from dense drivers to sparse drivers sharing similar driving styles so as to alleviate these two data sparsity problems simultaneously. The auxiliary task for improving the quality of driver embedding is with reference to [13].

spatial-temporal dependencies in large-scale data and achieve remarkable prediction performance. As a leading example, the Wide-Deep-Recurrent model (WDR) [5] jointly trains the wide, the deep, and the recurrent models to predict the travel time based on a rich set of input features in real-world applications. This kind of method avoids the local error accumulation. For deep learning based methods, feature extraction is needed from the raw trajectory data. Input features for an ETA model can be categorized as sequential (dynamic features) and non-sequential features (static features). In this paper, (1) sequential features consist of link ID, link length, and the estimated link travel speed (road condition); (2) non-sequential features include the time of the day, driver ID, and the day of the week.

In terms of the average estimation accuracy, deep learning-based methods, such as WDR, have made tremendous progress and perform effectively in practices [5]. Nevertheless, the accuracy is not satisfactory when there are data sparsity problems, which commonly exist in many real-world applications but have not been solved in the existing literature. In the online ride-hailing application, increasing prediction performance under data sparsity can enhance the experience of passengers evidently. Data sparsity problems in ETA mainly involve the following two aspects:

1) *Driver Data Sparsity*. Personalized driver information is one of the key features for ETA. It has been experimentally validated that embedding driver ID (i.e., personalized information) can significantly improve the precision of ETA [13]. However, since many drivers are part-time or new, their trajectory data is scarce, resulting in poor performance in forecasting [13];
2) *Road Network Data Sparsity*. Though ride-hailing platforms collect millions of trajectories per day in the real world, many links (i.e., road segments) still have a limited number of passing cars. This sparsity may result in an inaccurate estimation of travel time.

In a deep learning-based ETA system, the road network information is usually captured by the link-embedding vectors [14], [15], [16], where each link-embedding vector effectively encodes the semantic information through iterative training under a premise of sufficient data. We name links covered by few (or even zero in very rare cases) trajectories and sufficient trajectories as cold links and hot links, respectively. In the current deep learning-based ETA system, embedding vectors of cold links are easy to be under-fitting. Thus, for a trajectory that contains many cold links, the ETA prediction accuracy will drop significantly.

In this paper, we present a novel *RNML-ETA* to alleviate the road network sparsity problem. *RNML-ETA* adopts the multi-task learning (MTL) framework [17] where the link embedding is shared and learned jointly. For the main task, it refers to the Wide-Deep-Recurrent (WDR) [5] to estimate the travel time. What is different is that the auxiliary task is a specially designed road network metric learning which transfers the knowledge of hot links' patterns into cold links. We also propose a novel metric learning loss–triangle loss–specially designed for the links, which learns more precise positional relationships among links in the embedded space. *RNML-ETA* measures the similarity of different links according to the speed distribution across different times. Through this metric, links with similar traffic conditions are gathered closer while the dissimilar links are separated in the embedded space. Thus, embedding vectors of cold links are effectively improved with information from other similar hot links. Our method can effectively make up for the information lack of cold links and predict more accurately in trajectories with many cold links. The top half of Fig. 1 shows a conceptual demonstration of *RNML-ETA*. Furthermore, we explore a more difficult situation where the road network and driver sparsity occur simultaneously, extending *RNML-ETA* and proposing a framework named Alleviating Data Sparsity problems in ETA (*ADS-ETA*) to deal with such two data sparsity problems in a unified way. The conceptual demonstration of *ADS-ETA* is shown in Fig. 1. Its core idea is a simultaneous knowledge transfer for both drivers and links to alleviate the corresponding data sparsity problems. The superiority of *ADS-ETA* is demonstrated through the extensive experiments across the whole datasets and with driver and road network sparsities.

This paper is an extension of our prior conference paper [18]. There are additional improvements both in the method and experiment aspects. The substantively novel improvement are summarized as follows. (1) We conduct an extensive experiment, *RNML-ETA* with our triangle loss v.s. *RNML-ETA* with common triplet loss [19], to show the effectiveness and fast convergence of the proposed triangle loss function. (2) We extend the road network sparsity problem to unified data sparsity problems containing road network and driver sparsities. Furthermore, we propose a framework named *ADS-ETA* to alleviate the unified data sparsity problems in ETA. Extensive experiments on the data from the DiDi platform are conducted. The experimental results demonstrate that *ADS-ETA* further outperforms *RNML-ETA* significantly when the road network and driver sparsity problems occur simultaneously.

In this work, we make the following contributions (including our prior conference paper) as follows:

- To the best of our knowledge, *RNML-ETA* is the first deep learning method that effectively alleviates the data sparsity problem of the road network, and *ADS-ETA* is the first deep learning method that effectively addresses the data sparsity problems of ETA containing both road network and driver sparsity.
- We propose a novel metric learning framework to improve the quality of link-embedding vectors. We utilize the link traffic speed distribution across different time bins to construct the link difference matrix to define the link similarity. The novel triangle loss is designed specifically for improving the effectiveness of road network metric learning.
- We evaluated both *RNML-ETA* and *ADS-ETA* on massive real-world datasets with over 100 million trajectories. Experiments demonstrate that both *RNML-ETA* and *ADS-ETA* have significantly better prediction performance when two data sparsity problems occur compared with the competing methods.

We organize the rest of this article as follows. In Section II, we review the related works of ETA and metric learning. In Section III, we present the detailed descriptions of *RNML-ETA* and *ADS-ETA*. In Section IV, we analyze the experimental results on the large-scale vehicle travel datasets from the DiDi platform. In Section V, we conclude this paper and discuss the future work.

II. Related Work

A. Estimated Time of Arrival

As one of ITS's crucial tasks, ETA attracts widespread interest in both academic and industrial communities. Except for a few works [20], [21], [22] focusing on the prediction of ETA distribution, the studies of ETA are dedicated to the precise estimation of the individual time value. The methods of solving the ETA problem can be roughly divided into two categories.

The first category is the additive methods. These methods focus on predicting each link's travel time and the delay time at intersections that the route passes. Then, all the time of links and intersections is summed over to obtain the final ETA result for the trajectory. Methods for predicting the time of ingredients are rule-based methods or machine learning-based ones. A simple rule-based method is dividing the length of the link by real-time traffic speed provided by a traffic monitoring service. Although it is difficult to achieve accurate results in dynamic traffic systems, this method is popular in the industry because of its simplicity and fast inference. Various machine learning-based methods, such as dynamic Bayesian network [8], least-square minimization [23], pattern matching [9], gradient boosted regression tree [10] are adopted to capture spatial-temporal dependencies to estimate the time of ingredients. These methods obtain more accurate ETA results than rule-based ones. However, there is no specific strategy dealing with data sparsity problems. Wang *et al.* [7] discuss the road network sparsity problem that a part of links are traversed by too few trajectories. PTTE [7] is proposed to represent the trips as a tensor and utilize tensor decomposition to complete the missing values. However, alleviating the road network sparsity problem is still a challenging problem for ETA, and PTTE is not skilled in taking advantage of big data. As far as we know, before our previous conference paper [18], there is no method that can simultaneously leverage deep learning to capture spatial-temporal dependencies given massive data and effectively alleviate the road network data sparsity problem.

The second category is the global methods. These methods take the ETA problem as a whole and directly estimate the overall travel time given the route. Relatively early approaches use traditional machine learning-based methods to mine the spatial-temporal correlations for predicting ETA. For instance, TEMP [24] is a simple ETA method based on nearest neighbor as well as cyclical traffic conditions without the given route information. Yuan *et al.* [20] construct a time-dependent landmark graph to model driver intelligence, and the travel time distribution between two landmarks is predicted by variance-entropy-based clustering. Recently, due to the bloom of deep learning [11], [25], [26], many scholars apply deep neural network to the field of ITS, such as traffic forecasting [27], [28], [29], [30], [31]. Besides, several deep learning-based ETA methods are also put forward from different perspectives. Li *et al.* [12] also investigate the origin-destination ETA problem which is similar to [24] by proposing MURAT [12]. To reduce the accuracy gap, they adopt the deep residual network with graph embedding to preserve underlying road network information. Wang *et al.* [6] propose a Geo-Conv layer for transforming the longitude and latitude of raw GPS location points to feature maps encoding the local spatial correlations. Then the standard LSTM [32] is used to learn the sequential dependencies of the feature maps. Before the vehicle's departure, the GPS points cannot be obtained, which leads to limitations in real applications. At the inference stage, generating appropriate pseudo-GPS points given a planned route becomes a challenging issue. Wang *et al.* [5] present a Wide-Deep-Recurrent (WDR) model which jointly trains wide linear model, deep neural network, and LSTM to make good use of non-sequential features and sequential features purposefully. The wide module and the deep module can effectively and comprehensively extract non-sequential features from two angles. However, the above two modules are not adept at capturing the local information of sequential features corresponding to each link [5]. Thus, Wang *et al.* [5] add the recurrent module to deal with sequential features. The recurrent module is a standard LSTM and completes the task of learning the sequential dependencies between different links. Applying appropriate modules to different features is the key to achieving satisfactory ETA prediction performance. WDR is selected as the primary baseline in this paper for it is one of the state-of-the-art methods and can be deployed on the platform. The authors of [33], [34] transform the spatial information into the image sequence and adopt a convolutional neural network to mine spatial correlations for ETA. Besides the prediction accuracy, some works [35], [36] recently begin to focus on the inference speed of deep learning-based ETA systems for ensuring the efficiency of practical applications. Most deep learning-based ETA methods adopt the embedding technique to represent the geographical elements of the road network, such as the link embedding in [5], [12], and the grid embedding in [37]. The embedding of geographical elements suffers from the road network sparsity problem even in big data. We propose *RNML-ETA* to alleviate this problem purposefully. The driving style information is also usually added to the ETA system through the embedding vector of driverID [5], [6], [12] and the driver data sparsity problem is discussed and alleviated in [13]. An interesting and challenging problem is how to effectively deal with these two sparsity problems in the ETA system concurrently? A related research subfield is the data imputing on the traffic flow prediction task [38], [39], [40]. Tan *et al.* [38] introduce the tensor pattern in order to model the traffic data and present a tensor decomposition-based method for imputation. Li *et al.* [39] extend the probabilistic principal component analysis based imputing method through considering temporal as well as spatial dependence appropriately. The data sparsity problems in ETA are different from the scenario of missing values on the traffic flow prediction task. Therefore, the novel and systematic method, proposed *ADS-ETA* has some degree of research significance.

B. Metric Learning

Metric learning aims to learn a representation function that maps objects into an embedded space where the distance could preserve the samples' similarity. Intuitively speaking, similar samples get close and dissimilar samples get far away. Early methods adopt kernel approaches as a bridge to allow the linear projection to have access to dealing with non-linear characteristics of real-world problems [41]. Deep network-based metric learning methods with activation functions are proved to be more effective in recent years [41] and achieve great success in many fields, especially in computer vision. Loss functions that are also known as objective functions are essential for metric learning, and various loss functions are proposed. For example, the contrastive loss [42] guides the objects from the same class to be mapped to the same point and those from different classes to be mapped to different points whose distances are larger than a margin. Triplet loss [19] requires the distance between the anchor sample and the positive sample to be smaller than the distance between the anchor sample and the negative sample by a small margin. Triplet loss is famous for its success in face recognition and clustering. The case with one positive sample and multiple negative samples is extended in [43]. Metric learning often suffers from slow convergence, partially because the loss only captures limited interaction in one update. For instance, in one update of triplet loss, it is meaningless whether the distance between negative and positive is larger than the distance between negative and anchor. In this paper, for metric learning of links, we make good use of the characteristic that the speed distribution similarity distance of any two links can be meaningfully measured to realize more interactions and faster convergence.

III. Methodology

A road network consists of a set of links $\{l = 1, 2, \cdots, M\}$, where M is the total number of links in the network, and l is the link ID ranging from 1 to M. A trajectory is a path composed of a series of links connected end to end. The definition of ETA learning is introduced as follows.

Definition 1 (ETA Learning): Suppose we have a collection of historical trajectories $\{s_i, e_i, d_i, \boldsymbol{p}_i\}_{i=1}^N$, where N stands for the total number of trajectories, s_i is trajectory i's departure time, e_i is the arriving time, d_i is the associated driver ID and $\boldsymbol{p}_i$ is the travel path for i-th trajectory. Our goal is to fit a model that can predict the travel time y_i' given the departure time (the time slice in a day and the day of the week), the driver ID, and the travel path information. The ground-truth travel time y_i can be computed as $y_i = e_i - s_i$. The travel path $\boldsymbol{p}_i$ is represented as a sequence of links $\boldsymbol{p}_i = \{l_{i1}, l_{i2}, \cdots, l_{iT_i}\}$, where l_{ij} is the j-th link in path $\boldsymbol{p}_i$ and T_i is the total number of links for $\boldsymbol{p}_i$. After the feature extraction, the sequential features, i.e., the travel path information include link ID, link length, and the estimated link travel speed.

In this section, we elaborate *RNML-ETA* and *ADS-ETA*. The following subsections are organized as follows. Firstly, we introduce the overall framework of *RNML-ETA* in subsection III-A. Then, we give the measurement of link similarity in subsection III-B and introduce the triangle loss for links' metric learning in subsection III-C. Finally, we extend *RNML-ETA* to the framework of *ADS-ETA* in subsection III-D.

A. Overall Framework of RNML-ETA

The overall framework of *RNML-ETA* is visualized in Fig. 2. It contains two parts: the main task and the road network metric learning task (i.e., the auxiliary task 2 in the figure).

We first introduce the workflow of the main task which is the backbone task for predicting the travel time. Inputs for this part are features extracted from the raw trajectories [18]. They can be categorized as sequential features and non-sequential features. Sequential features are composed of a series of link features along the trajectory, including link ID, link length, and link travel speed, where the link travel speed is estimated by averaging the floating cars' speed within the latest 10-minute window to reflect the link's traffic condition. The non-sequential features correspond to each trajectory, including the time slice in a day (every 5 minutes), driver ID,

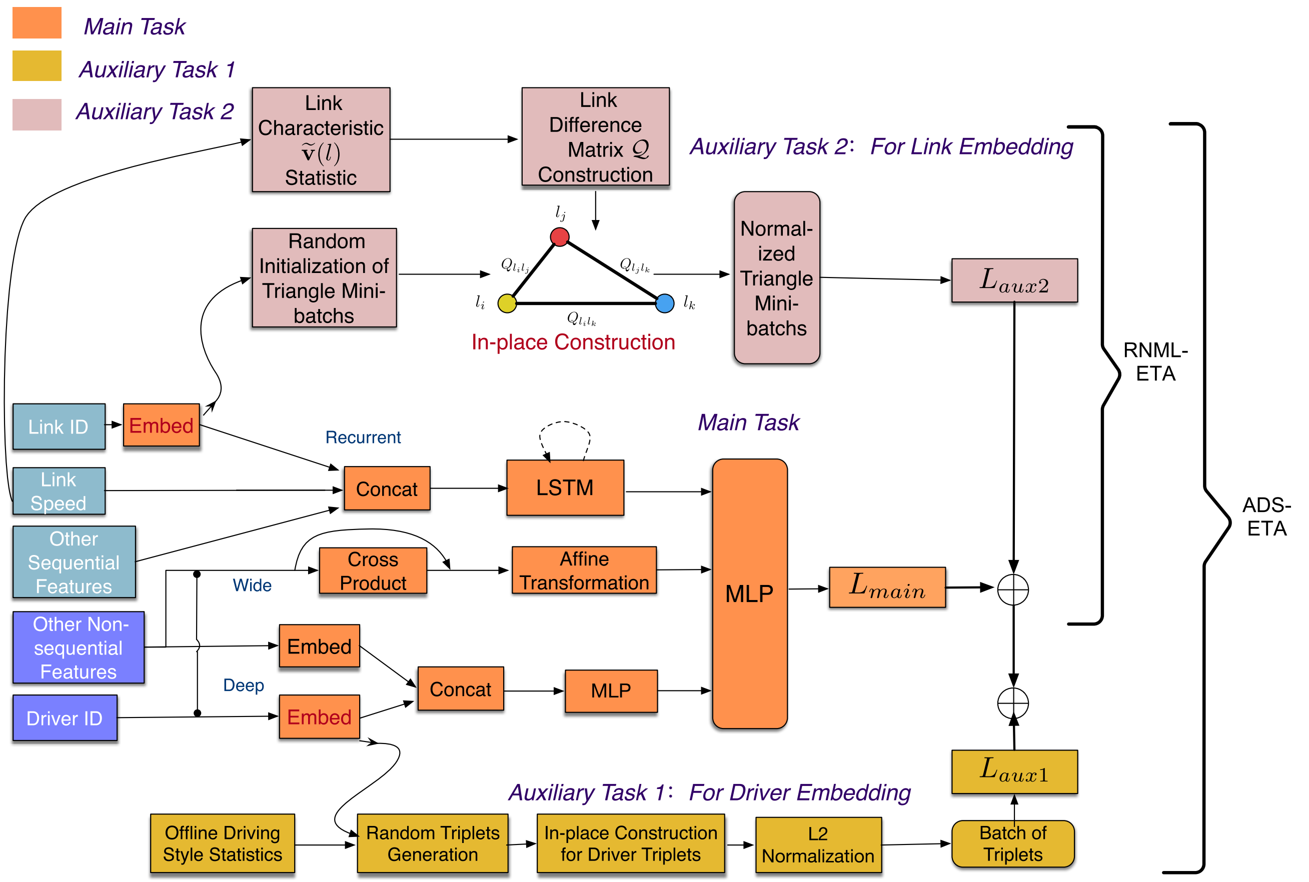

Main Task
Auxiliary Task 1
Auxiliary Task 2
Link Characteristic $\widetilde{\mathbf{v}}(l)$ Statistic
Link Difference Matrix $\mathcal{Q}$ Construction
Auxiliary Task 2: For Link Embedding
Random Initialization of Triangle Mini-batchs
l_j
$Q_{l_i l_j}$
$Q_{l_j l_k}$
l_i
l_k
$Q_{l_i l_k}$
In-place Construction
Normal-ized Triangle Mini-batchs
L_{aux2}
RNML-ETA
ADS-ETA
Link ID
Embed
Recurrent
Concat
LSTM
Main Task
Link Speed
Other Sequential Features
Wide
Cross Product
Affine Transformation
MLP
L_{main}
Other Non-sequential Features
Embed
Deep
Concat
MLP
Driver ID
Embed
L_{aux1}
Auxiliary Task 1: For Driver Embedding
Offline Driving Style Statistics
Random Triplets Generation
In-place Construction for Driver Triplets
L2 Normalization
Batch of Triplets

Fig. 2. The overall architecture of *RNML-ETA* and *ADS-ETA*. *ADS-ETA* contains three components: (1) the main task: mining the spatial-temporal dependencies from different category features; (2) the auxiliary task 1: to improve the quality of driver embedding vectors by combining the information of sparse drivers and dense drivers [13]; (3) the auxiliary task 2: Road Network Metric Learning task in order to improve the quality of link embedding vectors. *RNML-ETA* could be considered as a special case of *ADS-ETA* when only containing the main task as well as the auxiliary task 2, i.e., $\beta_1 = 0$.

and the day of the week. Both link IDs and driver IDs are embedded into vectors to encode the semantic information in a data-driven way. For example, in an embedding table $\boldsymbol{E}_L \in \mathbb{R}^{20\times M}$, the l_{ij}-th column $\boldsymbol{E}_L(:, l_{ij})$ is served as the distributional representation of link l_{ij}.

The Wide-Deep-Recurrent (WDR) model is adopted to deal with the mentioned input features in the main task [5], which is one of the state-of-the-art ETA models and is widely deployed in practical scenarios. The model contains the wide module, the deep module, and the recurrent module. The wide module constructs a second-order cross-product and an affine transformation for the non-sequential features, and the deep module also learns patterns from the non-sequential features. The deep module uses the embedding technology followed by the Multi-Layer Perceptron (MLP) after the concatenating operation. In our experiments, the dimension of the embedded space for all non-sequential features is 20. The hidden state size of MLP is 128 and we choose ReLU [25] as the activation function for the MLP. The recurrent module is the critical module for automatically learning and reserving the sequential dependencies between different trajectory links. Here Long-Short Term Memory network (LSTM) [32] is chosen as the feature extractor for the concatenated sequential features. The hidden state and memory cells of LSTM in our experiments are initialized as zeros, and the hidden state size is 128. Finally, the outputs of the wide and deep modules and the last hidden state of the LSTM are concatenated as the predictor's input to estimate the travel time y'_i, where the predictor is an MLP with a hidden state size of 128 in our paper.

The parameters of the main task are trained under the Mean Absolute Percentage Error (MAPE) loss:

$$L_{main} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_i - y'_i\right|}{y_i}, \tag{1}$$

where y_i is the ground-truth travel time. In line with the users' tolerance, MAPE it is relative and reasonable for the trips of different lengths. Thus, MAPE is popularly adopted as the objective function and evaluation metric for the research of ETA.

RNML-ETA introduces the auxiliary task 2 under MTL framework, as shown in Fig. 2. The function of auxiliary task 2 is to improve the link embedding quality of cold links to effectively alleviate the road network sparsity problem, which is discussed in section I. Due to the fact that link ID embedding is constantly and iteratively updated during the training process, it is well suited to be the object for applying metric learning. More specifically, we leverage the metric learning to reduce the distance between cold links and similar hot links in the embedded space. In such a manner, the knowledge, i.e., road network patterns of hot links are transferred to the cold links. The objective function of *RNML-ETA* is:

$$L_{RNML} = (1 - \beta_2) \cdot L_{main} + \beta_2 \cdot L_{aux2}, \tag{2}$$

where β_2 is a hyper-parameter to balance the trade-off between the main task and the auxiliary task 2. More detailed process of this auxiliary task will be introduced in the following subsections.

B. Link Similarity

The first step is to define the link similarity for auxiliary task 2. The distribution of travel speed across different times is adopted to depict the traffic characteristic of the link. For each day, a series of time bins $\{\tau_1, \tau_2, \cdots, \tau_K\}$ is constructed. These time bins satisfy the following conditions:

$$\tau_i \cap \tau_j = \emptyset, \ \forall i \neq j \tag{3}$$

$$\tau_1 \cup \tau_2 \cup \cdots \cup \tau_K = 24 \ hours \tag{4}$$

Then, we statistic the average travel speed for link l and time bin τ_k by computing:

$$\begin{aligned} \bar{v}_k(l) &= \frac{1}{Z} \sum_{i=1}^{N} \sum_{j=1}^{T_i} v_{ij} I_{s_i \in \tau_k} I_{l_{ij}=l}, \\ Z &= \sum_{i=1}^{N} \sum_{j=1}^{T_i} I_{s_i \in \tau_k} I_{l_{ij}=l}, \end{aligned} \tag{5}$$

where v_{ij} is the travel speed feature of j-th link in i-th trip, and I_{cond} is an indicator that $I_{cond} = 1$ if $cond$ is satisfied and $I_{cond} = 0$ otherwise. Intuitively, we find a subset of the link l's travel speed features by selecting those whose departure time belongs to the time bin τ_k, and then compute the average on the subset. In this work, we retain three-time bins, which represent the morning peak (τ_1 is from 5 a.m to 11 a.m), the evening peak (τ_2 is from 4 p.m to 10 p.m), and the off-peak time (τ_3 takes the remaining hours).

We further scale the speeds to be within [0, 1] by applying $\widetilde{v}_k(l) = (\bar{v}_k(l) - a)/(b - a)$, where a and b are the minimum and maximum of $\{\bar{v}_k(l), k = 1 \cdots K, l = 1 \cdots M\}$. We finally get a normalized speed histogram of link l:

$$\widetilde{\boldsymbol{v}}(l) = [\widetilde{v}_1(l), \ \widetilde{v}_2(l), \ \widetilde{v}_3(l)]^T. \tag{6}$$

The link difference matrix $\boldsymbol{Q} \in \mathbb{R}^{M \times M}$ can be computed as follows:

$$Q_{ij} = Q_{ji} = \|\widetilde{\boldsymbol{v}}(i) - \widetilde{\boldsymbol{v}}(j)\|_2, \tag{7}$$

where Q_{ij} is the element of $\boldsymbol{Q}$ measuring the difference between links with ID=i and ID=j. A smaller difference means a larger similarity. The proposed link similarity reflects the traffic patterns appropriately but does not need any extra information.

C. Triangle Loss

We also propose a novel metric learning loss function, triangle loss, for the link metric learning. Triangle loss is based on the triplet loss [19] while solving its weakness that the interaction is limited in one update. Unlike the triplet loss with only one restriction, our triangle loss has three restrictions by taking turns as the anchor. Furthermore, our triangle loss is more effective in improving the ETA accuracy and converges faster. We introduce the triangle loss as follows.

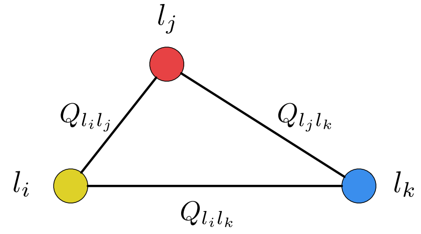


Fig. 3. The visualized demonstration of the triangle formed by the link distances. The order of the triangle's edge lengths should satisfy the relation in Eq. 8.

Suppose we have three links with ID=l_i, l_j, l_k and the corresponding differences $Q_{l_i l_j}$, $Q_{l_j l_k}$ and $Q_{l_i l_k}$, without loss of generality, we assume:

$$Q_{l_i l_j} < Q_{l_j l_k} < Q_{l_i l_k}. \tag{8}$$

We then compute the euclidean distances between the pair-wise embedding vectors of link l_i, l_j and l_k,

$$\begin{cases} D_{l_i l_j} = \|\widetilde{\boldsymbol{E}}_L(:, l_i) - \widetilde{\boldsymbol{E}}_L(:, l_j)\|_2 \\ D_{l_i l_k} = \|\widetilde{\boldsymbol{E}}_L(:, l_i) - \widetilde{\boldsymbol{E}}_L(:, l_k)\|_2 \\ D_{l_k l_j} = \|\widetilde{\boldsymbol{E}}_L(:, l_k) - \widetilde{\boldsymbol{E}}_L(:, l_j)\|_2 \end{cases} \tag{9}$$

where $\widetilde{\boldsymbol{E}}_L(:, l_i) = \boldsymbol{E}_L(:, l_i)/\|\boldsymbol{E}_L(:, l_i)\|_2$ is the L_2 normalized embedding vector. The three distances $D_{l_i l_j}$, $D_{l_j l_k}$ and $D_{l_i l_k}$ forms a triangle. We aims to restrict the lengths of the triangle edges to be in the same order as in Eq. 8, which derives three inequations:

$$\begin{aligned} &D^2_{l_i l_j} + \alpha_1 < D^2_{l_j l_k}, \\ &D^2_{l_i l_j} + \alpha_2 < D^2_{l_i l_k}, \\ &D^2_{l_j l_k} + \alpha_3 < D^2_{l_i l_k} \end{aligned} \tag{10}$$

where α_1, α_2 and α_3 are required margins. The distances forms a triangle and the visualized demonstration is shown in Fig. 3. The triangle loss is in the form of:

$$\begin{aligned} L_{aux2} = \frac{1}{U} \sum_{l_i, l_j, l_k} \Big(&\gamma_1 \Big[D^2_{l_i l_j} - D^2_{l_j l_k} + \alpha_1 \Big]_+ \\ &+ \gamma_2 \Big[D^2_{l_i l_j} - D^2_{l_i l_k} + \alpha_2 \Big]_+ \\ &+ \gamma_3 \Big[D^2_{l_j l_k} - D^2_{l_i l_k} + \alpha_3 \Big]_+ \Big), \end{aligned} \tag{11}$$

where the operator $[x]_+ = max(x, 0)$ and U is the number of possible triangles in the training set, γ_1, γ_2 and γ_3 are hyper-parameters to adjust the weights of the three distance relationships. For a mini-batch of trajectories, we compute the auxiliary loss by randomly combining triangles with all the links in the trajectories. In practice, the in-place construction is needed after getting an initialized triangle. More specifically, in the triangle, the point whose opposite side is the largest should be l_j. Analogously, the point whose opposite side is the smallest should be l_k and the point whose opposite side is the middle edge should be l_i.

D. Framework of ADS-ETA

We next introduce the model framework of *ADS-ETA*. *ADS-ETA* is proposed to deal with a more difficult situation where driver and road network sparsity problems occur simultaneously. In this situation, the driver preference information and the link information of the road network are scarce at the same time. Therefore, we propose this unified framework under the MTL paradigm by constructing two auxiliary tasks to improve the driver and link embedding vectors, respectively. As shown in Fig. 2, *ADS-ETA* consists of three tasks: the main task, the auxiliary task 1, and the auxiliary task 2. The main task is introduced in detail in Section III-A, which captures spatial-temporal dependencies from a variety of features and gives the final ETA results, the auxiliary task 1 is to transfer the knowledge from dense drivers to sparse drivers, and auxiliary task 2 is for the metric learning of road network which is the same as that of *RNML-ETA*. *RNML-ETA* concentrating on alleviating the road network sparsity problem is also the special edition ($\beta_1 = 0$) of the unified framework, *ADS-ETA*.

As illustrated in the lower third of Fig. 2, we adopt the auxiliary task of *CoDriver ETA* [13] to accomplish the auxiliary task 1. The workflow is briefly introduced as follows. Firstly, the driving style statistics that are done offline do not take up model training time. The average speed of one driver across the whole training dataset is adopted to measure the two drivers' similarities. Secondly, the triplet network [19] is adapted from face recognition and clustering to make drivers with similar driving styles closer and make drivers with different driving styles farther in the embedded space. The random driver triplets are generated in a batch manner, and the in-place construction involves a conditional branch judging whether to exchange samples in the positive and negative mini-batches [13]. Thirdly, the L_2 row normalization result of driver embedding table $\widetilde{\boldsymbol{E}}_d$ is used to construct the improved triplet loss [13]:

$$L_{aux1} = \frac{1}{N}\sum_{i=1}^{N}[\|\widetilde{\boldsymbol{E}}_d(x_i^{(a)}, :) - \widetilde{\boldsymbol{E}}_d(x_i^{(p)}, :)\|_2^2 - \|\widetilde{\boldsymbol{E}}_d(x_i^{(a)}, :) - \widetilde{\boldsymbol{E}}_d(x_i^{(n)}, :)\|_2^2 + \alpha]_+, \quad (12)$$

where the hyper-parameter α controls the driver embedding distance margin and N is the number of driver triplets which is also the total trajectory number.

All the model parameters of *ADS-ETA* are optimized under the following objective function:

$$L_{ADS} = (1 - \beta_1 - \beta_2) \cdot L_{main} + \beta_1 \cdot L_{aux1} + \beta_2 \cdot L_{aux2}, \quad (13)$$

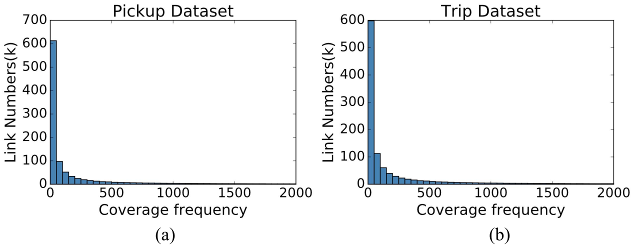
where the hyper-parameter β_1 and β_2 are the weighting coefficients of auxiliary task 1 and auxiliary task 2 respectively. In such a manner, the embedding vectors of sparse driver and cold link are improved concurrently and *ADS-ETA* effectively alleviate the driver and road network data sparsity problems simultaneously.

IV. Experiment

We conduct an extensive experimental verification based on large-scale real-world datasets collected from the DiDi platform. The datasets, competing methods, evaluation metrics, implementation details, experimental results, and analysis are presented in the following subsections.

TABLE I
STATISTICS OF DATASETS

	size	*pickup*	*trip*
training set	25 weeks	111.0M	105.5M
validation set	1 week	4.0M	4.5M
test set	1 week	4.1M	3.9M
# traversed link	-	1.2M	1.3M



(a) (b)

Fig. 4. Statistics of link coverage frequency. For both *pickup* and *trip* datasets, the links concentrate on the bands with small number of traversing trajectories.

A. Dataset

We collected massive floating car trajectories of one city in 2018 from the DiDi platform. All data is anonymized and aggregated for privacy concerns. These trajectories are split into *pickup* and *trip* datasets according to the driver's working status. A *pickup* trajectory starts when a driver responds to a passenger's request and ends when he/she picks up the passenger. A *trip* trajectory starts when a passenger gets on board and ends when he/she reaches the destination. For each trajectory order, the ground-truth value of travel time is computed as the arrival time minus the departure time. For each dataset, we use the first 25 weeks of data as the training set. The data of the 26-th week is treated as the validation set and that of the 27-th week as the test set. We remove the outliers with extremely short travel time (<60s) and extremely high average speed (>120km/h). Datasets are summarized in Table I.

A road network consists of various links in the real world, such as private community road links, local street links, and urban freeway links. Even though we have collected massive trajectories with more than 100M and cross over a half year, there is still a significant number of cold links with only a few trajectories covered. The histogram of link coverage frequency is plotted in Fig. 4 to demonstrate the sparsity problem. The median coverage frequencies of links are 42 on *pickup* and 69 on *trip*.

B. Competing Methods and Evaluation Metrics

We compare *RNML-ETA* and the extended version *ADS-ETA* with the following benchmark methods:

- Route-ETA: is a representative and straightforward method in industrial applications. It directly sums up the predicted travel time for each link and each intersection. Each link's travel time is estimated by dividing the link length by the link travel speed, and the time of each intersection is mined from the historical data. The biggest advantage of Route-ETA is the fast inference speed. However, the accuracy is often much worse than deep learning-based methods. We only choose this non-deep learning method as a representative. Other non-deep learning methods, such as PTTE [7], GBDT [5], TEMP [24] are also verified to be inferior to deep learning-based methods by [5].

- WDR [5]: is one of the state-of-the-art and most popular industry methods in terms of prediction performance for ETA. This method uses non-sequential and sequential features by combining the wide model, the deep model, and the recurrent model. It has been tested online in large-scale application scenarios and shows excellent performance [5].
- WDR-no-link-emb: is the WDR without embedding link ID. We use this model to demonstrate link embedding vectors' contribution where the *RNML-ETA* and *ADS-ETA* aim to improve.
- *RNML-ETA*-triplet: is the method replacing the auxiliary task's loss in our *RNML-ETA* with triplet loss [19]. We use it to demonstrate the effectiveness of our proposed triangle loss.

To compare the prediction performance of different methods, we introduce three widely used metrics: the Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). MAPE is relative to the ground truth travel time, thus leading to the objective measurement regarding the long and short trajectories. It also works as the loss function of the main task. MAE and RMSE are two other important and popular metrics to evaluate the performance of prediction tasks in ITS. They are formulated as follows:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_i - y_i'\right|}{y_i}, \tag{14}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - y_i'\right|, \tag{15}$$

$$\text{RMSE} = \left[\frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_i'\right)^2\right]^{1/2}, \tag{16}$$

where y_i and y_i' are the ground truth travel time and estimated time, respectively, and N is the number of samples.

C. Implementation Details

All the deep learning-based methods, i.e., WDR, WDR-no-link-emb, *RNML-ETA* and *ADS-ETA* are implemented in PyTorch [44]. They are trained and tested on a single NVIDIA Tesla P40 GPU. The mini-batch size and the maximal iteration number are set as 256 and 7 million, respectively. The hyper-parameters of *RNML-ETA* and *ADS-ETA* are selected on the validation set. For *RNML-ETA*, we use margins $\alpha_1 = \alpha_3 = 0.005$, $\alpha_2 = 0.02$ and weights $\gamma_1 = \gamma_3 = 0.3$, $\gamma_2 = 0.4$ in the triangle loss for both *pickup* and *trip* datasets. The task weight β_2 is 0.52 for *pickup* and 0.35 for *trip*. For *ADS-ETA*, the weights of two auxiliary tasks are $\beta_1 = 0.2$, $\beta_2 = 0.35$ for *pickup* and $\beta_1 = 0.25$, $\beta_2 = 0.2$ for *trip*. In the auxiliary task 1, the margin α is 0.01 for both *pickup* and *trip* datasets. In the auxiliary task 2, the margins and weights in the triangle loss are the same as those of *RNML-ETA*. All the parameters, such as the MLP weights and the embedding vectors, are jointly trained using Adam [45] optimizer, which is a stochastic gradient descending method. Adam can adaptively adjust the step size according to the historical gradients and thus accelerate the convergence. The initial learning rate is set to 0.0002.

TABLE II
RESULTS OF THE PICKUP DATASET

	MAPE (%)	MAE (sec)	RMSE (sec)
Route-ETA	25.010	69.008	106.966
WDR-no-link-emb	20.845	59.018	95.876
WDR	19.386	54.686	89.976
RNML-ETA-triplet (ours)	19.339	54.558	89.317
RNML-ETA (ours)	19.215	**53.546**	**87.617**
ADS-ETA (ours)	**19.108**	53.810	88.648

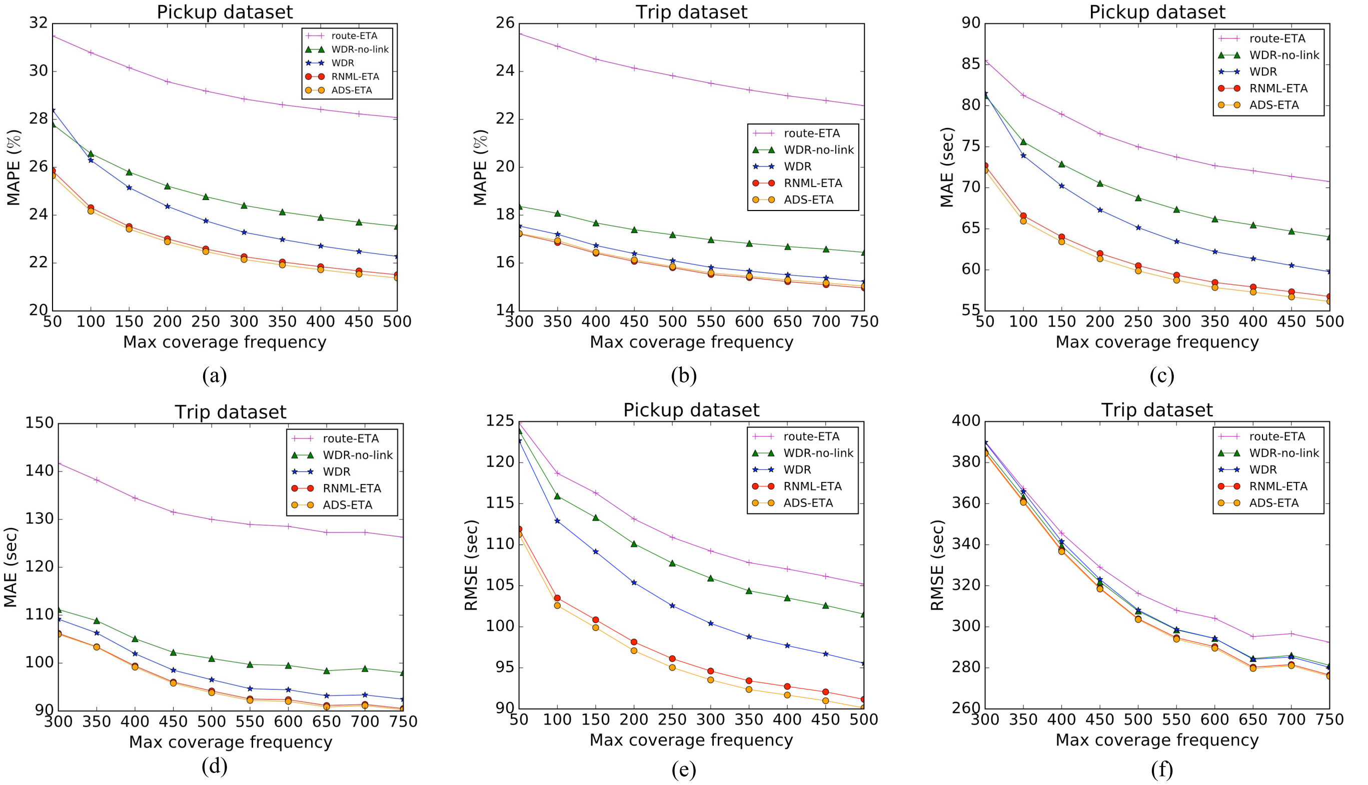
TABLE III
RESULTS OF THE TRIP DATASET

	MAPE(%)	MAE (sec)	RMSE (sec)
Route-ETA	15.440	150.560	248.736
WDR-no-link-emb	12.742	117.337	197.652
WDR	11.737	108.919	186.083
RNML-ETA-triplet (ours)	11.661	108.638	185.388
RNML-ETA (ours)	**11.597**	108.519	185.897
ADS-ETA (ours)	11.607	**108.014**	**184.315**

D. Experimental Results and Analysis

1) Competing Results of the Overall Precision: We list three evaluate metrics of different approaches corresponding to the *pickup* data and the *trip* data in Table II and Table III, respectively. The results and analysis in terms of the overall precision are described as follows. (1) The best scores marked with bold font are all obtained by our methods, i.e., *RNML-ETA* and *ADS-ETA*. (2) The metric learning task for links improves the quality of link embedding leading to a more accurate ETA by comparing WDR and *RNML-ETA*. In our experiment, *RNML-ETA* relatively reduces 2.62% RMSE on *pickup* data and 1.19% MAPE on *trip* data compared with WDR which is one of the state-of-the-art methods. (3) *ADS-ETA* is superior to the WDR and similar to *RNML-ETA* in terms of three metrics on two datasets. (4) *RNML-ETA* is better than *RNML-ETA*-triplet on all three metrics except RMSE on *trip* data. In terms of MAPE, *RNML-ETA* makes a relatively 0.64% improvement on *pickup* data and a relatively 0.55% improvement on *trip* data compared to *RNML-ETA*-triplet. These results demonstrate the effectiveness of triangle loss on link metric learning compared to the common triplet loss. (5) The link embedding technique's importance is demonstrated by comparing the WDR and the WDR-no-link-emb. Through link embedding, the WDR realizes relative 7.0% and 7.9% reduction regarding to MAPE on *pickup* and *trip* data, respectively. (6) Rule-based Route-ETA is inferior to deep learning-based methods in terms of all the metrics.

2) Road Network Sparsity Problem: Experiments using a series of subset data with cold links from the whole dataset are conducted to illustrate the road network sparsity problem.



(a) (b) (c) (d) (e) (f)

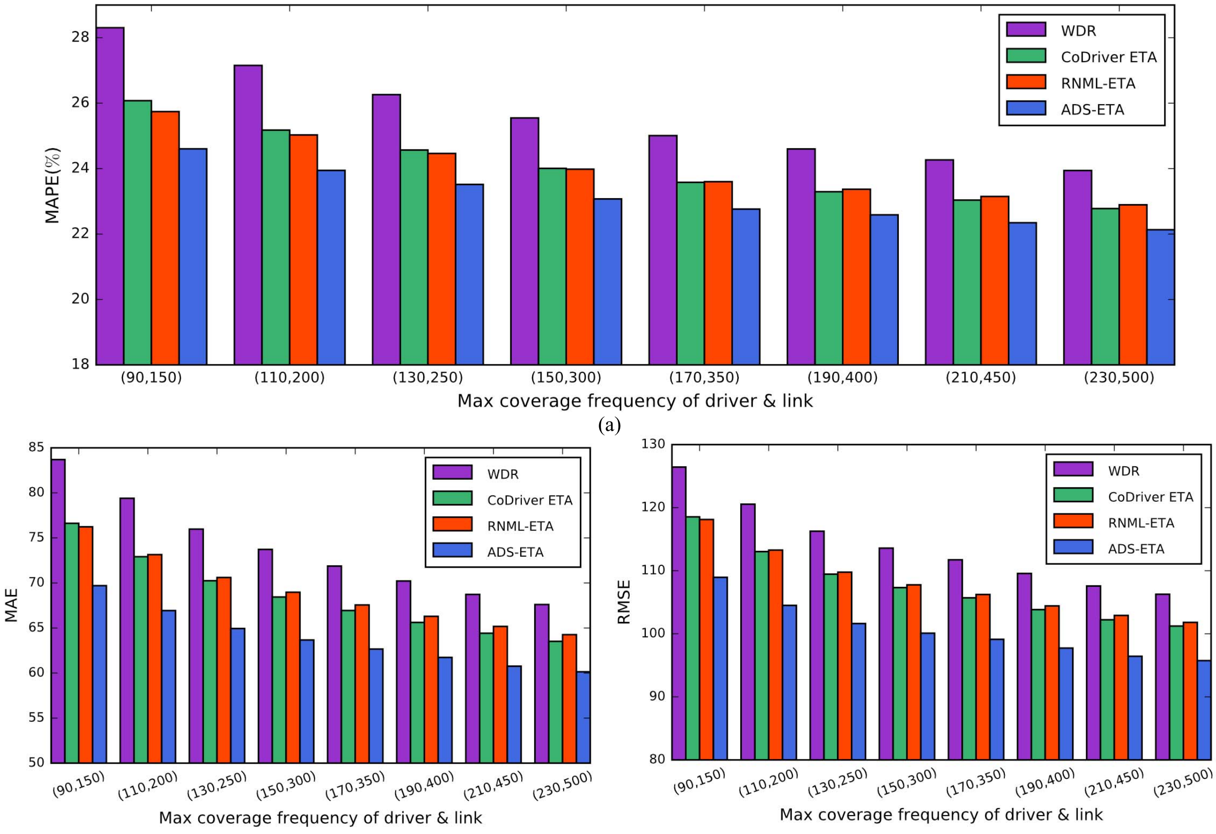
Fig. 5. Experimental results when road network data sparsity problem occurs with different link coverage levels. For a threshold δ_l, we keep the trajectory that at least 25% of the contained links have coverage frequencies less than δ_l. The 6 subfigures stand for (a) MAPE on *pickup* data, (b) MAPE on *trip* data, (c) MAE on *pickup* data, (d) MAE on *trip* data, (e) RMSE on *pickup* data and (f) RMSE on *trip* data.

Each subset contains trajectories of which at least 25% links have coverage trajectory orders less than a threshold δ_l. Varying δ_l from 50 to 500 on *pickup* data and from 300 to 750 on *trip* data in a step size 50, we obtain ten subsets for each dataset. A lower δ_l represents a more serious road network sparsity problem. Three metrics in the test phase of *RNML-ETA*, *ADS-ETA*, and other baselines are recorded, and the curves are displayed in Fig. 5.

The comparison results and analysis in the scene of the road network sparsity problem are presented below. (1) The road network sparsity problem makes the prediction accuracy of different methods drop compared with their overall accuracy, which illustrates the significance of addressing the road network sparsity problem. (2) As δ_l decreases, which represents the degree of road network sparsity increases, all the methods perform worse. (3) Compared with all baselines, *RNML-ETA* and *ADS-ETA* are superior significantly in all sparse subsets. For example, on *pickup* data, *ADS-ETA* relatively improves $[4.04\%, 9.66\%]$ on MAPE and $[6.05\%, 11.52\%]$ on MAE compared with the WDR. On *trip* data, *RNML-ETA* relatively improves $[1.73\%, 1.99\%]$ and $[2.10\%, 2.73\%]$ in terms of MAPE and MAE, respectively. These improvements demonstrate the effectiveness of *RNML-ETA* and *ADS-ETA* in alleviating the road network sparsity problem. (4) In Fig. 5, *ADS-ETA* and *RNML-ETA* are marked by circles with dark yellow and red, respectively. On *pickup* data, the curves of *ADS-ETA* have a similar and slightly lower trend compared with *RNML-ETA*. The curves of *ADS-ETA* are close to those of *RNML-ETA* on *trip* data. These results show that *ADS-ETA* and *RNML-ETA* perform similarly when handling the road network data sparsity problem.

3) Both Data Sparsity Problems: In real-world applications, scenarios with both road network and driver sparsities are also common. Prediction performances of different ETA models on the trajectories with sparse drivers and many cold links can reflect their ability to deal with such scenarios. A series of sparse subsets from the test dataset of the *pickup* data are selected to demonstrate the superiority of *ADS-ETA* in alleviating scenarios containing both sparsity problems. For each subset, the sparsity is restricted by a pair of thresholds, i.e., (δ_d, δ_l). The driver's maximum coverage frequency is less than δ_d, and trajectories have at least 25% of links with coverage frequencies less than δ_l. δ_d varies from 90 to 230 with an interval of 20, and δ_l varies from 150 to 500 with an interval of 50. Then, we obtain eight subsets, and they are used to test the effectiveness of different methods under different degrees of sparsity. The competing methods are WDR [5], *CoDriver ETA* [13] and *RNML-ETA*. The latter two methods have ever been adept in dealing with the scenario either with driver sparsity or road network sparsity. We test all models on all eight subsets with three metrics, i.e., MAPE, MAE, RMSE. Results are reported in Table IV, Table V and Table VI, respectively. Fig. 6 gives an intuitive bar chart to illustrate the performances of different approaches.

We summarize and analyze the results from the Table IV, Table V, Table VI and Fig. 6 as follows. (1) When the driver data sparsity problem and the road network sparsity problem co-occur, the ETA system's prediction performance drops a lot,



(a)

(b)

(c)

Fig. 6. Fine grained experimental result comparison when two data sparsity problems appear simultaneously. The experiment are conducted on different coverage levels of driver and link. *ADS-ETA* is obviously superior to WDR and more accurate than *CoDriver ETA* and *RNML-ETA* regarding to all metrics and all coverage levels. (a) Comparison in terms of MAPE. (b) Comparison in terms of MAE. (c) Comparison in terms of RMSE.

TABLE IV

EXPERIMENTAL RESULT COMPARISON: MAPE (%) WHEN BOTH DATA SPARSITY PROBLEMS OCCUR

Threshold set (proportion)	WDR	CoDriver	RNML	ADS
(90, 150) (0.18%)	28.306	26.076	25.742	**24.605**
(110, 200) (0.26%)	27.154	25.177	25.030	**23.946**
(130, 250) (0.36%)	26.261	24.568	24.463	**23.517**
(150, 300) (0.46%)	25.548	24.007	23.983	**23.074**
(170, 350) (0.57%)	25.008	23.580	23.600	**22.763**
(190, 400) (0.69%)	24.601	23.294	23.369	**22.587**
(210, 450) (0.82%)	24.266	23.037	23.148	**22.345**
(230, 500) (0.95%)	23.942	22.779	22.893	**22.132**

TABLE V

EXPERIMENTAL RESULT COMPARISON: MAE (SEC) WHEN BOTH DATA SPARSITY PROBLEMS OCCUR

Threshold set (proportion)	WDR	CoDriver	RNML	ADS
(90, 150) (0.18%)	83.694	76.624	76.232	**69.702**
(110, 200) (0.26%)	79.398	72.916	73.147	**66.937**
(130, 250) (0.36%)	75.977	70.248	70.607	**64.942**
(150, 300) (0.46%)	73.722	68.442	68.974	**63.671**
(170, 350) (0.57%)	71.875	66.945	67.560	**62.663**
(190, 400) (0.69%)	70.220	65.620	66.298	**61.737**
(210, 450) (0.82%)	68.730	64.424	65.176	**60.765**
(230, 500) (0.95%)	67.607	63.523	64.263	**60.141**

and the drop gap is larger than that with only road network sparsity. All the metrics of deep learning-based ETA methods, such as WDR and *RNML-ETA*, are worse compared with the scenarios only with the same road network sparsity degree. Furthermore, as the degree of both sparsity increases, i.e., the threshold set values decrease, the gap increases. These phenomenons demonstrate the necessity to solve such data sparsity problems in real-world scenarios. (2) The prediction performances of *CoDriver ETA* and *RNML-ETA* are better than WDR on all metrics. For instance, when the threshold set is (110, 200), *CoDriver ETA* and *RNML-ETA* obtain a relative 7.28% improvement and a 7.82% improvement in terms of MAPE to the WDR, respectively. This is because *CoDriver ETA* and *RNML-ETA* have effectively alleviated the driver data sparsity and the road network data sparsity, respectively. (3) *ADS-ETA* shows superior estimation ability compared to both *CoDriver ETA* and *RNML-ETA*. For example, when the threshold set is (110, 200), *ADS-ETA* obtains a relative improvement of 11.81%, 4.89% and 4.33% compared to WDR, *CoDriver ETA*, and *RNML-ETA* on MAPE, respectively. The detailed quantitative results demonstrate that *ADS-ETA* could effectively alleviate the driver and road network data sparsity problems simultaneously.

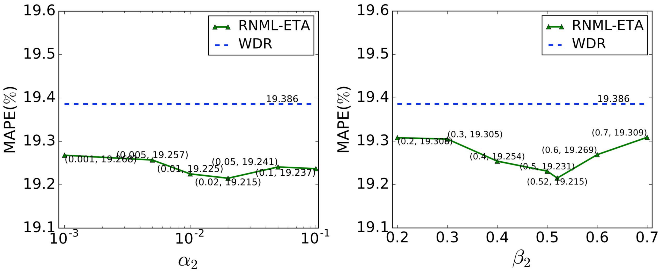
E. Influence of Hyper-Parameter

We explore the influence of *RNML-ETA*'s hyper-parameters, and plot the performance curves of *pickup* data

TABLE VI

EXPERIMENTAL RESULT COMPARISON: RMSE (SEC) WHEN BOTH DATA SPARSITY PROBLEMS OCCUR

Threshold set (proportion)	WDR	CoDriver	RNML	ADS
(90, 150) (0.18%)	126.436	118.544	118.124	**108.955**
(110, 200) (0.26%)	120.550	113.033	113.274	**104.501**
(130, 250) (0.36%)	116.267	109.459	109.774	**101.630**
(150, 300) (0.46%)	113.587	107.325	107.764	**100.100**
(170, 350) (0.57%)	111.735	105.704	106.235	**99.121**
(190, 400) (0.69%)	109.570	103.838	104.422	**97.736**
(210, 450) (0.82%)	107.579	102.224	102.915	**96.436**
(230, 500) (0.95%)	106.273	101.230	101.811	**95.749**



(a) (b)

Fig. 7. The influence of *RNML-ETA*'s hyper-parameters: (a) for the margin α_2 in the triangle loss, and (b) for the weight β_2 balancing the main task and the auxiliary task for link embedding. Under different hyper-parameters, *RNML-ETA* generally outperforms the competitor WDR, which demonstrates the robustness of *RNML-ETA*.

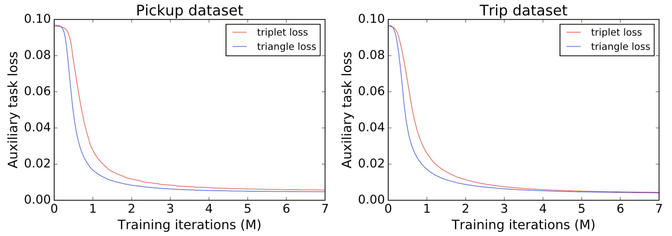
in Fig. 7 by varying two representative hyper-parameters, i.e., the margin α_2 and the weight β_2 for auxiliary task 2. The basic configuration is the same as in section IV-C, namely, $\alpha_1 = \alpha_3 = 0.005$, $\alpha_2 = 0.02$, $\gamma_1 = \gamma_3 = 0.3$, $\gamma_2 = 0.4$ and $\beta_2 = 0.52$.

The hyper-parameter α_2 is a bit more special than α_1 and α_3, because it controls the gap between the longest edge and the shortest edge in the triangle loss. If this restriction is broken, the model is far from our expected status and needs a stronger gradient to update the parameters. Usually, we set $\alpha_2 > \alpha_1 + \alpha_3$ and the curve in Fig. 7 (a) shows that the most accurate performance of *RNML-ETA* is achieved with a $\alpha_2 = 0.02$. Moreover, *RNML-ETA* achieves better performance than WDR from $\alpha_2 = 0.001$ to 0.1, which demonstrates that the superiority of *RNML-ETA* is not sensitive to the margin hyper-parameter.

The weight β_2 is to balance the trade-off between the main task and the auxiliary task for link embedding. In extreme cases, *RNML-ETA* degenerates to WDR if $\beta_2 = 0$ and degenerates to a pure metric learning model if $\beta_2 = 1$. Fig. 7 (b) shows that the advantage of *RNML-ETA* over WDR is robust in a wide range of β_2 from 0.2 to 0.7 and that the best performance is achieved at $\beta_2 = 0.52$.

F. Convergence Speed Comparison

We also compare the convergence efficiency of the triplet loss and the proposed triangle loss when serving as the loss function of the *RNML-ETA*'s auxiliary task. Fig. 8 shows the loss function curves versus the training iteration (mini-batch size = 256) on two datasets. The distance between every two points on the curve is 20k, and the value of each point is obtained by averaging the loss values of the last 20k iterations. From the figures, we could find that the proposed triangle loss needs fewer iterations to converge. The experimental results validate that the triangle loss is superior to the triplet loss in terms of the convergence rate.



(a) (b)

Fig. 8. The convergence speed comparison in the training phase: triplet loss v.s. triangle loss. (a) the loss function curves on pickup data, (b) the loss function curves on trip data.

V. CONCLUSION AND FUTURE WORK

In this paper, we discuss the impact of the data sparsity problems in terms of driver and road network information in the ETA system. A set of solutions are proposed to alleviate data sparsity problems, which are of great importance for enhancing user experience. Specifically, we propose a novel framework *RNML-ETA* equipped with a novel triangle loss to alleviate the road network sparsity problem. Furthermore, we extend *RNML-ETA* and propose a unified framework *ADS-ETA* to effectively alleviate data sparsity problems in ETA that arise from both road network and driver sparsity. Extensive experiments on two massive floating-car datasets from the DiDi platform demonstrate the effectiveness of *RNML-ETA* and *ADS-ETA*. When the road network sparsity occurs, both *RNML-ETA* and *ADS-ETA* significantly improve the prediction precision compared with the benchmark WDR model which is one of the state-of-the-art methods in the literature. When the road network and driver sparsity problems co-occur, *ADS-ETA* further outperforms *RNML-ETA* significantly.

There are still many other open questions in ETA. An interesting avenue of research in future work is to explore other deep-seated problems that hinder the prediction precision of deep learning-based ETA methods, such as emergencies in the dynamic traffic system and the long time traffic congestion.

REFERENCES

[1] G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010.

[2] L. Figueiredo, I. Jesus, J. T. Machado, J. R. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *Proc. ITSC*, Aug. 2001, pp. 1206–1211.

[3] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[4] S. Çolak, A. Lima, and M. C. González, "Understanding congested travel in urban areas," *Nature Commun.*, vol. 7, no. 1, pp. 1–8, Apr. 2016.

[5] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. ACM SIGKDD*, Jul. 2018, pp. 858–866.

[6] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *Proc. AAAI*, Apr. 2018, pp. 1–8.

[7] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. ACM SIGKDD*, Aug. 2014, pp. 25–34.

[8] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1679–1693, Dec. 2012.

[9] H. Chen, H. A. Rakha, and C. C. McGhee, "Dynamic travel time prediction using pattern recognition," in *Proc. 20th World Congr. Intell. Transp. Syst.* Delft, The Netherlands: TU Delft, 2013, pp. 1–17.

[10] F. Zhang, X. Zhu, T. Hu, W. Guo, C. Chen, and L. Liu, "Urban link travel time prediction based on a gradient boosting method considering spatiotemporal correlations," *ISPRS Int. J. Geo-Inf.*, vol. 5, no. 11, p. 201, Nov. 2016.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, May 2015.

[12] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. SIGKDD*, Jul. 2018, pp. 1695–1704.

[13] Y. Sun *et al.*, "CoDriver ETA: Combine driver information in estimated time of arrival by driving style learning auxiliary task," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 1–12, Dec. 2020.

[14] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[15] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proc. Interspeech*, Aug. 2013, pp. 3771–3775.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NeurIPS*, 2013, pp. 3111–3119.

[17] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, Dec. 1997, doi: 10.1007/978-3-030-01620-3_5.

[18] Y. Sun, K. Fu, Z. Wang, C. Zhang, and J. Ye, "Road network metric learning for estimated time of arrival," in *Proc. ICPR*, Jan. 2021, pp. 1820–1827.

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, Jun. 2015, pp. 815–823.

[20] J. Yuan *et al.*, "T-Drive: Driving directions based on taxi trajectories," in *Proc. SIGSPATIAL GIS*, 2010, pp. 99–108.

[21] M. Asghari, T. Emrich, U. Demiryurek, and C. Shahabi, "Probabilistic estimation of link travel times in dynamic road networks," in *Proc. SIGSPATIAL GIS*, Nov. 2015, pp. 1–10.

[22] X. Li, G. Cong, A. Sun, and Y. Cheng, "Learning travel time distributions with deep generative model," in *Proc. WWW*, 2019, pp. 1017–1027.

[23] X. Zhan, S. Hasan, S. V. Ukkusuri, and C. Kamga, "Urban link travel time estimation using large-scale taxi data with partial information," *Transp. Res. C, Emerg. Technol.*, vol. 33, pp. 37–49, Aug. 2013.

[24] H. Wang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," in *Proc. SIGSPATIAL GIS*, Oct. 2016, p. 61.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.

[26] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, no. 1, pp. 1–40, Jan. 2009.

[27] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[28] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[29] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3913–3926, Oct. 2019.

[30] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Aug. 2018.

[31] Y. Sun, Y. Wang, K. Fu, Z. Wang, C. Zhang, and J. Ye, "Constructing geographic and long-term temporal graph for traffic forecasting," in *Proc. ICPR*, May 2020, pp. 3483–3490.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Apr. 1997.

[33] T.-Y. Fu and W.-C. Lee, "DeepIST: Deep image-based spatio-temporal network for travel time estimation," in *Proc. ACM CIKM*, Nov. 2019, pp. 69–78.

[34] W. Lan, Y. Xu, and B. Zhao, "Travel time estimation without road networks: An urban morphological layout representation approach," in *Proc. IJCAI*. Palo Alto, CA, USA: AAAI Press, Aug. 2019, pp. 1772–1778.

[35] K. Fu, F. Meng, J. Ye, and Z. Wang, "CompactETA: A fast inference system for travel time prediction," in *Proc. ACM SIGKDD*, Aug. 2020, pp. 3337–3345.

[36] Y. Sun *et al.*, "FMA-ETA: Estimating travel time entirely based on FFN with attention," in *Proc. ICASSP*, Jun. 2021, pp. 3355–3359.

[37] H. Zhang, H. Wu, W. Sun, and B. Zheng, "DeepTravel: A neural network based travel time estimation model with auxiliary supervision," in *Proc. IJCAI*, Jul. 2018, pp. 3655–3661.

[38] H. Tan *et al.*, "A tensor-based method for missing traffic data completion," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 15–27, Mar. 2013.

[39] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 108–120, Sep. 2013.

[40] G. Fusco, C. Colombaroni, and N. Isaenko, "Short-term speed predictions exploiting big data on large urban road networks," *Transp. Res. C, Emerg. Technol.*, vol. 73, pp. 183–201, Dec. 2016.

[41] M. Kaya and H. C. S. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

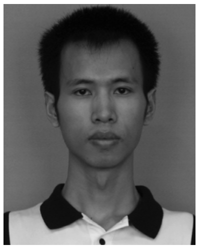
[42] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. CVPR*, vol. 1, Jun. 2005, pp. 539–546.

[43] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. NeurIPS*, 2016, pp. 1857–1865.

[44] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.

[45] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–10.

Yiwen Sun (Graduate Student Member, IEEE) received the B.E. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Automation. His research interests include machine learning, deep learning, sequence learning, spatio-temporal data mining, and intelligent transportation systems.



Wenzheng Hu received the B.S. degree from Beihang University, Beijing, China, in 2013, and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, in 2019. He is currently a Post-Doctoral Researcher with the State Key Laboratory of Automotive Satefy and Energy, Tsinghua University. His research interests include machine learning, deep learning, and intelligent transportation systems.

Donghua Zhou (Fellow, IEEE) received the B.Eng., M.Sci., and Ph.D. degrees in electrical engineering from Shanghai Jiaotong University, China, in 1985, 1988, and 1990, respectively. He was an Alexander von Humboldt Research Fellow with the University of Duisburg, Germany, from 1995 to 1996, and a Visiting Scholar with Yale University, USA, from 2001 to 2002. He joined Tsinghua University in 1996, and was promoted as a Full Professor in 1997. He was the Head of the Department of Automation, Tsinghua University, from 2008 to 2015. He is currently the Vice President of the Shandong University of Science and Technology and a Joint Professor of Tsinghua University. He has authored and coauthored over 210 peer-reviewed international journal articles and seven monographs in the areas of fault diagnosis, fault-tolerant control, and operational safety evaluation. He is a fellow of CAA and IET, a member of IFAC TC on SAFEPROCESS, an Associate Editor of *Journal of Process Control*, the Vice Chairperson of Chinese Association of Automation (CAA), and the TC Chair of the SAFEPROCESS Committee, CAA. He was also the NOC Chair of the Sixth IFAC Symposium on SAFEPROCESS 2006.

Baichuan Mo received the bachelor's degree in civil engineering from Tsinghua University and the dual master's degree in transportation and computer science from MIT, where he is currently pursuing the Ph.D. degree with the Department of Civil and Environmental Engineering. His research interests include data-driven transportation modeling, demand modeling, and applied machine learning, with a specific application in public transit systems.

Kun Fu (Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2011 and 2017, respectively. He is currently a Researcher at Beike AI Tech. Before that, he was a Researcher at DiDi AI Laboratories. His research interests include machine learning, deep learning, and intelligent transportation systems.

Zhengping Che (Member, IEEE) received the B.Eng. degree in computer science from Tsinghua University, Beijing, China, in 2013, and the Ph.D. degree in computer science from the University of Southern California, Los Angeles, CA, USA, in 2018. He is currently with the AI Innovation Center, Midea Group. His research interests lie in the areas of deep learning, computer vision, and time series analysis with applications to robot learning.

Zheng Wang (Member, IEEE) received the Ph.D. degree from the Department of Automation, Tsinghua University, China, in 2011. He is currently with Beike AI Tech. Before that, he was a Research Faculty with the University of Michigan, Ann Arbor, MI, USA, and then, he was a Researcher and the Principle Engineer with the DiDi AI Laboratories. His research interests include big data, machine learning, and data mining, with applications in transportation. His papers have been selected for the KDD Best Research Paper Runner-up in 2013 and the SocialCom Best Paper Award in 2013. He has served on Senior Program Committee/Program Committee for many conferences, including NIPS, ICML, KDD, ICLR, IJCAI, and AAAI.

Shenhao Wang received the Ph.D. (interdisciplinary) degree in computer and urban science from MIT in 2020. He is currently an Assistant Professor with the University of Florida and a Research Affiliate at MIT Urban Mobility Laboratory and Media Laboratory, Human Dynamics Group. His research focuses on developing interpretable and ethical deep learning models to analyze individual decision-making with applications to urban mobility.

Jinhua Zhao is currently the Edward H. and Joyce Linde Associate Professor of city and transportation planning at MIT. He brings behavioral science and transportation technology together to shape travel behavior, design mobility systems, and reform urban policies. He directs the MIT Urban Mobility Laboratory and Public Transit Laboratory.

Jieping Ye received the Ph.D. degree in computer science from the University of Minnesota, Twin Cities, MN, USA, in 2005. He is currently with Beike AI Tech and also a Professor with the University of Michigan, Ann Arbor, MI, USA. Before that, he was the Vice President of Didi Chuxing and a Didi Fellow. His research interests include big data, machine learning, and data mining, with applications in transportation and biomedicine. He was a recipient of the NSF CAREER Award in 2010. His papers have been selected for the Outstanding Student Paper at ICML in 2004, the KDD Best Research Paper Runner Up in 2013, and the KDD Best Student Paper Award in 2014. He has served as a Senior Program Committee Chair/the Area Chair/the Program Committee Vice Chair for many conferences, including NIPS, ICML, KDD, IJCAI, ICDM, and SDM. He has served as an Associate Editor for *Data Mining and Knowledge Discovery*, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

Jian Tang received the Ph.D. degree in computer science from Arizona State University in 2006. He is currently with the Midea Group. He has published over 180 papers in premier journals and conferences. His research interests lie in the areas of AI, the IoT, wireless networking, and mobile computing. He is an ACM Distinguished Member. He received an NSF CAREER Award in 2009. He also received several best paper awards, including the 2019 William R. Bennett Prize and the 2019 Technical Committee on Big Data (TCBD) Best Journal Paper Award from IEEE Communications Society (ComSoc), the 2016 Best Vehicular Electronics Paper Award from IEEE Vehicular Technology Society (VTS), and best paper awards from the 2014 IEEE International Conference on Communications (ICC) and the 2015 IEEE Global Communications Conference (Globecom), respectively. He has served as an Editor for several IEEE journals, including IEEE TRANSACTIONS ON BIG DATA and IEEE TRANSACTIONS ON MOBILE COMPUTING. In addition, he served as the TPC Co-Chair for a few international conferences, including the IEEE/ACM IWQoS'2019, MobiQuitous'2018, and IEEE iThings'2015, the TPC Vice Chair for the INFOCOM'2019, and an Area TPC Chair for INFOCOM 2017-2018. He was also an IEEE VTS Distinguished Lecturer and the Chair of the Communications Switching and Routing Committee of IEEE ComSoc.

Changshui Zhang (Fellow, IEEE) received the B.S. degree in mathematics from Peking University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, in 1989 and 1992, respectively. In 1992, he joined the Department of Automation, Tsinghua University, where he is currently a Professor. He is an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has authored more than 200 articles. His current research interests include pattern recognition and machine learning.