DSETA: Driving Style-Aware Estimated Time of Arrival

Bolin Zhang
Shenzhen University
College of Computer Science and Software Engineering
Shenzhen, Guangdong, China
zhangbolin2023@email.szu.edu.cn

Zhidan Liu*
The Hong Kong University of Science and Technology (Guangzhou)
Intelligent Transportation Thrust, System Hub
Guangzhou, Guangdong, China
zhidanliu@hkust-gz.edu.cn

Abstract

The accurate estimated time of arrival (ETA) is crucial for mobility and transportation applications. Although significant efforts have been made to improve ETA prediction, most existing approaches ignore the influence of individual driving habits and preferences, known as the *driving style*. Since different drivers may prefer specific routes and speeds based on their experience and familiarity with traffic conditions, driving styles play a crucial role in determining the actual ETA. To fill this gap, we present a novel approach, DSETA, which leverages deep learning to learn and then integrate driving style representations for personalized and precise ETA predictions. Our method employs a diffusion model that captures nuanced driving styles by generating driving speed distribution. We also utilize attention mechanisms to dynamically adjust the impacts of various spatio-temporal factors and driving styles on ETA predictions. Additionally, we introduce a Multi-View Multi-Task framework that incorporates auxiliary tasks, including segment-view driving style classification and route-view speed distribution prediction, to enhance the ETA learning process. A route-level speed prior regularization strategy further improves the model's generalization capabilities. Extensive experiments conducted on a large real-world trip trajectory dataset demonstrate that DSETA achieves high effectiveness and outperforms various baselines across multiple evaluation metrics.

CCS Concepts

• **Applied computing** → **Forecasting**; **Transportation**.

Keywords

Estimated Time of Arrival, Driving Style, Diffusion Model, Attention Mechanism, Multi-View Multi-Task

ACM Reference Format:
Bolin Zhang and Zhidan Liu. 2025. DSETA: Driving Style-Aware Estimated Time of Arrival. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761171

*Corresponding author

CIKM '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761171

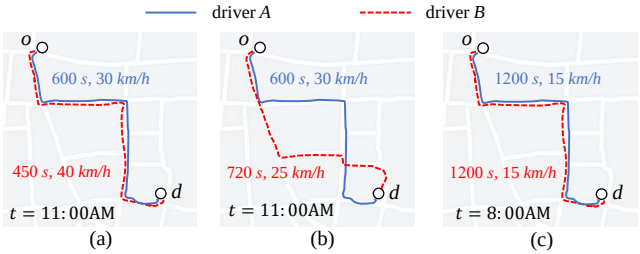


Figure 1: Variation in travel durations between the same origin o and destination d for driver *A* and driver *B* due to: (a) distinct driving styles; (b) familiarities on travel routes with the same travel distance; and (c) different departure times t. Annotations along the lines indicate total travel duration (in *seconds*) and driver's average driving speed (in *km/h*).

1 Introduction

Estimated time of arrival (ETA) is designed to forecast the travel duration from a specified origin to a destination along a planned travel route, taking into account a particular departure time. As a crucial component of location-based services, ETA is extensively utilized in a variety of contemporary mobility and transportation applications, including map navigation [3], instant delivery services [13], online ride-hailing platforms [25], and vehicle dispatch systems [24]. Accurate ETA predictions can greatly boost transportation efficiency, enhance user experiences, and promote environmental sustainability by reducing energy consumption.

Due to its essential functionality, ETA prediction has recently garnered significant attention from both industry [7, 9, 10, 22, 35] and academia [20, 21, 38]. For instance, Didi's WDR [35] employs a deep learning model that integrates spatio-temporal and route features for ETA, and has recently evolved into ProbTTE [22]. Similarly, Baidu Map's ConSTGAT [10] utilizes a graph attention mechanism to simultaneously model spatial and temporal information, later advancing to a meta-learning approach known as SSML [9]. Moreover, Google Maps [7] employs graph neural networks (GNNs) to capture spatial dependencies within road networks for ETA estimates. In academia, various ETA methods [15, 31, 32, 37] have been developed as well. In particular, the ADS-ETA framework [32] aims to address data sparsity in ETA predictions, while STHR [37] enhances ETA accuracy by leveraging spatio-temporal features across multiple dimensions. Despite significant research advancements in ETA prediction, most existing methods overlook the impact of

individual driving habits and preferences, referred to as *driving style* [29]. These driving styles are crucial in determining actual ETA, as different drivers may favor specific routes and speeds based on their experience and familiarity with traffic conditions. In reality, many drivers perceive the estimated travel times provided by map services as inaccurate. This discrepancy arises primarily because these services fail to account for variations in driving styles when predicting ETA. In fact, drivers can experience notably different travel times for identical trips, even with the same origin, destination, and departure time. For example, as illustrated in Figure 1(a), if driver B is an aggressive driver with a high average speed, B will complete the trip more quickly than driver A, who drives cautiously at a lower speed. Due to varying driving styles, their ETAs for the same journey should vary considerably.

Some notable studies [23, 31, 40] have considered driving styles in ETA predictions. These works, namely CoDriver [31] and MTSTAN [40], typically use travel speeds derived from historical trip data to provide personalized ETA predictions. However, a high average speed does not imply that a driver maintains that pace on every route. Using only speed does not capture the complexity of individual driving habits and preferences. Therefore, a more nuanced representation of driving styles is essential to enhance the accuracy of ETA predictions.

To address this gap, we thus propose a Driving Style-aware Estimated Time of Arrival (DSETA) approach. This method learns effective driving styles from historical trip data and utilizes these learned features to provide personalized and accurate ETA predictions. However, implementing DSETA presents several challenges that must be addressed.

First, *how to effectively represent driver's driving style for ETA predictions?* While using average speed is straightforward, it fails to capture the complexity and variability of a driver's style across different roads and traffic conditions. We contend that driving style should not be viewed as simply fast or slow, but as a distribution of speeds under varying circumstances. For example, Figure 1(b) further shows that driver B takes an alternative route with the same travel distance between the same origin and destination. However, this route takes longer due to a lower average speed on unfamiliar roads compared to the original route. Thus, it is crucial to conceptualize driving style as a speed distribution rather than a single value. To achieve this, we introduce the concept of *distance-duration pair* (DDP), which represents a trip by its travel distance and duration. We then propose a diffusion-based driving style representation method that learns from historical DDPs while simultaneously generating virtual DDPs. This approach allows the model to capture the implicit driving styles reflected in the distribution of a driver's speeds. More importantly, our driving style representation can be seamlessly integrated with existing ETA prediction methods, enhancing their overall performance.

Second, *how to dynamically adjust the impacts of driving style and various factors on ETA predictions?* In addition to driving style, external factors such as origin and destination locations, road types, and departure time significantly influence travel time. For example, even an aggressive driver cannot maintain high speeds on congested roads during peak traffic hours. As shown in Figure 1(c), both driver A and driver B are compelled to drive slowly during rush hour. Integrating driving style with these spatial-temporal factors and precisely adjusting their influences on ETA predictions pose a challenge. To address this challenge, we devise embedding techniques for spatial-temporal factors and employ attention mechanisms to model the relationships between driving style and these factors. The attention weights learned by the model allow for varying degrees of focus on each factor, thereby regulating their respective influences on the final ETA estimation.

Third, *how to efficiently train the model to achieve accurate ETA predictions?* A single ETA prediction task often lacks sufficient constraints to effectively capture the nuances of driving styles, resulting in representation deficiencies. Therefore, it is essential to design additional mechanisms to guide the model in leveraging driving styles for accurate ETA predictions. To tackle this issue, we propose a Multi-View Multi-Task (MVMT) learning framework that incorporates several well-designed auxiliary tasks to support the main ETA learning task. Specifically, we introduce a route-view speed distribution prediction task to learn the driver's speed distribution across various travel routes. Additionally, we implement a segment-view driving style classification task to capture the fine-grained effects of driving styles. Furthermore, we incorporate route-level speed prior regularization (PSPR) to bolster the model's generalization capabilities. Together, these auxiliary tasks and the ETA learning task fully leverage the potential of driving styles for more accurate ETA predictions.

In summary, we make the main contributions as follows:

- To our best knowledge, this is the first work to design the DDP generation task and employ a diffusion model to learn driving style representations. This representation enables deep learning models to effectively embed semantics related to practical driving styles.
- We introduce DSETA, a driving style-aware ETA approach that utilizes attention mechanisms to explore the relationships between spatio-temporal factors and driving styles.
- We propose a multi-view multi-task learning framework that incorporates several auxiliary tasks to guide driving style learning and enhance ETA predictions. Additionally, we devise the PSPR to improve the model's generalization capabilities.
- We evaluate DSETA using a large real-world trip trajectory dataset. Experimental results demonstrate that DSETA outperforms various baselines across all metrics, confirming the effectiveness of our approach.

The rest of this paper is organized as follows. Section 2 introduces the ETA prediction problem and the system overview of our DSETA. Later, we elaborate the design of diffusion-based driving style learning, transformer-based multi-factor fusion, and the MVMT framework for ETA learning in Sections 3, 4, and 5, respectively. Experimental results are reported in Section 6. We review the related works in Section 7. Finally, Section 8 concludes this paper.

2 Preliminary and Framework Overview

In this section, we present some preliminary definitions and the ETA prediction problem statement, and then introduce the system overview of our solution DSETA.

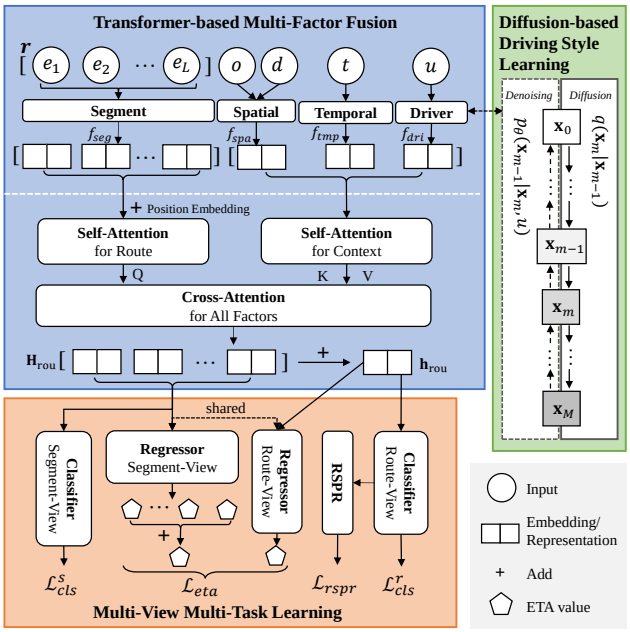


Figure 2: The architecture of DSETA.

2.1 Definitions and Problem Statement

Definition 1. ***(Road Network)*** *A road network is denoted by a directed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where each vertex in $\mathcal{V}$ represents a road intersection and each edge $e \in \mathcal{E}$ represents a road segment in the road network.*

Definition 2. ***(Travel Route)*** *A travel route $\mathbf{r}$ is defined as a sequence of connected road segments $\mathbf{r} = \langle e_1, e_2, \cdots, e_L \rangle$, where L is the number of road segments in the route.*

Definition 3. ***(Historical Trip)*** *A historical trip is defined as a 6-tuple $\mathbf{s} = \langle u, o, d, t, \mathbf{r}, y \rangle$, where u denotes the driver who accomplished the trip, o and d represent the trip origin and destination locations, respectively, t is the departure time, $\mathbf{r}$ denotes the travel route, and y is the travel time of this trip.*

Definition 4. ***(Distance-Duration Pair, DDP)*** *Given a trip $\mathbf{s}$, a DDP can be extracted and denoted by a 2-tuple $x = \langle \ell, y \rangle$, where ℓ and y represent the total travel distance and duration of this trip. Particularly, $\mathbf{x}$ denotes a set of DDPs.*

Definition 5. ***(Trip Query)*** *A trip query is defined as a 5-tuple $\mathbf{z} = \langle u, o, d, t, \mathbf{r} \rangle$, where u denotes the driver of the trip, o and d represent the trip origin and destination, respectively, t is the departure time, and $\mathbf{r}$ denotes the planned travel route.*

Problem 1. ***Driving Style-aware Estimated Time of Arrival Problem:*** *Given a trip query $\mathbf{z} = \langle u, o, d, t, \mathbf{r} \rangle$, we aim to predict the travel time $\hat{y}$ for driver u who drives from origin o to destination d along a given travel route $\mathbf{r}$ based on $\hat{y} = \mathsf{F}(\mathbf{z}, \mathcal{G})$, where $\mathcal{G}$ is the underlying road network and $\mathsf{F}(\cdot)$ is a mapping function, which perceives driver u's driving style and can be learned from historical trip dataset $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^{n}$.*

2.2 Overview of DSETA Solution

To address above problem, we propose a novel ETA learning approach – DSETA. As illustrated in Figure 2, DSETA comprises three key modules. First, the *Diffusion-based Driving Style Learning* module generates DDPs, enabling the extraction of implicit driving style representations for drivers (Section 3). Next, the *Transformer-based Multi-Factor Fusion* module integrates learned driving styles with spatial-temporal factors to enhance ETA predictions (Section 4). Finally, the *Multi-View Multi-Task (MVMT) Learning* module incorporates auxiliary tasks focusing on driving style from both route and segment perspectives, thereby supporting and improving the ETA learning task (Section 5).

The operational workflow of DSETA is outlined as follows.

- *Training phase:* DSETA learns driver style representations from dataset $\mathcal{S}$ with a diffusion model for DDP generation. A transformer encoder, guided by an MVMT module, is trained on embeddings of trip features and the style representations.
- *Inference phase:* The auxiliary classifiers in MVMT are inactive during inference. DDP generation is also unnecessary at this stage. Consequently, these components do not participate in the computation.

3 Diffusion-based Driving Style Learning

3.1 Motivation

Instead of a single average speed, we propose using speed distribution to better represent driving style. Analyzing drivers A and B from a real-world dataset, we show in Figure 3(a) that driver B, despite a higher average speed, has a broader speed distribution, indicating frequent speed adjustments due to changing conditions.

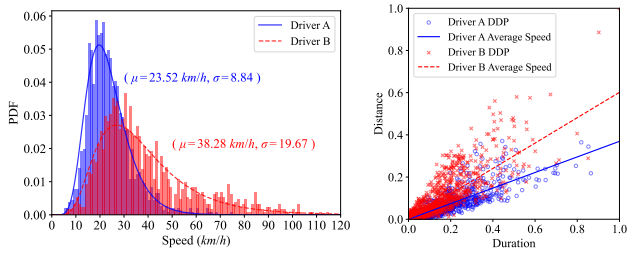
To analyze trip characteristics, we generate distance-duration pairs (DDPs) for drivers A and B, as visualized in Figure 3(b) with Min-Max normalization, while overlaying the average speeds. The DDP dispersion indicates that a single average speed is insufficient to represent driving style across varying distances and durations. Thus, DDPs are proposed as a better representation.

However, DDPs' discrete nature and the impracticality of modeling them with functions like Gaussian mixed models [30] for integration with deep learning-based ETA prediction frameworks [22, 35] lead us to propose a DDP generation task. This task synthesizes DDPs based on actual speed distributions, indirectly encoding driver information along with driving style semantics.

Drawing on the generative capabilities of diffusion models [39], we develop a DDP generation model using Denoising Diffusion Probabilistic Models (DDPM) [16]. The right part of Figure 2 shows the model's two Markov processes: the *forward diffusion process* and the *reverse denoising process*, detailed as follows.

3.2 Diffusion Process

The diffusion process, commonly referred to as *forward diffusion*, incrementally introduces noise into a driver's original DDP data, denoted as $\mathbf{x}_0$, over a series of time steps until the resulting corrupted DDPs conform to a predefined prior distribution, such as a Gaussian distribution.



(a) (b)

Figure 3: Statistics on the trip data for driver *A* (blue) and driver *B* (red): (a) PDF of driver's driving speeds (μ: average speed, σ: variance); (b) Visualization of DDPs, where each point is a normalized DDP, corresponding to one trip, and the lines represent drivers' average driving speeds.

Formally, the transformation occurring during the diffusion process, e.g., from x_0 to x_m, for one DDP can be expressed as follows:

$$q(x_m|x_0) = \mathcal{N}(x_m; \sqrt{\bar{\alpha}_m}x_0, (1-\bar{\alpha}_m)\mathbf{I}), \tag{1}$$

where x_m is the DDP derived at the m-th step, $\mathcal{N}(\cdot)$ is the Gaussian distribution, $\bar{\alpha}_m$ is the hyperparameters used for controlling the Gaussian noise level in the m-th step, and $\mathbf{I}$ is the identity matrix.

3.3 Denoising Process

The denoising process serves as the reverse of the forward diffusion, aiming to produce a clean DDP x_0 from noise sampled from the standard Gaussian distribution, represented as $x_M \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$. Specifically, the restored DDP x_0 belong to the speed distribution of a particular driver u, thereby reflecting the driving style of that individual. Formally, a single transition from the m-th step to the $(m-1)$-th step of the denoising process for one DDP can be expressed as follows:

$$p_\theta(x_{m-1}|x_m, u) = \mathcal{N}(x_{m-1}; \mu_\theta(x_m, m, u), \Sigma_\theta(x_m, m, u)), \tag{2}$$

where θ is the parameters of neural network model employed to instantiate the denoising process, and $\mu_\theta(\cdot)$ and $\Sigma_\theta(\cdot)$ are the mean and covariance matrix, respectively. Since the denoising process is also a Markov process, the clean DDP x_0 can be progressively restored from the noisy DDP x_M using the following approach:

$$p_\theta(x_0|x_M, u) = \prod_{i=1}^{M} p_\theta(x_{i-1}|x_i, u). \tag{3}$$

Consequently, the intended neural networks will incorporate driver priors and subsequently generate DDPs that reflect the driving characteristics of the driver.

3.4 DDP Denoiser

To integrate driver priors into denoising, we redesign the diffusion model's denoiser, creating our DDP denoiser capable of accepting driver information u as input. Since DDP only has two dimensions, i.e., travel distance and duration, we utilize a simple Multilayer Perceptron (MLP) as the backbone network for the DDP denoiser.

Given that both driver ID u and current time step m are numerical values, we employ one-hot encoding and a fully connected layer to obtain their embeddings, resulting in f_{dri} and f_m respectively. Specifically, f_{dri} is obtained as follows:

$$f_{dri} = \text{OneHot}(u)\mathbf{W}_u{}^T + \mathbf{b}_u, \tag{4}$$

where $\mathbf{W}_u$ and $\mathbf{b}_u$ denote the model's weights and bias. We perform similar operations as shown in Eq. (4) to obtain f_m. Additionally, since a noisy DDP in x_m is a two-dimensional vector as well, we use another fully connected layer to project each DDP in x_m into the same dimensional space as f_{dri} and f_m, resulting in f_{x_m}. Finally, the three vectors, i.e., f_{x_m}, f_{dri} and f_m, are combined to form the input of the DDP denoiser, while the output of the DDP denoiser is the predicted noise.

We employ the same training algorithm as the DDPM [16] to optimize our diffusion model. Upon completion of training, we obtain driver features imbued with driving style semantics. At this stage, we have two strategies for utilizing the learned driver features. One approach is to treat the learned embedding f_{dri} as the definitive representation of driving style, allowing direct application for ETA predictions. Alternatively, we can utilize model parameters $\mathbf{W}_u$ and $\mathbf{b}_u$ from Eq. (4) as pre-trained parameters, enabling them to be jointly trained alongside the downstream ETA prediction task. In our DSETA design, we choose the latter to dynamically update driving style features alongside downstream tasks.

4 Transformer-based Multi-Factor Fusion

In addition to driver driving style, several other factors influence the travel time of a trip, including the origin and destination, departure time, and the travel route.

4.1 Feature Embedding

To incorporate these factors and assess their impacts on the ETA prediction, DSETA represents them as segment feature f_{seg}, spatial feature f_{spa}, temporal feature f_{tmp}, and driving style feature f_{dri}. A route feature is represented as a feature matrix $\mathbf{F_r}$ by sequentially stacking the feature of each segment within $\mathbf{r}$ as $\mathbf{F_r} = [f_{seg_1}; f_{seg_2}; \cdots; f_{seg_L}]$. As a preprocessing step, various feature embedding techniques can be utilized. To effectively and reasonably consider the characteristics of various factors, we employ different embedding techniques to represent the factors. For example, graph structural features can be embedded using Node2Vec [14], and sparse features can be embedded using One-Hot encoding. Furthermore, DSETA adjusts the weights assigned to these factors in determining ETA using attention mechanisms, as follows.

4.2 Multi-Factor Fusion via Attention

Taking the derived segment embedding, spatial and temporal embeddings, and driving style representation as inputs, the Transformer-based encoder of DSETA aims to learn the relationships among these factors and dynamically adjust their influences on ETA predictions through attention mechanism. The cross-attention module requires three key components: the query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$. Accordingly, we organize the segment embedding as the query, and concatenate the spatial-temporal embeddings and driving style representation into a sequence to serve as the key and value, as follows:

$$\mathbf{Q} = \mathit{SelfAttention}(\mathbf{F_r}), \tag{5}$$

$$\mathbf{K} = \mathbf{V} = \mathit{SelfAttention}([f_{spa}; f_{tmp}; f_{dri}]). \tag{6}$$

In this formulation, Eq. (5) learns the relationship among different segments of a travel route, while Eq. (6) captures the relationship between spatial-temporal features and driving style. The advantage of organizing $[f_{spa}, f_{tmp}, f_{dri}]$ in this manner is that when a new contextual factor needs to be considered, its representation can be seamlessly concatenated into the existing sequence. Prior to concatenation, a fully connected layer is employed to project f_{spa}, f_{tmp}, and f_{dri} onto the same dimensional space as f_{e_i}.

With $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ as inputs, the Transformer leverages the cross-attention mechanism to generate representations as:

$$\mathbf{H}_{rou} = CrossAttention(\mathbf{Q}, \mathbf{K}, \mathbf{V}). \tag{7}$$

Specifically, we obtain $\mathbf{H}_{rou} = [\mathbf{h}_{e_1}, \mathbf{h}_{e_2}, \cdots, \mathbf{h}_{e_L}]$, where each vector $\mathbf{h}_{e_i}$ is fulfilled with the attention weight values that reflect the impacts of various influencing factors on the segment e_i within travel route $\mathbf{r}$. Unlike previous approaches that utilize the final vector $\mathbf{h}_{e_L}$ as the sole representation [22], we propose to aggregate all vectors in $\mathbf{H}_{rou}$ to form a comprehensive representation for the travel route as follows:

$$\mathbf{h}_{rou} = SumPooling(\mathbf{H}_{rou}). \tag{8}$$

This design offers a significant advantage: by employing the sequence of segments as the query $\mathbf{Q}$, we can derive segment-level representations that integrate multiple influencing factors into a single attention query. These segment-level representations facilitate the exploration of various segment combinations (i.e., different travel routes) beyond the currently inputted one, thereby providing more detailed information for downstream tasks.

5 Multi-View Multi-Task Learning

To achieve robust and accurate ETA predictions, we propose the Multi-View Multi-Task (MVMT) learning framework. This framework integrates auxiliary tasks for driving style classification to support the main task of ETA learning. The learning process is conducted at both the route view and the segment view, thereby enhancing the driving style-aware ETA predictions by taking into account the influences of global route characteristics as well as local segment details.

5.1 ETA Learning Tasks

The primary objective of this main task is to predict the travel time of a trip based on its origin, destination, departure time, and the planned travel route.

5.1.1 Route-View. The route-view ETA is modeled as a regression task aimed at predicting the travel time of a trip given its route representation $\mathbf{h}_{rou}$, as derived from Eq. (8). We employ an MLP as the regressor, i.e.,

$$\hat{y}^r = MLP_{eta}(\mathbf{h}_{rou}), \tag{9}$$

where $\hat{y}^r$ is the predicted ETA given $\mathbf{h}_{rou}$.

Additionally, we compute the mean average error (MAE) as the instantaneous training loss, defined as follows:

$$\mathcal{L}^r_{eta} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}^r_i|, \tag{10}$$

where n is the total number of travel routes in historical trip dataset $\mathcal{S} = \{\mathbf{s}_j\}_{j=1}^n$.

5.1.2 Segment-View. From the segment view, we calculate the travel time for each individual road segment and aggregate these times to derive the final ETA prediction for the entire travel route. The segment-level time is predicted as $\hat{y}^s_{j,e_i} = MLP_{eta}(\mathbf{h}_{e_i}), e_i \in \mathbf{r}_j$, where the MLP regressor shares parameters with the one used in the route-view. By summing the segment-level travel times, we obtain an estimation of travel time over the route $\mathbf{r}_j$ as $\hat{y}^{rs}_j = \sum_{i=1}^{L_j} \hat{y}^s_{j,e_i}$. Consequently, we can derive two training losses, both measured using MAE, from the segment-view ETA predictions. Specifically, Eq. (11) calculates the MAE loss associated with segment-level travel time, while Eq. (12) computes the MAE loss for route-level time.

$$\mathcal{L}^s_{eta} = \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{L_j} |y_{j,e_i} - \hat{y}^s_{j,e_i}|, \tag{11}$$

$$\mathcal{L}^{rs}_{eta} = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}^{rs}_j|, \tag{12}$$

where L_j represents the number of segments in the j-th travel route, and $N = \sum_{j=1}^{n}\sum_{i=1}^{L_j} 1$ denotes the total number of segments traveled by the trips in $\mathcal{S}$.

Therefore, the training loss associated with the main ETA learning task can be summarized as follows:

$$\mathcal{L}_{eta} = \mathcal{L}^r_{eta} + \mathcal{L}^s_{eta} + \mathcal{L}^{rs}_{eta}. \tag{13}$$

5.2 Driving Style Classification Tasks

5.2.1 Route-View. In addition to implicit representation of driving styles, travel speed serves as an intuitive indicator of a driver's driving preferences. To this end, we introduce an auxiliary task focused on route-view speed prediction that aims to learn a driver's speed preference for the planned travel route. Instead of directly predicting absolute speed value as a regression problem, we reformulate the task as a multi-class classification problem. Specifically, we discretize continuous speed values into distinct categories, with a maximum speed limit of 120 km/h and a speed interval denoted as Δ (e.g., 10 km/h). The boundary values b_j for all category intervals are calculated as $b_i = i \cdot \Delta, 0 \le i \le C$, where $C = \lceil 120/\Delta \rceil$ stands for the total number of categories.

Based on these settings, we construct a speed classifier using an MLP combined with Softmax. The classifier takes the route-view representation $\mathbf{h}_{rou}$, as derived from Eq. (8), as input and produces a probability vector $\hat{\mathbf{v}}$ representing the predicted speed distribution:

$$\hat{\mathbf{v}} = Softmax(MLP^r_{cls}(\mathbf{h}_{rou})). \tag{14}$$

To train the classifier, we minimize the classification loss $\mathcal{L}^r_{cls}$, defined as the cross entropy that quantifies the divergence between the predicted speed distribution and true labels:

$$\mathcal{L}^r_{cls} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i \cdot \log(\hat{\mathbf{v}}_i), \tag{15}$$

where $\mathbf{v}_i$ is the one-hot encoded label for the i-th route. Specifically, if v_i is the ground true average speed for the i-th route, the k-th element of $\mathbf{v}_i$ is set to 1, where $k = \lfloor v_i/\Delta \rfloor$.

Furthermore, we introduce Route-level Speed Prior Regularization (RSPR) to incorporate speed prior knowledge. Unlike the previous Route-Wise Prior Regularization [22], which needs trips on

the same route, RSPR needs trips by the same driver. It's easier to get trajectories from the same driver than the same travel route.

Figure 3(a) shows that a driver's speed distribution often follows a *log-normal* pattern, which we use to model the driver's speed prior. The expected average speed can be computed by equation $\hat{v} = e^{\mu + \frac{\sigma^2}{2}}$, where μ and σ^2 represent the expectation and variance of the log-normal distribution, respectively. To model the speed prior, we introduce a parameterized log-normal distribution. Let the random variable V denote the average speed of driver u on a specific route $\mathbf{r}$, which is assumed to follow the distribution $p(v)$:

$$V \sim p(v) = \frac{1}{v\sqrt{2\pi\sigma^2}} \exp(-\frac{(\log(v) - \mu)^2}{2\sigma^2}), \tag{16}$$

where μ and σ^2 represent the expectation and variance of $\log(V)$, respectively. We discretize the continuous distribution into a discrete form as $p_i = p(\frac{b_i + b_{i+1}}{2}) \cdot \Delta, 0 \leq i \leq C-1$, where $\frac{b_i + b_{i+1}}{2}$ stands for the speed cluster center. Then, we obtain the discrete variable $\mathbf{b} = [\frac{b_0+b_1}{2}, \frac{b_1+b_2}{2}, \cdots, \frac{b_{C-1}+b_C}{2}]$ and a probability vector $\mathbf{p} = [p_0, p_1, \cdots, p_{C-1}]$.

Subsequently, we leverage the cross-entropy between $\hat{\mathbf{v}}$ and $\mathbf{p}$ to derive the log-normal distribution that best approximates the model's output distribution. By setting $\nabla L_{prior} = 0$ to address the optimization problem defined in Eq. (17), we obtain the analytical solution represented in Eq. (18).

$$\min_{\mu,\sigma} \mathcal{L}_{prior} = -\hat{\mathbf{v}} \cdot \log(\mathbf{p}), \tag{17}$$

$$\mu = \hat{\mathbf{v}} \cdot \log(\mathbf{b}),\ \sigma^2 = \hat{\mathbf{v}} \cdot (\log(\mathbf{b}) - \mu \cdot \mathbf{1}_C)^2, \tag{18}$$

where $\mathbf{1}_C$ is a vector of ones with a length of C.

Once we have determined μ and σ^2 for the log-normal distribution, we compute the expected average speed as $\hat{v} = e^{\mu + \frac{\sigma^2}{2}}$. To facilitate the model's learning of the route-level speed prior, we introduce the loss function $\mathcal{L}_{rspr}$ as follows:

$$\mathcal{L}_{rspr} = \frac{1}{n} \sum_{i=1}^{n} |\hat{v}_i - v_i|, \tag{19}$$

where v_i is the ground true average speed for the i-th route.

5.2.2 Segment-View. In addition to assessing driving style at the segment level, we have designed an auxiliary task focused on segment-level driving style classification. This task can be analogized to sequence annotation task , such as text annotation [5] . In our context, the travel route is analogous to a sentence, while each segment represents a word in that sentence. Consequently, segment-level driving style classification involves annotating the driving style of a driver for the current segment.

Formally, we classify a driver's driving style based on the average speed for a segment using the following rules:

$$c = \begin{cases} 0, & \mu + \alpha < v, \\ 1, & \mu - \alpha < v < \mu + \alpha, \\ 2, & v < \mu - \alpha, \end{cases} \tag{20}$$

where v represents the average speed of a driver u on a specific road segment, μ is the average speed of all drivers on the same segment, and α is a threshold that defines the category boundaries. For our implementation, we set $\alpha = 0.1 \times \mu$ to maintain relative control, ensuring a balanced distribution of instances across the three classification categories.

Thus, we classify a driver u's driving style on a road segment into three distinct categories. Specifically, *category 0* indicates that driver u drives significantly faster than most other drivers; *category 1* signifies that driver u maintains a speed close to the average speed of all drivers on that segment; and *category 2* suggests that driver u drives more slowly compared to the average speed.

For this auxiliary task, we employ an MLP as the classifier to determine the driving style classification for drivers on segments, expressed as $\hat{\mathbf{c}} = Softmax(MLP^s_{cls}(h^u_{e_i}))$. Subsequently, we get the category as $\hat{c} = \arg\max(\hat{\mathbf{c}})$. We train this classifier by minimizing the cross-entropy loss $\mathcal{L}^s_{cls}$, defined as:

$$\mathcal{L}^s_{cls} = \frac{1}{N} \sum_{j=1}^{n} \sum_{i=1}^{L_j} \mathbf{c}_{j,e_i} \cdot \log(\hat{\mathbf{c}}_{j,e_i}), \tag{21}$$

where $\mathbf{c}_{j,e_i}$ represents the one-hot encoded label for the i-th segment in the j-th travel route.

5.3 Joint Optimization

Building upon the MVMT learning framework, we train the DSETA model by jointly optimizing the following objective:

$$\mathcal{L}_{overall} = \mathcal{L}_{eta} + \lambda_1 \mathcal{L}^r_{cls} + \lambda_2 \mathcal{L}_{rspr} + \lambda_3 \mathcal{L}^s_{cls}, \tag{22}$$

where $\mathcal{L}_{eta}$ represents the ETA regression loss as defined in Eq. (13). The terms $\mathcal{L}^r_{cls}$, $\mathcal{L}_{rspr}$, $\mathcal{L}^s_{cls}$ correspond to the auxiliary losses specified in Eq. (15), Eq. (19), and Eq. (21), respectively. Additionally, λ_1, λ_2, and λ_3 are hyperparameters that regulate the importance of different auxiliary components in the overall objective.

6 Performance Evaluation

6.1 Experimental Setup

Dataset. We conduct experiments using a large, real-world anonymous trip trajectory dataset collected in Shanghai city, China, in April 2015. This dataset comprises nearly 1.5 million trips generated by 1000 drivers. Each trip includes information of the origin, destination, departure time, and GPS records sampled during the trip at a low frequency of 0.1 Hz. We split the dataset into training, validation, and testing sets following a ratio of 21:2:7. Specifically, trips from the first three weeks are utilized for training, data from the subsequent two days are designated as the validation set, and trips from the last week serve as testing set. We extract the road network corresponding to the area covered by these trips from OSMnx [1] which models the road network as a graph $\mathcal{G}$. Using the

Table 1: Overall performance. The best results are marked in bold, and the second-best results are underlined.

Model	MAE (s)	RMSE (s)	MAPE (%)
Transformer	69.65±0.56	114.09±2.04	18.59±0.42
MURAT	69.05±0.44	116.71±1.00	18.20±0.15
WDR	67.92±0.27	113.65±0.49	<u>17.74</u>±0.21
ProbTTE	68.20±0.22	<u>113.10</u>±0.39	18.18±0.27
CoDriver	<u>67.83</u>±0.24	113.50±0.90	17.93±0.10
DSETA (ours)	**65.00**±0.57	**106.99**±1.68	**17.47**±0.37

Table 2: Performance comparison of different approaches across varying travel distances. The values in the last row indicate the average improvement of DSETA over the four baseline methods.

Model	*Short*			*Medium*			*Long*		
	MAE (*s*)	**RMSE** (*s*)	**MAPE** (%)	**MAE** (*s*)	**RMSE** (*s*)	**MAPE** (%)	MAE (*s*)	**RMSE** (*s*)	**MAPE** (%)
Transformer	47.53±0.40	69.47±1.08	21.45±0.68	81.58±0.38	113.76±1.37	14.03±0.26	145.95±3.00	217.31±5.52	12.77±0.29
MURAT	46.51±0.24	69.02±0.51	20.95±0.22	81.20±0.54	115.78±0.97	13.72±0.06	146.75±1.28	225.38±2.22	12.68±0.10
WDR	46.16±0.27	69.07±0.38	20.33±0.28	80.12±0.70	113.98±1.01	13.61±0.13	142.38±1.05	216.19±1.79	12.46±0.13
ProbTTE	46.71±0.07	69.20±0.41	20.97±0.38	80.52±0.31	113.74±0.32	13.80±0.16	141.33±0.88	214.32±1.56	12.40±0.10
CoDriver	46.12±0.22	68.60±0.49	20.64±0.10	79.87±0.23	113.44±0.81	13.63±0.15	142.24±0.55	216.69±2.07	12.40±0.11
DSETA (ours)	**45.11**±0.29	**67.09**±1.08	**20.30**±0.57	**77.56**±0.92	**109.81**±1.72	**13.12**±0.05	**131.13**±1.68	**198.89**±3.49	**11.52**±0.16
Avg. ↑	1.01 (2.19%)	1.51 (2.20%)	0.03 (0.15%)	2.31 (2.89%)	3.63 (3.20%)	0.49 (3.60%)	10.20 (7.22%)	15.43 (7.20%)	0.88 (7.10%)

Table 3: Performance comparison among DSETA variants.

Variant	**MAE** (*s*)	**RMSE** (*s*)	**MAPE** (%)
V_0 (*Transformer*)	69.65±0.56	114.09±2.04	18.59±0.42
V_1 (V_0 + A)	68.03±0.59	111.14±1.59	18.21±0.30
V_2 (V_1 + E)	67.48±1.03	112.39±2.01	17.35±0.06
V_3 (V_2 + D)	66.98±1.29	110.71±1.60	17.42±0.21
V_4 (V_3 + R)	65.79±0.87	108.45±1.88	17.72±0.32
V_5 (V_4 + P)	65.44±0.48	107.91±1.75	**17.26**±0.06
DSETA (V_5 + S)	**65.00**±0.57	**106.99**±1.68	17.47±0.37

graph $\mathcal{G}$, we employ an advanced map-matching algorithm, FMM [36], to accurately reconstruct the travel route for each trip based on its GPS records.

Baselines. We compare DSETA with several baseline approaches for ETA predictions, including *Transformer* [34], *MURAT* [20], *WDR* [35], *ProbTTE* [22], and *CoDriver* [31]. It is worth noting that the middle three methods are developed by Didi [6], the largest ride-hailing company in China, and have undergone online validation.

Performance metrics. We employ three widely used metrics, including Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), for model performance evaluation.

Environment settings. All experiments are conducted on a workstation server equipped with an i7-10700K CPU operating at a frequency of 3.80GHz, 32GB of RAM, and an RTX 3090 GPU with 24GB of memory.

6.2 Performance Comparison

Table 1 presents the overall performance comparison among various methods across three metrics. It is clear that DSETA outperforms all baseline methods across all metrics, achieving an average accuracy improvement of 2.83 seconds over the best-performing baseline in terms of MAE, 6.11 seconds for RMSE, and an enhancement of 0.27% in MAPE. Among the baseline methods, *Transformer* demonstrates the weakest performance, exhibiting significant gaps compared to other approaches, primarily because it does not account for the specific characteristics of the ETA problem. In contrast, *WDR* secures the best-second results in MAPE, while *ProbTTE* secures the best-second results in RMSE. Additionally, *CoDriver* achieves performance similar to *WDR*, owning to its same main architecture as *WDR*. Overall, DSETA consistently delivers the most superior results for ETA predictions, largely due to our incorporation of driver driving styles into the modeling process and the design of effective auxiliary tasks to support ETA learning.

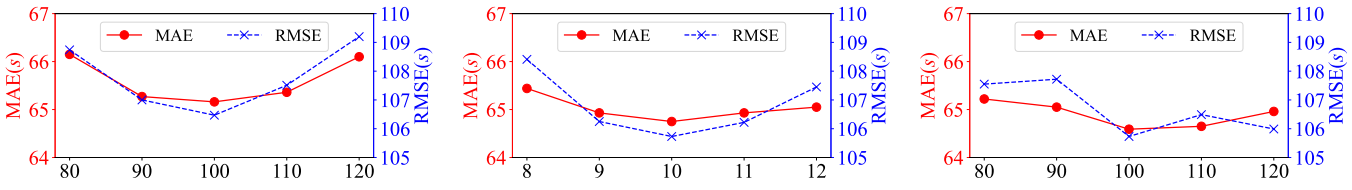
To gain a deeper understanding of the performance of various methods across different types of trips, we categorize the testing trips based on travel distance ℓ into three groups: *Short* ($\ell \leq 3\,km$), *Medium* ($3\,km < \ell \leq 6\,km$), and *Long* ($\ell > 6\,km$). Based on the experimental results presented in Table 2, we have the following key observations.

(1) DSETA achieves the best performance across all three trip categories and metrics. This indicates that DSETA can accurately predict travel time regardless of trip distance. The superiority of DSETA can be attributed to our ETA learning framework, which integrates both route-view and segment-view tasks to capture both global and local constraints.
(2) *CoDriver* and *WDR* demonstrates strong performance for both short and medium-distance trips. This success may be attributed to their utilization of LSTM as its underlying backbone, which effectively models dependencies in sequences of short and medium lengths.
(3) As travel distance increases, ProbTTE outperforms other baseline models due to the Transformer's exceptional ability to learn from long sequences. Notably, DSETA shows more pronounced improvements (i.e., greater than 7% enhancement) for long-distance trips relative to the short and medium trips. This suggests that the manifestations of driving style become more pronounced during longer trips.

6.3 Ablation Study

To evaluate the effectiveness of our design, we systematically incorporate components into the original Transformer architecture, which serves as the baseline variant V_0. This incremental approach generates various variants, ultimately culminating in the complete DSETA design. For clarity, we provide the abbreviations for each key component below.

- A: route representation via sum pooling (Add) in Eq. (8);
- E: segment-view ETA learning task;
- D: Driver embedding with driving style semantics;
- R: Route-view driving style classification task;
- P: route-level speed Prior regularization (RSPR);
- S: Segment-view driving style classification task.



(a) Effect of λ_1. (b) Effect of λ_2. (c) Effect of λ_3.

Figure 4: Impact of hyperparameter settings for λ_1, λ_2, and λ_3.

The results of ablation experiment are presented in Table 3.

Effect of improved route representation (+*A*): The design, adopting the sum of the vectors from each segment as the route representation (as described in Eq. (8)), yields significant reductions in prediction error across all three metrics, specifically a decrease of 1.62 *s* for MAE, 2.95 *s* for RMSE, and 0.38% for MAPE, as observed in Table 3 (comparing V_0 and V_1). This indicates the enhanced representation can effectively captures the nuances of a travel route.

Effect of segment-view ETA learning task (+*E*): A comparison between variants V_2 and V_1 reveals that the addition of segment-view ETA learning task results in a notable reduction of 0.86% in MAPE. This component effectively enhances the model's focus on local segments, leading to a significant decrease in MAPE which is particularly sensitive to routes with shorter travel distances.

Effect of enhanced driver embedding (+*D*): The distinction between V_2 and V_3 lies in the use of driver embeddings from the DDP task in the ETA task. Incorporating embeddings with explicit driving style semantics in V_3 improves MAE and RMSE metrics, indicating a positive contribution to ETA prediction performance.

Effect of route-view driving style classification task (+*R*): Updating V_3 to V_4 with component *R* results in further reductions in MAE and RMSE, though MAPE slightly increases. Since the route-view driving style classification is a discretized form of route-level speed prediction, favoring longer distances, the performance improvement is mainly due to long trips.

Effect of route-level speed prior regularization (+*P*): RSPR introduction further reduces all metrics, achieving the best MAPE and second-best MAE and RMSE results. This indicates RSPR improves performance for both long and short trips in the route-view driving style classification task.

Effect of segment-view driving style classification task (+*S*): Incorporation of *S* completes DSETA 's design, showing best MAE and RMSE results, and reasonable MAPE performance. Operating at the segment level, it enables driving style comparisons among drivers, potentially improving the model's ability to weigh driving style influence on ETA predictions.

6.4 Impact of Hyperparameters

In this section, we sequentially determine the optimal values for λ_1, λ_2, and λ_3. Once a hyperparameter is established, we fix its value and proceed to identify the next one in the sequence. All experiments conducted in this section utilize the same random seed to ensure consistency in results.

We first evaluate the effect of λ_1 (varied from 80 to 120), finding that DSETA performs optimally at $\lambda_1 = 100$ (Figure 4(a)). Next, we assess λ_2 in the range of 8 to 12; as shown in Figure 4(b), both MAE and RMSE initially decrease and then rise, peaking at $\lambda_2 = 10$. Finally, with λ_1 and λ_2 fixed, we vary λ_3 from 80 to 120 and observe optimal performance at $\lambda_3 = 100$ (Figure 4(c)).

Table 4: Accuracy of auxiliary classification tasks.

Task	All (%)	Short (%)	Medium (%)	Long (%)
Route-view	51.09±3.43	46.71±4.37	59.02±2.87	58.82±0.60
Segment-view	47.22±0.48	47.02±0.55	46.05±0.53	48.35±0.40

6.5 Performance of Auxiliary Tasks

We have designed two auxiliary driving style classification tasks, on both route view and segment view, to support the ETA learning task. We conduct experiments to evaluate their classification accuracies, with results presented in Table 4. In the route-view classification task, which includes 12 speed categories, our approach achieves a high accuracy of 51.09%, significantly exceeding the random guessing accuracy of 8.33% (i.e., $\frac{1}{12}$). When comparing accuracies across different trip sets with varying distances, we find that this task performs notably better for medium and long trips.

In the segment-view classification task, which includes 3 driving style categories, we find an overall accuracy of 47.22%, demonstrating a relatively stable performance across the three trip sets. This may be attributed to the task's focus on local segments, independent of the travel distance of a trip.

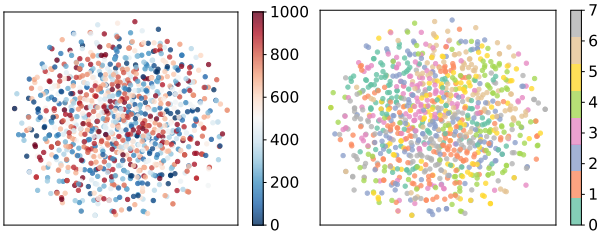
6.6 Effectiveness of Driving Style Embedding

To evaluate the versatility of our diffusion-based driving style learning approach, we integrate the derived driver embedding f_{dri} with other methods. As illustrated in Table 5, the models enhanced with our driver embedding show improved ETA prediction performance across all metrics compared to the original versions. These results suggest that our driving style representation, learned through the DDP generation task, can be effectively integrated with other ETA prediction methods, enhancing their performance by incorporating nuanced driving style semantics.

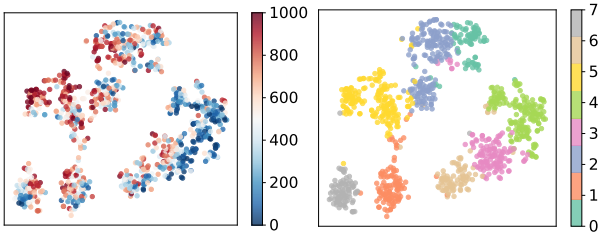
To further clarify the observed improvements, we visually compare the driving embeddings generated by WDR and our method using two visualization schemes. First, we rank 1000 drivers by average speed and apply a blue-to-red gradient. Second, we cluster the embeddings into eight groups using k-means [28], each assigned a unique color, and visualize them with t-SNE [33].

Table 5: Performance comparison between the original model and the one enhanced with our driver embedding f_{dri}.

Model	MAE (s)	RMSE (s)	MAPE (%)
WDR	67.92±0.27	113.65±0.49	17.74±0.21
WDR w/ f_{dri}	67.42±0.46	113.05±1.81	17.51±0.06
Improvement	↓ 0.50	↓ 0.60	↓ 0.23%
ProbTTE	68.20±0.22	113.10±0.39	18.18±0.27
ProbTTE w/ f_{dri}	67.80±0.71	112.26±2.13	17.92±0.27
Improvement	↓ 0.40	↓ 0.84	↓ 0.26%



(a) ***WDR***



(b) DSETA

Figure 5: t-SNE visualization of driver embeddings generated by *WDR* and DSETA using two coloring schemes. In the first scheme, drivers are ranked by their average speeds. In the second scheme, drivers are clustered using their embeddings.

As shown in Figure 5(a), WDR's embeddings exhibit intermingled blue and red dots with indistinct category boundaries. In contrast, Figure 5(b) shows our method produces a gradual blue-to-red transition and well-separated clusters. This demonstrates that our approach effectively captures speed-related driving style semantics, enabling clear driver categorization and labeling.

7 Related Works

7.1 Estimated Time of Arrival

Estimated Time of Arrival (ETA) is a critical component of mobility and transportation services. Accurate ETA predictions can effectively enhance the operational efficiency of transportation systems [12, 20] and improve user travel experience [4, 15]. Considerable efforts have been made to achieve more precise travel time estimates. As the capabilities of deep learning models continues to evolve, recent approaches to addressing the ETA problem have predominantly leveraged deep learning architectures. Approaches are categorized by route information in trip queries:

Without specified routes. Yuan et al. utilize historical trajectory representations as an auxiliary task [38]. Lin et al. adopt a diffusion model to generate pixel-level trajectories as route information [21]. Liu et al. use adversarial inverse reinforcement learning for personalized route inference [23]. Due to the absence of route information in this category of tasks, these methods are employed to generate or introduce trajectory information in various manners.

With specified routes (our focus). Wang et al. treat ETA as regression, using RNNs for route representation [35]. Liu et al. advance this with probabilistic prediction and a Transformer architecture [22]. Other research furtehr considers spatio-temporal factors [17], route context [10], and data sparsity [32]. Sun et al. explore driving style using average speed [31], but this is too simplistic for capturing the complexities of driving behavior. Zou et al. simultaneously predict traffic speed and travel time, taking into account individual preferences reflected in ETC data [40].

7.2 Driving Style

Driving style, often referred to as driving behavior and driving pattern in various contexts [29], has been extensively studied due to its significant implications in practical applications such as vehicle insurance [11, 19], safe driving [2, 18], and other domains. Numerous methods have been developed to learn driving style representations from trajectory data. For examply, Dong et al. employ an autoencoder framework to directly learn driving style from GPS data [8], while Liu et al. utilize an adversarial generative network to indirectly learn driving style [26, 27].

However, these methods may be overly complex and lack specificity for the context where ETA prediction is the primary task. In this regard, Sun et al. use average speed as a measure of driving style to learn driver embeddings [31], demonstrating the advantages of driving-style-aware ETA prediction. In this paper, we propose, for the first time, a diffusion-model based DDP generation task to implicitly learn a driver's driving style and leverage this knowledge to enhance ETA predictions.

8 Conclusion

This paper presents DSETA, a novel approach designed to effectively learn driver driving styles for accurate and personalized ETA predictions. We propose, for the first time, the use of a diffusion model to implicitly derive driving style representations. Additionally, we employ attention mechanisms to dynamically assess the impacts of driving style and various spatio-temporal factors on travel time estimates. Furthermore, we develop a multi-view multi-task leaning framework to enhance ETA learning, enabling the model to learn simultaneously from both segment and route perspectives through well-designed auxiliary driving style learning tasks. Extensive experiments on a large real-world trip dataset demonstrate the superiority of our approach. Notably, the driving styles derived from our method can be seamlessly integrated into other ETA prediction models, greatly improving their performance.

Acknowledgments

This work was supported in part by China NSFC under Grant 62172284.

GenAI Usage Disclosure

In the preparation of this manuscript, we have utilized the GenAI tools (i.e., ChatGPT) exclusively for the purpose of grammar checking and refinement of the English language in our writing. We did not employ any GenAI tools for the generation of codes or data at any stage of our research. All content presented in this paper is the result of our original work and analysis.

We affirm that our use of ChatGPT was limited to enhancing the clarity and readability of our manuscript, ensuring compliance with the ACM's Authorship Policy regarding the use of GenAI.

References

[1] Geoff Boeing. 2024. Modeling and Analyzing Urban Networks and Amenities with OSMnx. (2024).

[2] German Castignani, Thierry Derrmann, Raphael Frank, and Thomas Engel. 2015. Driver behavior profiling using smartphones: a low-cost platform for driver monitoring. *IEEE Intelligent Transportation Systems Magazine* 7, 1 (2015), 91–102.

[3] Di Chen, Ye Yuan, Wenjin Du, Yurong Cheng, and Guoren Wang. 2021. Online route planning over time-dependent road networks. In *IEEE ICDE*.

[4] Zebin Chen, Xiaolin Xiao, Yue-Jiao Gong, Jun Fang, Nan Ma, Hua Chai, and Zhiguang Cao. 2022. Interpreting trajectories from multiple views: a hierarchical self-attention network for estimating the time of arrival. In *ACM SIGKDD*.

[5] Alebachew Chiche and Betselot Yitagesu. 2022. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data* 9, 1 (2022), 10.

[6] Didi Chuxing. 2025. https://www.didiglobal.com/.

[7] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, Peter W. Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Velickovic. 2021. ETA prediction with graph neural networks in Google Maps. In *ACM CIKM*.

[8] Weishan Dong, Ting Yuan, Kai Yang, Changsheng Li, and Shilei Zhang. 2017. Autoencoder regularized network for driving style representation learning. In *IJCAI*.

[9] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. 2021. SSML: self-supervised meta-learner for en route travel time Estimation at Baidu Maps. In *ACM SIGKDD*.

[10] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. 2020. ConSTGAT: contextual spatial-temporal graph attention network for travel time estimation at Baidu Maps. In *ACM SIGKDD*.

[11] Zhihan Fang, Guang Yang, Dian Zhang, Xiaoyang Xie, Guang Wang, Yu Yang, Fan Zhang, and Desheng Zhang. 2021. MoCha: large-scale driving pattern characterization for usage-based insurance. In *ACM SIGKDD*.

[12] Kun Fu, Fanlin Meng, Jieping Ye, and Zheng Wang. 2020. CompactETA: a fast inference system for travel time prediction. In *ACM SIGKDD*.

[13] Chengliang Gao, Fan Zhang, Guanqun Wu, Qiwan Hu, Qiang Ru, Jinghua Hao, Renqing He, and Zhizhao Sun. 2021. A deep learning method for route and time prediction in food delivery service. In *ACM SIGKDD*.

[14] Aditya Grover and Jure Leskovec. 2016. node2vec: scalable feature learning for networks. In *ACM SIGKDD*.

[15] Jindong Han, Hao Liu, Shui Liu, Xi Chen, Naiqiang Tan, Hua Chai, and Hui Xiong. 2023. iETA: a robust and scalable incremental learning framework for time-of-arrival estimation. In *ACM SIGKDD*.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

[17] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. 2020. HetETA: heterogeneous information network embedding for estimating time of arrival. In *ACM SIGKDD*.

[18] Derick A. Johnson and Mohan M. Trivedi. 2011. Driving style recognition using a smartphone as a sensor platform. In *IEEE ITSC*.

[19] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S. Jensen. 2018. Distinguishing trajectories from different drivers using incompletely labeled trajectories. In *ACM CIKM*.

[20] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *ACM SIGKDD*.

[21] Yan Lin, Huaiyu Wan, Jilin Hu, Shengnan Guo, Bin Yang, Youfang Lin, and Christian S Jensen. 2023. Origin-destination travel time oracle for map-based services. *Proceedings of the ACM on Management of Data* 1, 3 (2023), 1–27.

[22] Hao Liu, Wenzhao Jiang, Shui Liu, and Xi Chen. 2023. Uncertainty-aware probabilistic travel time prediction for on-demand ride-hailing at DiDi. In *ACM SIGKDD*.

[23] Shan Liu, Ya Zhang, Zhengli Wang, Xiang Liu, and Hai Yang. 2025. Personalized origin–destination travel time estimation with active adversarial inverse reinforcement learning and Transformer. *Transportation Research Part E: Logistics and Transportation Review* 193 (2025), 103839.

[24] Zhidan Liu, Jiangzhou Li, and Kaishun Wu. 2022. Context-aware taxi dispatching at city-scale using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems* 23, 3 (2022), 1996–2009.

[25] Zhidan Liu, Hongquan Zhang, Guofeng Ouyang, Junyang Chen, and Kaishun Wu. 2024. Data-driven pick-up location recommendation for ride-hailing services. *IEEE Transactions on Mobile Computing* 23, 2 (2024), 1001–1015.

[26] Zhidan Liu, Junhong Zheng, Zengyang Gong, Haodi Zhang, and Kaishun Wu. 2021. Exploiting multi-source data for adversarial driving style representation learning. In *Database Systems for Advanced Applications*.

[27] Zhidan Liu, Junhong Zheng, Jinye Lin, Liang Wang, and Kaishun Wu. 2023. Radar: adversarial Driving Style Representation Learning With Data Augmentation. *IEEE Transactions on Mobile Computing* 22, 12 (2023), 7070–7085.

[28] J MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.

[29] Clara Marina Martinez, Mira Heucke, Fei-Yue Wang, Bo Gao, and Dongpu Cao. 2018. Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey. *IEEE Transactions on Intelligent Transportation Systems* 19, 3 (2018), 666–676.

[30] Douglas A Reynolds et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics* 741, 659-663 (2009).

[31] Yiwen Sun, Kun Fu, Zheng Wang, Donghua Zhou, Kailun Wu, Jieping Ye, and Changshui Zhang. 2022. CoDriver ETA: combine driver information in estimated time of arrival by driving style learning auxiliary task. *IEEE Transactions on Intelligent Transportation Systems* 23, 5 (2022), 4037–4048.

[32] Yiwen Sun, Wenzheng Hu, Donghua Zhou, Baichuan Mo, Kun Fu, Zhengping Che, Zheng Wang, Shenhao Wang, Jinhua Zhao, Jieping Ye, Jian Tang, and Changshui Zhang. 2022. Alleviating data sparsity problems in estimated time of arrival via auxiliary metric learning. *IEEE Transactions on Intelligent Transportation Systems* 23, 12 (2022), 23231–23243.

[33] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

[35] Zheng Wang, Kun Fu, and Jieping Ye. 2018. Learning to estimate the travel time. In *ACM SIGKDD*.

[36] Can Yang and Győző Gidófalvi. 2018. Fast map matching, an algorithm integrating hidden Markov model with precomputation. *International Journal of Geographical Information Science* 32, 3 (2018), 547–570.

[37] Haitao Yuan, Guoliang Li, and Zhifeng Bao. 2022. Route travel time estimation on a road network revisited: heterogeneity, proximity, periodicity and dynamicity. *Proceedings of the VLDB Endowment* 16, 3 (2022), 393–405.

[38] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. 2020. Effective travel time estimation: when historical trajectories over road networks matter. In *ACM SIGMOD*.

[39] Yuan Yuan, Jingtao Ding, Chenyang Shao, Depeng Jin, and Yong Li. 2023. Spatio-temporal diffusion point processes. In *ACM SIGKDD*.

[40] Guojian Zou, Ziliang Lai, Changxi Ma, Meiting Tu, Jing Fan, and Ye Li. 2023. When Will We Arrive? A Novel Multi-Task Spatio-Temporal Attention Network Based on Individual Preference for Estimating Travel Time. *IEEE Transactions on Intelligent Transportation Systems* 24, 10 (2023), 11438–11452.