HSETA: A Heterogeneous and Sparse Data Learning Hybrid Framework for Estimating Time of Arrival

Kaiqi Chen, Guowei Chu, Xuexi Yang, Yan Shi, Kaiyuan Lei, and Min Deng

***Abstract*— The estimated time of arrival (ETA) plays a vital role in intelligent transportation systems and has been widely used as a basic service in ride-hailing platforms. Obtaining a precise ETA is a challenging task due to the complexity of the real-world geographic and traffic environments. Previous works suffer from heterogeneous sparse data learning and multiple-correlation extraction issues. Therefore, this paper presents a hybrid deep learning framework (HSETA) to estimate the vehicle travel time from massive data. First, we encode heterogeneous data to represent various features in different respects. Then, we develop an ensemble factorization machine block (EFMB) structure combined with gated recurrent unit (GRU) and multilayer perceptron (MLP) to extract information from sparse and dense features. Next, the multiple-correlation learning block (MCLB) structure that we propose is utilized to aggregate information based on multiple correlations. Finally, the travel time can be estimated by simple regression. Our extensive evaluations on two real-world datasets show that HSETA significantly outperforms all baselines. Our PyTorch implementation of HSETA and sample data are available at https://github.com/LouisChenki/HSETA**

***Index Terms*— Correlations, deep neural network, estimated time of arrival, heterogeneous data, sparse data.**

I. Introduction

THE estimated time of arrival (ETA) forecasts the travel time of vehicles from the origin to the destination under certain traffic conditions and environments. Recently, the ETA has become a core of location-based services (LBSs) and intelligent transportation systems (ITSs), which have been widely used by ride-hailing mobile apps, such as Uber and Didi Chuxing. A high-precision ETA is essential for achieving better route planning and navigation services and for enhancing user experience.

A simple way to obtain the ETA is to average the historical travel time between the origin and destination pair [1], [2]. However, in addition to historical records, various global factors, including the traffic environment, geographic environment, and driver status, affect the travel time in varying ways. As shown in Fig. 1, for tidal roads, the direction of traffic flow changes over time (Fig. 1a), and for roads near residential areas, bad weather obviously aggravates congestion (Fig. 1b). These influences ultimately determine the travel time. Because such factors are ignored, the accuracy of these methods is poor, and they are unable to meet the needs of ride-hailing platforms and high-quality transportation services [3]. Recently, due to the powerful representation and self-learning ability of deep learning, deep neural network-based techniques have been widely developed for ETA tasks [3]–[13] and have achieved better performance than historical average-based models. The existing works make effective use of path features and global factors through various methods. Based on this, these works can be divided into two categories. One is path-based methods. These approaches focus on the origin and destination of the entire trip and directly predict the overall travel time. The other type is route-based methods, which split the path into several road segments and obtain the travel time by summing over the travel times of all segments. Path-based methods easily accumulate errors due to uncertain routes and travel distances. An example is shown in Fig. 1c. The travel times of different paths between the same origin and destination range from 10 minutes to 16 minutes. The ETA gap is over half of the shortest travel time. Due to the ability to aggregate information for each road segment, route-based methods perform better and are more popular in both academia and industry. However, determining the ETA is a nontrivial problem, and even if a deep learning approach can model this effect, heterogeneity and sparsity issues still result. Additionally, the intercorrelation between global factors and route features and the intracorrelation between each segment pair need to be extracted.

For heterogeneity, different factors of the trip and environment are recorded as different data types and given different properties, such as sequential numbers (driver IDs), periodic numbers (times), text (weather information) and graphs (road networks). How can heterogeneous data be organized to retain more information, and how can the model structure be designed to extract useful features from heterogeneous data? These issues remain to be addressed.

Regarding sparsity, there are millions of road segments and hundreds of thousands of drivers in a city, but most of them have very few trip records. The sparsity problem also

Manuscript received January 4, 2022; revised April 5, 2022 and April 25, 2022; accepted April 25, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 42171459, Grant 41901319, Grant 41730105, and Grant 42071452; in part by the Joint Fund of Collaborative Innovation Center of Geo-Information Technology for Smart Central Plains, Henan; and in part by the Key Laboratory of Spatiotemporal Perception and Intelligent processing, Ministry of Natural Resources, under Grant 212203. The Associate Editor for this article was G. Guo. *(Corresponding author: Xuexi Yang.)*

The authors are with the School of Geosciences and Info-Physics, Central South University (CSU), Changsha 410083, China (e-mail: yangxuexi@csu.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3170917

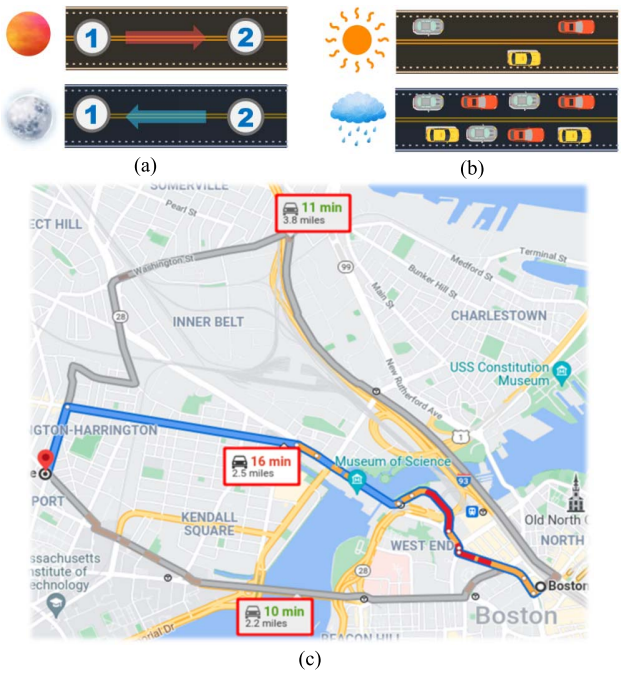


Fig. 1. Effect of the route and global factors on travel time. (a) Different traffic rules at different times. (b) Different road conditions in different weather conditions. (c) Same OD pair with different routes.

occurs in the process of feature learning. When utilizing neural networks to learn from data with a million dimensions, the representation of features is very sparse and makes the models unable to estimate reliable parameters [14].

There are various factors that affect the ETA, which forces the model to learn features from many aspects. The model should pay more attention to features that are rich in information or difficult to learn. For example, when the route-based method addresses a trip with over 100 segments, it should take the key segments using the current global factors as the main learning objects. This leads to requirements for the model in terms of learning multiple correlations between various factors.

Some deep learning-based methods have achieved good performance for ETA prediction [3]–[13] and other traffic prediction tasks [15]–[18]. However, they do not effectively respond to the above issues, which significantly affects the prediction accuracy. Therefore, to address the heterogeneity, sparsity and multiple-correlation learning problems, we propose a novel ETA hybrid framework named HSETA. Specifically, we model heterogeneous data by utilizing encoding methods and extract various features by combining a variety of neural networks. In addition, we propose an ensemble factorization machine block (EFMB) to solve the problem of sparse feature learning. Moreover, to address the insufficiency of capturing intercorrelation and intracorrelation information, we design an expectation calculation block, namely, the multiple-correlation learning block (MCLB). This hybrid framework leverages the advantages of machine learning and deep learning as well as the advantages of sequence learning and attention learning. We verify the performance gain of the proposed method with experiments on two real-world datasets. The results show that HSETA achieves state-of-the-art results compared to baseline models. Our contributions can be summarized as follows:

(1) We propose a novel ETA learning hybrid framework named HSETA that integrates various neural network structures and factorization machines to address the learning challenges of heterogeneous sparse data in a targeted manner.
(2) We analyze the effect mechanism of multiple correlations on the ETA and design an expectation calculation block to learn this mechanism. To the best of our knowledge, this is the first work focusing on the complex multiple correlations in the ETA problem.
(3) We evaluate our method under realistic conditions using two real-world datasets from Didi Chuxing and compare it with state-of-the-art techniques. The results show that HSETA can achieve the best performance.

The rest of the paper is organized as follows: Section II systematically reviews the related works on ETA tasks. Section III presents a detailed description of the proposed HSETA framework, while Section IV describes the performance of our framework through two real-world traffic datasets, including the setting of the model hyperparameters and the result analysis. Section V concludes the paper.

II. Related Works

ETA is one of the fundamental functions of an intelligent transportation system, and it assists other ITS tasks, such as route planning and autonomous driving [4]. Therefore, a variety of techniques have been proposed in both academic and industrial communities to solve ETA problems [19]. In general, ETA methods can be divided into two categories: path-based methods and route-based methods.

Path-based methods aim to predict travel time depending only on origin-destination (OD) information and environmental factors. Wang *et al.* [2] designed a model based on the nearest neighbor method that averages the scaled travel time of all historical trips with similar OD information. With the development of deep learning, Jindal *et al.* [20] proposed the spatiotemporal neural network (ST-NN) by simply applying a multilayer perceptron. The ST-NN is able to consider both the predicted distance and the travel time of a given path [20]. Based on this, Li *et al.* [10] developed a more diversified network structure and a more complex multitask learning framework to propose a new state-of-the-art path-based model named MURAT. This model utilizes the representation learning framework to extract meaningful features from path information and achieves clear improvements over traditional models [10]. The main drawback of path-based methods is that they cannot model complex traffic conditions and uncertainty over the entire path, including traffic lights and route selection [13].

The second category is route-based methods, which estimate the travel time by aggregating information for each road segment. Because they have better performance than path-based methods, route-based methods have been widely applied in various ride-hailing platforms [21]. According to different representation forms of the route, such as images [3], point

sequences [13] and embedding vectors [10], different kinds of neural networks have been utilized to solve the ETA problem. For example, Fu and Lee regarded the route of a vehicle as a sequence of "generalized images" and proposed a convolutional neural network-based model named DeepIST to extract spatiotemporal patterns among trips [3]. Wang *et al*. [13] presented a geoconvolution operation combined with a recurrent neural network (RNN) to capture the travel time information from GPS point sequences. To maximize the extraction of route information, various graph convolution neural networks (GNNs) have been established to learn the embedding of each segment [9], [22]. Based on a meaningful embedding, Wang *et al*. [21] proposed a wide-deep-recurrent (WDR) learning model that significantly outperforms the other models and was deployed on Didi Chuxing's platform. However, due to the scale of real-world urban road networks (millions of vertices and billions of edges), building and learning such a hyperscale GNN has become challenging. Additionally, for trips of varying length, it is necessary to analyze the multiple correlations between each segment and focus on the main segment.

For the learning problem of global factors, the existing ETA studies mainly have two strategies. One is to design various representation methods and transform global factors to numeric vectors [6], [12], [13], [21], [23], and the other is to combine an auxiliary task with the ETA model to form an end-to-end multitask learning framework [4], [10]. The former strategy lacks the ability to extract the intercorrelations between the route and global features and cannot estimate the travel time in a specific scenario. The latter cannot learn the features of multiple heterogeneous data in parallel. None of the above approaches are optimized for data sparseness problems.

In conclusion, the common weaknesses of existing works are as follows: 1) Strong representation and learning methods for heterogeneous sparse data are needed. 2) Intercorrelations and intracorrelations between various factors are not captured.

III. Methodology

A. Problem Statement

In this work, we propose a hybrid framework to estimate the travel time that is required for a vehicle to traverse a path of the road network with route and environmental information. To formalize the ETA problem using mathematical formulas, we give the definitions below. In particular, we use nonbold letters (e.g., x) to denote scalars. We use bold capital letters (e.g., $\boldsymbol{X}$) to represent matrices and bold lowercase letters (e.g., $\boldsymbol{x}$) to represent vectors. For tensors, we utilize calligraphic letters (e.g., $\mathcal{X}$).

1) Road Network: We define the road network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i | 0 \leq i \leq M\}$ is a vertex set representing each road segment in the road network. $\mathcal{E}$ is an edge set that denotes the connectivity among vertices. $e_{ij} = 1$ if and only if vertices v_i and v_j are directly connected.

2) Route: Route $r^{(i)}$ is a vertex sequence $r^{(i)} = [v_1, v_2, \ldots, v_{l_i}]$ with variable length l_i. Usually, $r^{(i)}$ contains dozens to hundreds of vertices.

TABLE I
Explanation of Different Data Representations

data	method	feature	information
router$^{(i)}$	trans2vec	sequential features	Vehicle transfer behaviors
global factors $gf^{(i)}$	temporal encoding (daily)	embedding features	Departure time information with different periods
	temporal encoding (weekly)	embedding features	
	one-hot encoding	categorical features	Memorization information
	real-valued embedding	embedding features	Generalization information

3) Global Factors: A set of global factors of route $r^{(i)}$ are denoted $gf^{(i)} = \{time^{(i)}, weather^{(i)}, driverID^{(i)}, \ldots\}$, which represents various geographic or traffic environmental factors that affect travel time. Data containing this information are usually heterogeneous and sparse.

4) Trip: Based on the above definitions, trip $x^{(i)}$, where $0 \leq i \leq N_T$, can be represented as $x^{(i)} = \{r^{(i)}, gf^{(i)}, \tau^{(i)}\}$, where $\tau^{(i)}$ is the travel time of the trip and N_T denotes the number of trips.

Our task is to construct a sophisticated feature set $\mathcal{F}(x^{(i)})$ from data $x^{(i)}$ and to map the features to the estimated travel time $\widetilde{\tau^{(i)}}$ with the minimal error $|\tau^{(i)} - \widetilde{\tau^{(i)}}|$. In the remainder of this section, the details of HSETA are presented.

B. Feature Representation of Heterogeneous and Sparse Data

In this section, we detail the $\mathcal{F}(\cdot)$ methods responsible for extracting rich features from trip $x^{(i)}$. To address the heterogeneity of data, we utilize a variety of feature representation methods, and for the sparsity of the data, we develop a multilevel representation method. We summarize the data into five types: route data, temporal data, traffic data, climate data and personalized data. The last three are global factors and have different degrees of sparseness. Additionally, we summarize the features that are extracted into three types: sequential features, embedding features and categorical features. The relationships of the different data, features and representation methods are shown in Table I.

The sequential features $\mathcal{F}(r^{(i)})$ are extracted from route $r^{(i)}$, and they represent the vehicle transfer behaviors and play a core role in determining the ETA. There are several sequential feature representation methods in existing works, including raw GPS points [13], subpaths [7], images [3] and GNN-based embeddings [9]. To maximize the amount of information while considering computational efficiency, we develop an improved node2vec embedding method called trans2vec. First, we map the vehicle trajectory onto the road network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and obtain a set of routes $\{r^{(i)} = [v_1, \ldots, v_{l_i}] | 0 \leq i \leq N_T\}$, where $v_j \in \mathcal{V}$ [24]. Then, we calculate two types of matrices $\boldsymbol{P}^T$ and $\boldsymbol{P}^M$:

$$\boldsymbol{P}^T = diag(\boldsymbol{E}\boldsymbol{1})^{-1}\boldsymbol{E}$$

where

$$\boldsymbol{E} = \begin{pmatrix} e_{11} & \cdots & e_{1M} \\ \vdots & \ddots & \vdots \\ e_{M1} & \cdots & e_{MM} \end{pmatrix} \boldsymbol{P}^M = diag(\boldsymbol{\Gamma}\mathbf{1})^{-1}\boldsymbol{\Gamma}$$

where

$$\boldsymbol{\Gamma} = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1M} \\ \vdots & \ddots & \vdots \\ \gamma_{M1} & \cdots & \gamma_{MM} \end{pmatrix} \tag{1}$$

γ_{ij} denotes the number of vehicles that move from segment v_i to v_j according to $\{r^{(i)}\}$. $\boldsymbol{P^T}$ and $\boldsymbol{P}^M$ are the transition matrices of the adjacency graph and Markov chain of $\mathcal{G}$, respectively. Next, we utilize breadth-first sampling and depth-first sampling to generate random walk sequences based on the probabilities of $\boldsymbol{P}^T$ and $\boldsymbol{P}^M$, respectively. Finally, the embedding vector of each road segment $\mathcal{F}(v_j)$ can be learned by the skip-gram architecture using the generated sequences. The details of the sampling strategy are the same as those of node2vec [25]. The advantage of trans2vec is that it considers the topological characteristics and vehicle transfer characteristics simultaneously. A trip is composed of numerous road segments; therefore, $\mathcal{F}(r^{(i)})$ is denoted sequence $\mathcal{F}(r^{(i)}) = [\mathcal{F}(v_1), \ldots, \mathcal{F}(v_{l_i})]$.

The features of temporal data are extracted from the departure time of the trip. We introduce the temporal encoding method [26] from the transformer structure to learn the embedding vectors of temporal features $\mathcal{F}(time)$ from multiple scales, such as daily features $\mathcal{F}^{(d)}(time^{(i)})$ and weekly features $\mathcal{F}^{(w)}(time^{(i)})$. $\mathcal{F}(time^{(i)}) = \{\mathcal{F}^{(d)}(time^{(i)}), \mathcal{F}^{(w)}(time^{(i)})\}$ can be calculated as follows:

$$\begin{aligned} \mathcal{F}^{(scale)}(time)(pos_{scale}, 2j) &= sin(pos/10000^{\frac{2j}{h}}) \\ \mathcal{F}^{(scale)}(time)(pos_{scale}, 2j+1) &= cos(pos/10000^{\frac{2j}{h}}) \end{aligned} \tag{2}$$

where pos is the position index of $time^{(i)}$ in the period of the corresponding scale, j is the *j-th* dimension of the position embedding and h is the dimension of $\mathcal{F}^{(scale)}(time)$. When $scale$ is "daily", pos_{daily} indicates the hour in the day, and when $scale$ is "weekly", pos_{weekly} indicates the day in the week.

The features of climate data (i.e., weather and temperature), personalized data (i.e., driver ID) and traffic data (i.e., state) are binarized sparse features. We use multilevel representation methods to encode the same feature into categorical vectors and embedding vectors. This multilevel method ensures that HSETA can extract the generalization information and memorization information simultaneously. Categorical vectors can be obtained by one-hot encoding. For example, the binary element "weather=cloudy" has a value of 1 if the trip occurred on a cloudy day. The embedding vector is a real-valued vector that is initialized randomly for each category. The high-dimensional sparse vector (categorical vector) and low-dimensional dense vector (embedding vector) for one data point can fit the design of HSETA to learn the generalization and memorization information, respectively [27].

C. Hybrid Feature Learning Strategy

Learning heterogeneous features (e.g., sequences, dense vectors, sparse vectors) is also challenging for ETA framework design. In this section, we design a hybrid framework that integrates various structures to extract multiple pieces of information from heterogeneous features. Specifically, the hybrid framework consists of three parts: sequential feature learning, embedding feature learning and categorical feature learning.

1) Sequential Feature Learning: For sequential features, i.e., the route feature $\mathcal{F}(r^{(i)}) = [\mathcal{F}(v_1), \ldots, \mathcal{F}(v_{l_i})]$, we utilize the gated recurrent unit (GRU) structure [28] to control the information flow. The GRU is a variant of recurrent neural networks (RNNs), which have achieved great success in several sequential data learning tasks, such as natural language processing (NLP) [29]. The advantage of GRU is that it can deal with variable-length sequences and can also capture long-term dependencies along sequences when some trips travel over 600 road segments in real-world traffic datasets. The advantages of GRU fit the characteristics of the sequential features $\mathcal{F}(r^{(i)})$ in the ETA task.

Taking the t-th input ($\boldsymbol{x}_t$) of the sequence feature as an example, the specific calculation process of the GRU is shown below:

$$\boldsymbol{z}_t = f(\boldsymbol{W}_z\boldsymbol{x}_t + \boldsymbol{U}_z\boldsymbol{h}_{t-1}) \tag{3}$$

$$\boldsymbol{i}_t = f(\boldsymbol{W}_i\boldsymbol{x}_t + \boldsymbol{U}_i\boldsymbol{h}_{t-1}) \tag{4}$$

$$\boldsymbol{c}_t = f(\boldsymbol{W}_c x_t + \boldsymbol{U}_c(\boldsymbol{i}_t \odot \boldsymbol{h}_{t-1})) \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (\mathbf{1} - \boldsymbol{z}_t) \odot \boldsymbol{c}_t \tag{6}$$

where the operator $\odot$ indicates the Hadamard product and matrices $\boldsymbol{W}_z$, $\boldsymbol{W}_i$, $\boldsymbol{W}_c$, $\boldsymbol{U}_i$, $\boldsymbol{U}_z$, and $\boldsymbol{U}_c$ denote the learnable parameters. f is the nonlinear activation function. The update gate ($\boldsymbol{z}_t$ in equation 3) and the reset gate ($\boldsymbol{i}_t$ in equation 4) in the GRU determine how much of the past information $\boldsymbol{h}_{t-1}$ needs to be remembered or forgotten. The single output $\boldsymbol{h}_t$ at current step t is obtained by equation 6 based on the current memory content $\boldsymbol{c}_t$. After one loop of the whole sequence, a set consisting of all outputs $[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots]$ represents the learned knowledge from the sequential feature.

2) Embedding Feature Learning: Embedding vectors are low-dimension and dense vectors that represent generalization information about sparse global factors. Based on this characteristic, the multilayer perceptron (MLP), which is composed of multiple layers of neurons, is suitable for feature learning. In one layer of the MLP, each neuron uses a nonlinear activation function f to control the information flow:

$$\boldsymbol{h}^{(l+1)} = f(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l)} + \boldsymbol{b}^{(l)}) \tag{7}$$

where l denotes the current number of layers and $\boldsymbol{h}^{(l)}$ represents hidden features of the corresponding layer. When $l = 1$, $\boldsymbol{h}^{(l)}$ indicates the original embedding features. $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are the learnable parameters of the MLP.

3) Categorical Feature Learning: Categorical vectors are high-dimensional vectors that represent the memorization of sparse global factor information. Note that one global factor usually interacts with another global factor to affect the ETA. For example, some drivers who are cautious drive slower on rainy days, while some experienced drivers maintain a similar

speed regardless of the weather. Based on this, we utilize a factorization machine (FM) [14] to design an EFMB to handle categorical feature learning issues. First, the factorization machine for a single categorical vector $\boldsymbol{x}$ can be calculated as follows:

$$h = w_0 + \sum_{i=1}^{k_x} w_i x_i + \sum_{i=1}^{k_x} \sum_{j=i+1}^{k_x} \boldsymbol{s}_i{}^T \boldsymbol{s}_j x_i x_j \quad (8)$$

where w_i and x_i represent the i-th dimension of vectors $\boldsymbol{w}$ and x and $\boldsymbol{s}_i$ represents the i-th row of the matrix $\boldsymbol{S} \in \mathbb{R}^{k_x \times k_c}$. k_c denotes the dimension of $\boldsymbol{s}_i$, while k_x denotes the dimension of $\boldsymbol{x}$. $\boldsymbol{w}$ and $\boldsymbol{S}$ are the learnable parameters of the FM, where $\boldsymbol{w}$ is responsible for extracting generalization information and $\boldsymbol{S}$ is responsible for extracting memorization information. Note that we have already designed a more professional representation and learning method to obtain generalization information; therefore, we can simplify equation 8 to

$$h = \sum_{i=1}^{k_x} \sum_{j=i+1}^{k_x} \boldsymbol{s}_i{}^T \boldsymbol{s}_j x_i x_j \quad (9)$$

To improve computational efficiency and facilitate integration with other parts, we rewrite equation 9 as a matrix formulation:

$$h = \frac{1}{2}[\boldsymbol{x}^T \boldsymbol{S} \odot \boldsymbol{x}^T \boldsymbol{S} - \mathbf{1}^T (diag(\boldsymbol{x})\boldsymbol{S} \odot diag(\boldsymbol{x})\boldsymbol{S})]\mathbf{1} \quad (10)$$

where $\mathbf{1}$ is a vector of all ones. Then, based on the theory in machine learning that "*using multiple learning algorithms can obtain better predictive performance*" [30], we integrate multiple FMs to design an EFMB that is similar to the MLP in the second part. Let tensor $\mathcal{S} \in \mathbb{R}^{d_{fm} \times k_x \times k_c}$ denote the learnable parameters; we can then derive the definition of the ensemble factorization machine block:

$$\boldsymbol{h} = \frac{1}{2}\{\boldsymbol{x}^T \Delta \mathcal{V} \odot \boldsymbol{x}^T \Delta \mathcal{V} - \mathbf{1}^T [(diag(\boldsymbol{x})\Delta \mathcal{S}) \odot (diag(\boldsymbol{x})\Delta \mathcal{S})]\}\mathbf{1} \quad (11)$$

Δ denotes the matrix product, which can be broadcast in the first dimension of $\mathcal{S}$. $\boldsymbol{h} \in \mathbb{R}^{d_{fm}}$ is a vector representing the output of the EFMB, and d_{fm} is the hyperparameter that denotes the hidden units of the EFMB.

We combine sequential feature learning, embedding feature learning and categorical feature learning to construct the basic HSETA framework to extract multiple pieces of information from heterogeneous and sparse data to construct a complex traffic environment.

D. Multiple-Correlation Learning Strategy for Information Aggregation

Intracorrelations indicate associations between sequential features (i.e., road segments), which can push a model to pay attention to the core feature or the core subpath. When a trip consists of more segments, intracorrelation extraction is more important. Intercorrelations indicate the associations between the route features and the global features. For example, on Friday afternoons, the roads near a school become more congested, which ultimately affects the intracorrelations among the road segments. We can refine the law of the multiple-correlation effect process as {intercorrelation→intracorrelation→ETA}; therefore, the multiple information aggregation process can be regarded as taking the weighted sum of sequential features based on the intracorrelations, where intracorrelations are calculated from intercorrelations among the route features and the global features. In addition, intracorrelations also represent the associations among the global factors; however, this has already been addressed by the proposed EFMB. Based on the mechanism of multiple correlations, we propose the MCLB for extraction.

First, we utilize conditional probability $p(\mathcal{F}(v_j)_c | \mathcal{F}(gf^{(i)})_q)$ to express the intercorrelations between the route features $\mathcal{F}(v_1), \ldots, \mathcal{F}(v_{l_i})$ and the global features $\mathcal{F}(gf^{(i)})$ in trip $x^{(i)}$. Note that $\mathcal{F}(v_j)_c$ and $\mathcal{F}(gf^{(i)})_q$ are the vectors that are linearly transformed from $\mathcal{F}(v_j)$ and $\mathcal{F}(gf^{(i)})$. Based on this, for the whole route $\mathcal{F}(r^{(i)})$ of trip $x^{(i)}$, the intracorrelations under the intercorrelations and the results of information aggregation can also be calculated as expectation $\mathbb{E}_{p(\mathcal{F}(r^{(i)})|\mathcal{F}(gf^{(i)})_q)}[\mathcal{F}(r^{(i)})]$ with conditional probability $p(\mathcal{F}(r^{(i)})|\mathcal{F}(gf^{(i)})_q)$ [31]:

$$\mathbb{E}_{p(\mathcal{F}(r^{(i)})|\mathcal{F}(gf^{(i)})_q)}[\mathcal{F}(r^{(i)})] = \sum_{j \in [1,l_i]} \frac{k(\mathcal{F}(gf^{(i)})_q, \mathcal{F}(v_j)_c)}{\sum_{u \in [1,l_i]} k(\mathcal{F}(gf^{(i)})_q, \mathcal{F}(v_u)_c)} \mathcal{F}(v_j)_c \quad (12)$$

where $k(\cdot,\cdot)$ is a function that is always positive. Here, we refer to [32] to define $k(\cdot,\cdot)$ as

$$k(\mathcal{F}(gf^{(i)})_q, \mathcal{F}(v_j)_c) = e^{(\mathcal{F}(gf^{(i)})_q{}^T \mathcal{F}(v_j)_c)/\sqrt{d_c}} \quad (13)$$

$\mathcal{F}(v_j)_c$ and $\mathcal{F}(gf^{(i)})_q$ can be calculated by $\mathcal{F}(v_j)_c = \mathcal{F}(v_j)\boldsymbol{W}_c$ and $\mathcal{F}(gf^{(i)})_q = \mathcal{F}(gf^{(i)})_q \boldsymbol{W}_q$ with learnable parameters $\{\boldsymbol{W}_c, \boldsymbol{W}_q\}$. d_c indicates the dimension of $\mathcal{F}(v_j)_c$, and $\mathcal{F}(gf^{(i)})_q$ also represents the hidden size of this block. Note that the formula of $\mathbb{E}_{p(\mathcal{F}(r^{(i)})|\mathcal{F}(gf^{(i)})_q)}[\mathcal{F}(r^{(i)})]$ is similar to the simplified attention mechanism [32], so we also call these MCLBs attention-based expectation calculation blocks.

E. Heterogeneous and Sparse Data Learning Hybrid Framework

Finally, we propose the heterogeneous and sparse data learning hybrid framework (HSETA) based on the schemes mentioned above to solve the learning problems of heterogeneous sparse data and multiple correlations. The details of the HSETA structure are illustrated in Fig. 2.

First, the input data are divided into route factors and global factors, and various representation methods are applied to extract sequential features, dense features, and sparse features from them. After feature extraction, an end-to-end hybrid learning strategy that contains MLP, EMFB, GRU and MCLB neural structures is co-trained. Finally, a fully connected (FC) layer is utilized to output the ETA results.

During the training phase, we use the mean absolute percentage error (MAPE) as our loss function. The MAPE is a relative error and can ensure that the model obtains high-precision results for both short paths and long paths.

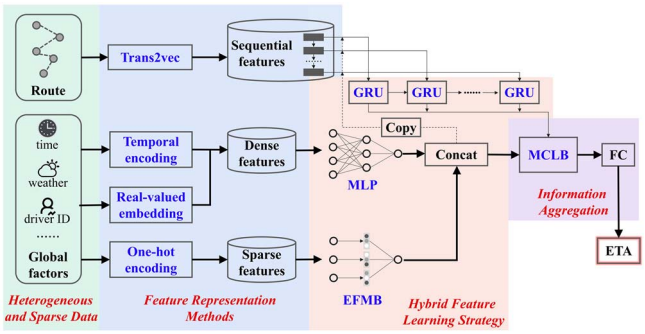


Fig. 2. HSETA architecture.

TABLE II
EXPLANATION OF DIFFERENT DATA REPRESENTATIONS

Dataset	# of trips	# of links	Distance mean (km)	Travel time mean (sec)	Average links per trip
Shenzhen	1,048,576	8,651,005	5.265	834.14	86.98
Xian	312,898	1620	1.444	261.32	7.28

The MAPE can be calculated as

$$loss = \frac{100\%}{N_T} \sum_{i=1}^{N_T} \left| \frac{\widetilde{\tau^{(i)}} - \tau^{(i)}}{\tau^{(i)}} \right| \tag{14}$$

where $\widetilde{\tau^{(i)}}$ and $\tau^{(i)}$ represent the observed and predicted values of travel time.

IV. EXPERIMENTS AND ANALYSIS

A. Datasets

To verify the effectiveness and superiority of our proposed HSETA framework, we use the HSETA and baseline models on two real-world traffic datasets:

(1) Shenzhen Dataset: The Shenzhen dataset consists of 8,651,005 probe car trips in Shenzhen from August 1 to August 31, 2020, on the Didi platform. This is a large-scale dataset that contains a variety of travel patterns. The shortest trip contains only 3 road segments, while the longest trip contains over 900 road segments. This is a great challenge for the learning ability of the ETA model.

(2) Xi'an Dataset: Compared to the Shenzhen dataset, the Xi'an dataset is a smaller-scale dataset, and it contains only 312,898 probe car trips on the Didi platform. In addition, short trips account for the majority of trips. The original data are implemented as raw GPS records, so we utilize the ST-matching method to constrain the GPS points to the road network and generate the route sequences [24].

Both datasets are associated with the corresponding global factors, including the timestamp, weather, driverID, traffic status, and temperature. The details of the datasets are illustrated in Table II. The raw data are available on the GAIA Open Dataset website [33].

B. Baselines and Evaluation Metrics

The baseline methods include three types: historical average-based models, including simple averages and real-time averages. The second type is machine learning-based methods, which perform well in many traditional traffic tasks. The last and most competitive type is deep learning-based methods, which have achieved outstanding performance in ETA prediction in the last few years. The details of the baseline methods for comparison are listed as follows:

(1) **AVG**: Simple average method. We use the average travel time of each road segment during a specific time interval for estimation. For a given trip, the corresponding travel time is calculated by summing the average travel time of each road segment.

(2) **RealTimeAVG**: Real-time average method. Currently, digital map apps (e.g., Google Maps and Baidu Maps) can provide real-time traffic information for every road segment. This model simulates the scenario where real-time traffic information is used to solve the ETA issue by averaging the real-time travel time of each road segment at the time of departure.

(3) **GBDT**: The gradient boosting decision tree [34] is one of the most powerful machine learning methods and is widely applied in practice. In the ETA task, the GBDT maintains the same input features as HSETA. However, the GBDT cannot handle variable-length sequences, and we uniformly sample each trip's route sequence to a fixed length of 100.

(4) **MlpETA**: Multiple-layer perceptron for the ETA task. As the basic baseline of deep learning methods, we utilize a 5-layer perceptron with the ReLU activation function and dropout to construct MlpETA. The inputs of MlpETA remain the same as those of the GBDT, and the number of hidden units is set to 128.

(5) **DeepTTE**: Deep travel time estimation (DeepTTE) [13] is an outstanding method for large-scale ETA using deep learning technology. The geo-convolution operation and long short-term memory (LSTM) are utilized in DeepTTE to track spatial and temporal dependence.

(6) **WDR**: The wide-deep-recurrent learning model [21] is a state-of-the-art ETA neural network that outperforms the best methods in the original experiments. WDR utilizes a wide model, a deep model and LSTM to extract features from global statistical information and local traffic information. Compared to our HSETA, the main difference is that HSETA can learn the correlations between various features and road segments.

(7) **FMA-ETA**: A deep learning-based framework mainly composed of a feed-forward network and multifactor self-attention rather than an RNN structure [12]. On a massive real-world traffic dataset, this framework showed comparable estimation precision in the original paper.

For a comprehensive quantitative measurement of estimation error, we adopt three classic metrics, the MAPE, root-mean-square error (RMSE) and mean average error (MAE), which are widely used in ETA tasks. They are

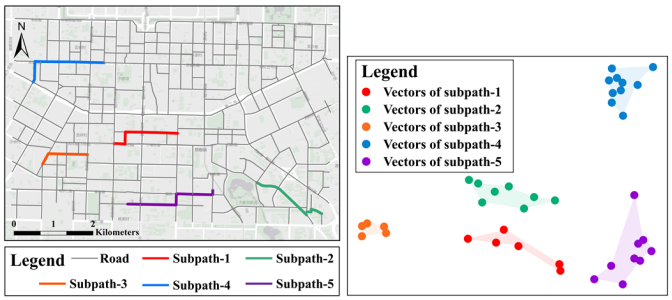


Fig. 3. Visualization of the subpaths and their corresponding embedding vectors. (a) Randomly selected subpaths from different trips in the Xi'an dataset. (b) Corresponding embedding vectors that are projected into 2-D space by PCA.

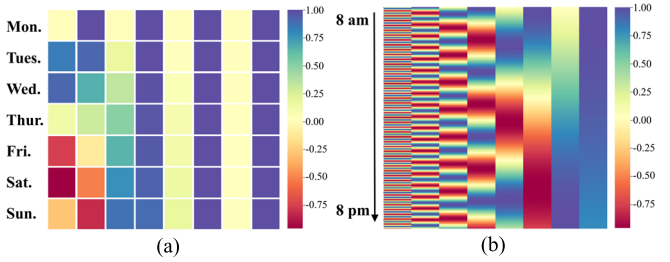


Fig. 4. Visualization of the temporal data representations at different scales. (a) Temporal encoding at the daily scale, where each row represents the embedding vector of the specific day of the week. (b) Temporal encoding at the weekly scale, where each row represents the embedding vector of the specific moment of the day.

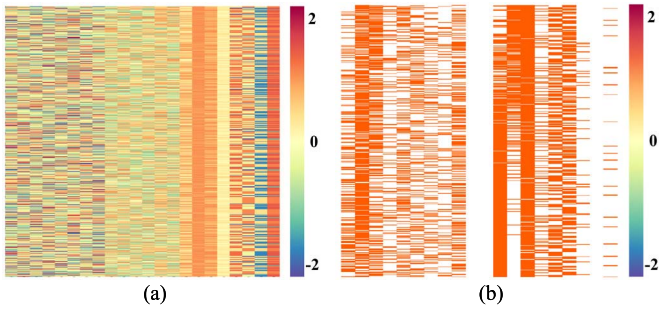


Fig. 5. Visualization of the (a) embedding matrix and (b) categorical matrix in the same mini-batch.

calculated as follows:

$$MAPE(\widetilde{\tau^{(i)}}, \tau^{(i)}) = \frac{100\%}{N_T} \sum_{i=1}^{N_T} \left| \frac{\widetilde{\tau^{(i)}} - \tau^{(i)}}{\tau^{(i)}} \right| \quad (15)$$

$$MSE(\widetilde{\tau^{(i)}}, \tau^{(i)}) = \sqrt{\frac{1}{N_T} \sum_{i=1}^{N_T} (\widetilde{\tau^{(i)}} - \tau^{(i)})^2} \quad (16)$$

$$MAE(\widetilde{\tau^{(i)}}, \tau^{(i)}) = \frac{1}{N_T} \sum_{i=1}^{N_T} \left| \widetilde{\tau^{(i)}} - \tau^{(i)} \right| \quad (17)$$

C. Experimental Settings

We set the hyperparameters of all the deep learning-based methods according to the configuration with the best performance in the 6-fold cross-validation. For HSETA, the dimension of the embedding is set to 32, and the hidden sizes of the MLP, GRU and MCLB are all set to 128. For the EFMB, the number of hidden units is set to 8.

During the training phase of the deep learning-based methods, we utilize the Adam [34] optimizer and adaptive learning rate mechanism, which warmup at the beginning of training and decrease at the end of training [32]. The learning rate can be calculated as

$$lr = d_{model}^{-0.5} \times min(step^{-0.5}, step \times warmup^{-1.5}) \quad (18)$$

where $step$ denotes the number of training steps. d_{model} and $warmup$ are set to 64 and 20, respectively. We implement our models on one machine with an NVIDIA RTX3090 GPU by the PyTorch library [35]. The source code and sample data are available at https://github.com/LouisChenki/HSETA.

D. Heterogeneity and Sparse Data Processing

Before performance evaluation, we show the processing of heterogeneity and sparse data in the experiment from two aspects: data representation and information aggregation.

There are five types of data in our real-world traffic datasets: route data, temporal data, traffic data, climate data and personalized data. For route data, which represent the road segments traveled by vehicles, we utilize the proposed trans2vec embedding method for encoding. As illustrated in Fig. 3, we randomly select six subpaths of real trips in the Xi'an dataset and visualize the corresponding embedding vectors, which are projected into 2-D space by principal component analysis (PCA). The relative positions of the embedding vectors are highly correlated with the composition of the subpaths. Specifically, the distance between the vectors of road segments in the same subpath is smaller than that of road segments in different subpaths. This demonstrates that the proposed trans2vec method can effectively represent the route information.

For temporal data, two 8-dimensional embedding vectors are obtained from the daily scale and weekly scale. As illustrated in Fig. 4, the weekly scale encoding on the left indicates the specific day of the week, while the daily scale encoding on the right indicates the specific moment of the day. This temporal encoding method ensures that each trip is precisely positioned on its time coordinate during the HSETA calculation.

Traffic data, climate data and personalized data have different degrees of sparseness. Thus, we utilize the proposed multilevel representation methods to encode each item of these data into embedding vectors and categorical vectors simultaneously. In Fig. 5, we visualize two matrices composed of the above two vectors in the same mini-batch. Obviously, the embedding matrix is dense and full of continuous values, while the categorical matrix is sparse, and its composition is either 0 or 1. The two distinct matrices can fit the design of our data learning hybrid framework to extract the individual information and correlation information.

TABLE III
PERFORMANCE COMPARISON

	Shenzhen			Xi'an		
	MAPE (%)	RMSE (sec)	MAE (sec)	MAPE (%)	RMSE (sec)	MAE (sec)
AVG	28.19%	322.57	243.52	30.99%	147.85	75.98
RealTimeAVG	16.16%	197.44	132.10	39.41%	132.02	73.97
GBDT	22.33%	248.61	169.06	40.55%	109.23	64.32
MlpETA	14.45%	578.41	349.00	10.37%	133.16	75.06
DeepTTE	13.60%	153.61	106.80	8.82%	21.15	16.12
WDR	13.86%	155.33	108.42	6.56%	19.86	16.14
FMA-ETA	23.44%	283.47	204.33	26.05%	77.70	61.78
HSETA	**12.80%**	**141.90**	**99.33**	**5.31%**	**16.55**	**12.31**

After data representation, data fusion is further demonstrated based on the design of the MCLB module proposed in HSETA. We visualize the weights of sequential features, which represent the relative scores between road segments in a trip in the multiple correlation. As shown in Fig. 6, the rectangles with different colors represent different road segments in one trip, and the length of the rectangle represents its corresponding weight. When we only adjust a specific factor (drive ID in Fig. 6a, weather in Fig. 6b, and period in Fig. 6c) of global features, the weights of each road segment change accordingly. This result shows that our framework can adaptively adjust the data fusion process according to the different traffic environments.

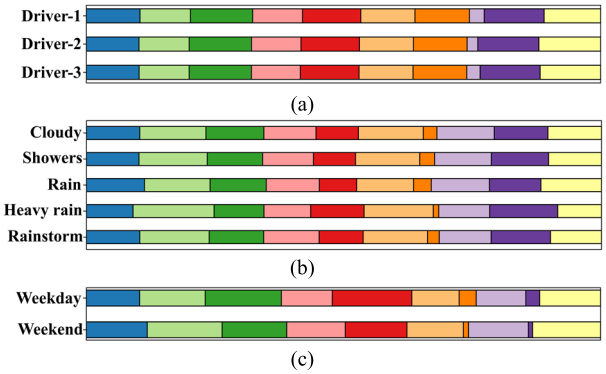


Fig. 6. Learned weights of road segments in a randomly selected trip with different specific factors of global features. (a) Weight distribution changes when adjusting for drive-ID. (b) Weight distribution changes when adjusting for weather. (c) Weight distribution changes when adjusting for departure time.

E. Experimental Results

To demonstrate the advantages of HSETA, we first evaluate the performance of all methods for the ETA task on both datasets. The results are reported in Table III, and the bolded and underlined numbers indicate the best performance among all methods and among baseline methods, respectively. The historical average-based methods (e.g., AVG and RealTimeAVG) have an obvious gap compared to other methods, which means that traffic information alone cannot indicate the travel time. Since the scale of the Xi'an dataset is smaller, this gap phenomenon is particularly prominent. GDBT and MlpETA seem to have achieved a good result in the MAPE metric on the Shenzhen dataset but did not perform as well as expected in the RMSE and MAE metrics. We speculate that these models cannot handle variable-length data, so the longer the trip is, the greater the absolute error of the estimation, and the MAPE, which is based on percentage calculations, cannot capture this phenomenon. Among the rest of the baselines, the RNN-based methods (e.g., DeepTTE and WDR) outperformed the attention-based method (FMA-ETA) and achieved much better results than the above traditional methods. This shows that the integration of traffic information and environmental information is essential for accurate ETA prediction, and the RNN can extract path information better than the attention mechanism.

HSETA is obviously better than all baseline methods. The error rates of HSETA on the two datasets are only 12.80% and 5.31%. In particular, on the Shenzhen dataset, our method is the only method that reduces the MAE to less than 100. This means that the structure of our model is more reasonable for the ETA prediction task. To further illustrate the details of the superiority of HSETA in multiple dimensions, we plot the MAPE curves for different trip lengths in Fig. 7. In most cases, HSETA, DeepTTE and WDR present the same trend, in which the MAPE decreases as the distance or segment number grows. They maintain similar accuracies at different travel times. However, FMA-ETA shows the opposite trend, in which the MAPE increases as the distance, travel time or segment number grows. We speculate that the self-attention mechanism in FMA-ETA cannot handle long trip sequences compare to the similar attention-based expectation calculation block (MCLB) in HSETA. Overall, our model maintains the smallest error in most cases.

Regarding the temporal dimension, the error distributions of the methods in different time periods are illustrated in Fig. 8. In general, HSETA always maintains the best performance in all temporal dimensions. During the morning peaks, evening peaks, weekdays and weekends, our HSETA can produce more stable ETA results than baseline methods, which can satisfy the needs of users to travel at any time. We further analyze the error distributions for trips that depart within the above periods in detail. As illustrated in Fig. 9, the signed MAPE, which can describe whether the ETA is greater than or less than the ground truth, is utilized to visualize the distributions of errors. The curves of DeepTTE, WDR and HSETA are similar to those of a normal distribution with a mean of 0, which means that these methods can produce nearly unbiased

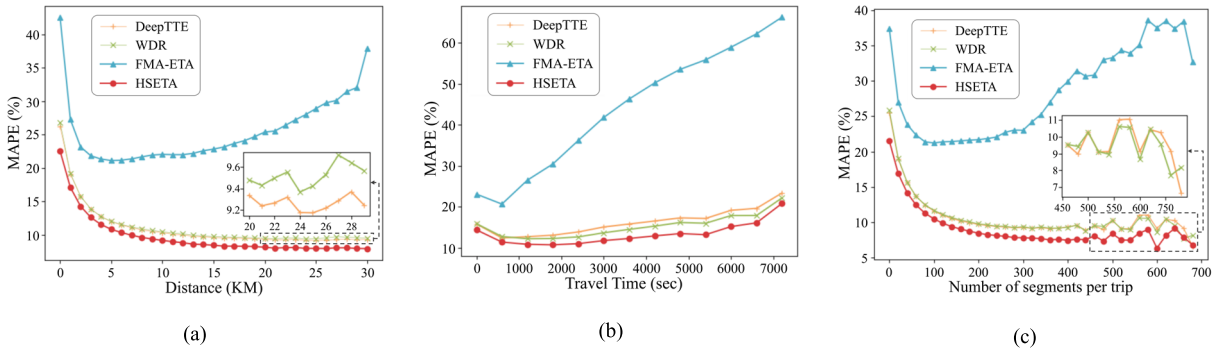


Fig. 7. Details of the comparison in multiple dimensions of trip length. (a) MAPE vs. distance. (b) MAPE vs. travel time. (c) MAPE vs. segment number.

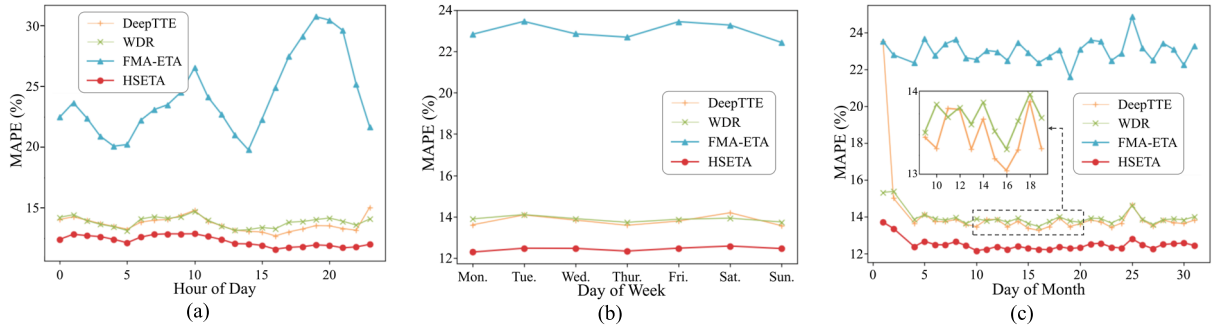


Fig. 8. Details of the comparison in multiple dimensions of time. (a) Hour-of-day interval. (b) Day-of-week interval. (c) Day-of-month interval.

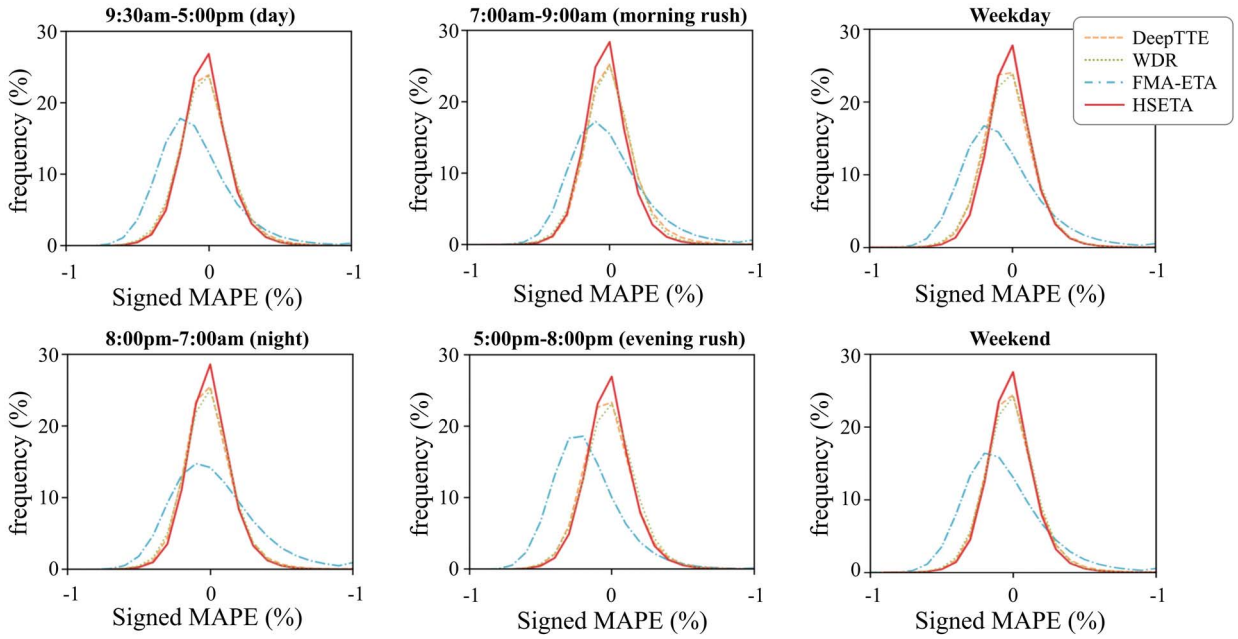


Fig. 9. Distributions of prediction errors for trips that depart within different periods.

estimations of the travel time, while FMA-ETA always underestimates it.

In addition to the accuracy, the inference speed is important in applications. According to statistics, the average number of orders per second per city on the Didi Chuxing platform is approximately 289 [36]. Therefore, we randomly sampled 289 samples at each sequence length for HSETA, WDR and DeepTTE. The results are illustrated in Fig. 10. The dotted line represents the trend of the inference time with the trip length, while the solid line represents the error interval of repeated experiments. Our HSETA always maintains a high inference speed; although it is not the fastest, it is the most stable, and it meets the requirements of real-time estimation in a real city. Note that due to a large number of self-attention

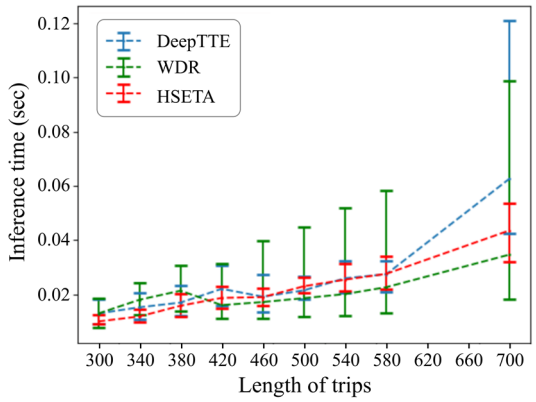


Fig. 10. Inference time per 289 trips for HSETA, WDR and DeepTTE.

blocks, FMA-ETA's inference speed on a single GPU (no parallelism) is significantly slower than that of other methods, so no comparison is made here.

F. Effect of the Framework Architecture

To better understand the effectiveness of the different components of HSETA in the ETA task, we further design some variant models that remove the GRU, MLP, EFMB and MCLB structures. The details of the variants are as follows:

(1) **HSETA-a**: Only the GRU structure in the proposed HSETA remains.
(2) **HSETA-b**: HSETA framework without the GRU structure.
(3) **HSETA-c**: HSETA framework without the MLP structure.
(4) **HSETA-d**: HSETA framework without the EFMB structure.
(5) **HSETA-e**: HSETA framework without the MCLB structure.

Table IV shows a comparison of the results of the variant models. As expected, the model with a complete structure obtained the best performance and had a maximum lead of 59%, 38.3%, and 40.3% in the three metrics. This fully demonstrates the rationality of the different component settings in HSETA. Specifically, HSETA-b, which removed the GRU structure, performs the worst and has a significant error gap compared to the other variants. This indicates that the route information is the most important for solving the ETA task. We can also observe that using only route information (HSETA-a) is not better than the variants that consider both route information and global factors. This means that comprehensively considering the overall urban traffic environment can achieve high-precision travel time estimation. Furthermore, we find that whether the sparsity of the data is addressed (HSETA-d) and whether multiple correlations of different features are calculated (HSETA-e) also affect the ETA accuracy. This means that using a reasonable method to aggregate the knowledge of heterogeneous sparse data for a complex traffic environment is the key to solving the ETA issue. This is exactly what HSETA achieved.

TABLE IV
PERFORMANCE COMPARISON BETWEEN HSETA AND ITS VARIANTS ON THE SHENZHEN DATASET

	MAPE (%)	RMSE (sec)	MAE (sec)
HSETA-a	15.91%	185.90	132.50
HSETA-b	20.40%	230.07	166.49
HSETA-c	14.28%	161.14	113.33
HSETA-d	13.09%	143.70	100.55
HSETA-e	13.46%	148.87	104.18
HSETA	**12.80%**	**141.90**	**99.33**

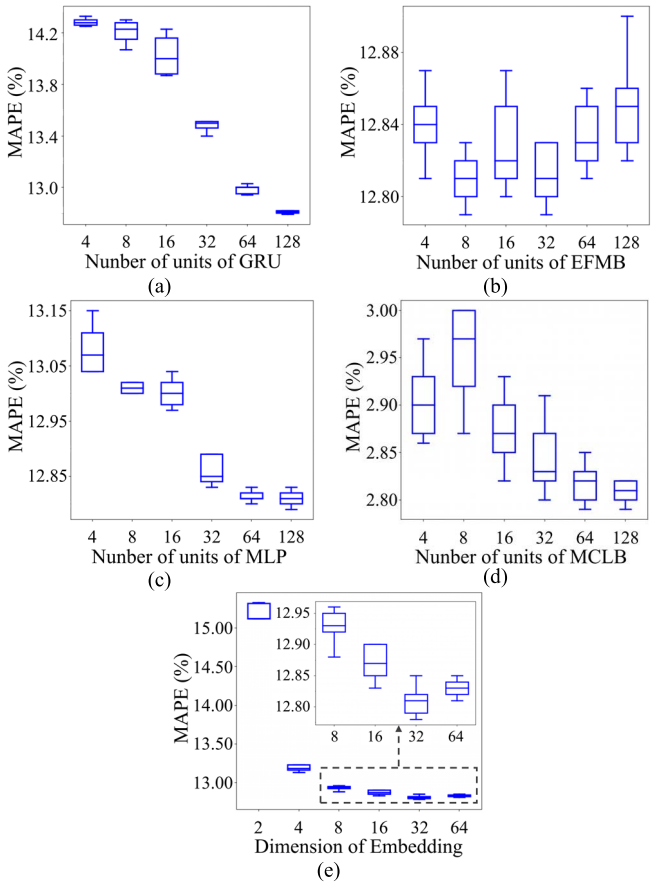


Fig. 11. How ETA error (MAPE) changes with different hyperparameter Settings. (a) GRU setting. (b) EFMB setting. (c) MLP setting. (d) MCLB setting. (e) Embedding setting.

G. Effect of Hyperparameter Settings

We further train 150 models on the Shenzhen dataset with different combinations of hyperparameters, including the number of hidden units of the GRU, the EFMB, MLP and MCLB and the dimension of the feature embedding. Due to the limitation of GPU memory, we set the range of the number of hidden units as [4, 128] and set the range of the embedding dimension as [2, 64].

Fig. 11 shows the results. In most cases, the error decreases when the number of hidden units/dimensions grows. This trend is most obvious in the hyperparameter changes of the GRU in Fig. 7a, which also verifies the judgment of the importance of route information in the previous section. Fig. 7b demonstrates the effect of the number of hidden units of the EFMB. Generally, there is no significant difference among the different

settings; when the number is 8, the error is the lowest and most stable. This is probably due to the 8 hidden units best matching the number of intracorrelation patterns among the sparse global factors. For the hyperparameter settings of the MLP and MCLB in Fig. 7c and d, we also observe a clear trend of decreasing error as the number of units grows. This is because for the ETA task, there is much information hidden in the intercorrelations between route features and global factors. Finally, Fig. 7e shows the effect of the embedding dimension. Note that there are sharp error decreases when the number of dimensions increases from 2 to 4. After that, the performance tends to stabilize. This phenomenon is consistent with the conclusions in another ETA study [10]. The above analysis can help us optimize the hyperparameter selection process and can also help us better understand the performance of HSETA.

V. Conclusion

In this paper, we proposed a heterogeneous and sparse data learning hybrid framework, namely, HSETA, to perform route-based ETA tasks. Our method can effectively extract various features from heterogeneous and sparse data to learn the comprehensive information of traffic environments. Additionally, our method achieves nonlinear mapping from aggregated information to travel time by calculating multiple correlations between various features. The experimental results show that our method significantly outperforms the other off-the-shell methods.

For future works, there are still some limitations in our method that need to be addressed. First, in addition to the multiple correlations between various factors, there is also spatial heterogeneity and spatial dependency between different regions or different cities. How to optimize the framework by considering these geographical properties has important practical significance for the deployment and application of the method. Moreover, the information extracted by our framework is also suitable for other transportation tasks. This motivates us to design a more general framework based on HSETA, which has the ability to solve multiple transportation tasks simultaneously.

Acknowledgment

This work was carried out in part using computing resources at the High Performance Computing Platform of Central South University.

References

[1] S. Maiti, A. Pal, A. Pal, T. Chattopadhyay, and A. Mukherjee, "Historical data based real time prediction of vehicle arrival time," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1837–1842.

[2] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–22, Mar. 2019, doi: 10.1145/3293317.

[3] T.-Y. Fu and W.-C. Lee, "DeepIST: Deep image-based spatio-temporal network for travel time estimation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 69–78.

[4] 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.*, early access, Dec. 9, 2021, doi: 10.1109/TITS.2020.3040386.

[5] W. Lan, Y. Xu, and B. Zhao, "Travel time estimation without road networks: An urban morphological layout representation approach," 2019, *arXiv:1907.03381*.

[6] N. Zygouras, N. Panagiotou, Y. Li, D. Gunopulos, and L. Guibas, "HTTE: A hybrid technique for travel time estimation in sparse data environments," in *Proc. 27th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2019, pp. 99–108.

[7] Y. Li, D. Gunopulos, C. Lu, and L. Guibas, "Urban travel time prediction using a small number of GPS floating cars," in *Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2017, pp. 1–10.

[8] R. Barnes, S. Buthpitiya, J. Cook, A. Fabrikant, A. Tomkins, and F. Xu, "BusTr: Predicting bus travel times from real-time traffic," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 3243–3251.

[9] H. Hong *et al.*, "HetETA: Heterogeneous information network embedding for estimating time of arrival," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2444–2454.

[10] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 1695–1704.

[11] K. Fu, F. Meng, J. Ye, and Z. Wang, "CompactETA: A fast inference system for travel time prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2020, pp. 3337–3345.

[12] Y. Sun *et al.*, "FMA-ETA: Estimating travel time entirely based on FFN with attention," 2020, *arXiv:2006.04077*.

[13] 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. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, Apr. 2018, pp. 1–8.

[14] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Sydney, NSW, Australia, Dec. 2010, pp. 995–1000.

[15] T. Zhang and G. Guo, "Graph attention LSTM: A spatiotemporal approach for traffic flow forecasting," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 190–196, Mar. 2022, doi: 10.1109/MITS.2020.2990165.

[16] Guo, Ge, and Wei Yuan, "Short-term traffic speed forecasting based on graph attention temporal convolutional networks," *Neurocomputing*, vol. 410, no. 14, pp. 387–393, Oct. 2020, doi: 10.1016/j.neucom.2020.06.001.

[17] G. Guo, "Traffic forecasting via dilated temporal convolution with peak-sensitive loss," *IEEE Intell. Transp. Syst. Mag.*, early access, Nov. 17, 2021, doi: 10.1109/MITS.2021.3119869.

[18] G. Guo and T. Zhang, "A residual spatio-temporal architecture for travel demand forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102728, doi: 10.1016/j.trc.2020.102639.

[19] Y. Sun, K. Fu, Z. Wang, C. Zhang, and J. Ye, "Road network metric learning for estimated time of arrival," in *Proc. 25th IEEE Int. Conf. Pattern. Recognit.*, Milan, Italy, Jan. 2021, pp. 1820–1827.

[20] I. Jindal, X. Chen, M. Nokleby, and J. Ye, "A unified neural network approach for estimating travel time and distance for a taxi trip," 2017, *arXiv:1710.04350*.

[21] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 858–866.

[22] A. Derrow-Pinion *et al.*, "ETA prediction with graph neural networks in Google maps," 2021, *arXiv:2108.11482*.

[23] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "ConSTGAT: Contextual spatial-temporal graph attention network for travel time estimation at Baidu maps," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2697–2705.

[24] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Washington, DC, USA, Nov. 2009, pp. 352–361.

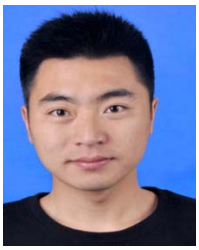
[25] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[26] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Trans. GIS*, vol. 24, no. 3, pp. 736–755, Jun. 2020, doi: 10.1111/tgis.12644.

[27] H. T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA:, Sep. 2016, pp. 7–10.

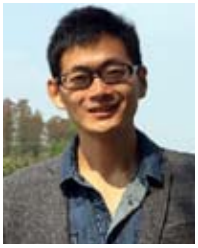
[28] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[30] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999, doi: 10.1613/jair.614.
[31] Y.-H. H. Tsai, S. Bai, M. Yamada, L.-P. Morency, and R. Salakhutdinov, "Transformer dissection: A unified understanding of transformer's attention via the lens of kernel," 2019, *arXiv:1908.11775*.
[32] A. Vaswani *et al.*, "Attention is all you need," 2017, *arXiv:1706.03762*.
[33] *GAIA Open Dataset*. Accessed: May 1, 2021. [Online]. Available: https://outreach.didichuxing.com/research/opendata/
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[35] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32. Vancouver, BS, Canada, Dec. 2019, pp. 8026–8037.
[36] Sohu. *The Average Number of Orders Per Second Per City on the Didi Chuxing Platform*. Accessed: Jan. 15, 2022. [Online]. Available: https://www.sohu.com/a/215372247_584224

Kaiqi Chen received the B.S. degree from Central South University (CSU), Changsha, China, in 2018, where he is currently pursuing the doctorate degree with the School of Geosciences and Info-Physics. His research interests include data mining and machine learning.



Guowei Chu received the B.S. degree from Central South University (CSU) in 2019, where he is currently a graduate student with the Spatiotemporal Data Mining Research Group. His research interest mainly includes spatio-temporal data mining of big geodata.

Xuexi Yang received the Ph.D. degree from Central South University (CSU), Changsha, China, in 2018. He was a Post-Doctoral Researcher in geographic information science with Peking University. He is currently a Lecturer with the School of Geosciences and Info-Physics, CSU. His research interests include spatio-temporal anomaly detection, spatio-temporal interaction pattern mining, and social sensing. He was supported by the National Natural Science Foundation of China.



Yan Shi received the Ph.D. degree from Central South University (CSU), Changsha, China, in 2015. He was a visiting Ph.D. student with Virginia Tech University (VT), Blacksburg, VA, USA. He is currently an Associate Professor with the School of Geosciences and Info-Physics and a Master's Supervisor with the School of Geosciences and Info-Physics, CSU. His research interests include spatio-temporal clustering, anomaly detection, and association rule mining. He was supported by the National Natural Science Foundation of China.

Kaiyuan Lei received the B.S. degree in 2021. She is currently pursuing the Ph.D. degree with the School of Geosciences and Info-Physics, Central South University (CSU), Changsha, China. Her research interest mainly includes uncertainty in spatio-temporal data mining.

Min Deng received the Ph.D. degrees from Wuhan University (WHU) in 2003 and the Asian Institute of Technology (AIT) in 2004. He is currently a Doctoral Supervisor and an Associate Dean of the School of Geosciences and Info-Physics, Central South University (CSU). He has hosted numerous major projects, including a key project of the National Natural Science Foundation of China. His current research interests include coordinated planning and spatio-temporal data mining, analysis, and modeling.