SHAPE: Spatial Hyperbolic-Embedding Augmented Pretraining for Estimated Time of Arrival on Obfuscated Trajectories

Ruopu Li[1], Yong Chu[1], Kai Yang[1], James Jianqiao Yu[1], Lixia Wu[2], Xun Zhou[1,*]
[1]*School of Computer Science and Technology*, *Harbin Institute of Technology, Shenzhen*, Shenzhen, China
24s151034@stu.hit.edu.cn, chuyong@stu.hit.edu.cn, kaiyang.cs@outlook.com, jqyu@ieee.org, zhouxun2023@hit.edu.cn
[2]*ByteDance Inc.*, Hangzhou, China
chentianya.123@bytedance.com

***Abstract*—Estimated Time of Arrival (ETA) prediction is a cornerstone of Intelligent Transportation Systems (ITS). Existing ETA models have achieved strong performance by leveraging raw GPS coordinates, map matching and road-network attributes. However, due to privacy, commercial, and cross-platform constraints, downstream ETA services often cannot access raw geographic trajectories in practical scenarios. Instead, they are provided only with obfuscated or transformed coordinate sequences, where the direct map alignment becomes infeasible. As a result, map-dependent ETA models are difficult to directly apply under obfuscated or transformed coordinate settings. To address this problem, we propose SHAPE, a Spatial Hyperbolic-Augmented Pretraining framework for ETA prediction on obfuscated trajectories. We first design a physics-informed differential feature decomposition to capture relative geometric and sequential movement patterns while reducing dependence on absolute coordinates. We further introduce a multi-task self-supervised pre-training framework with hybrid Euclidean-hyperbolic contrastive learning, where the Euclidean and hyperbolic components provide complementary inductive biases for modeling local continuity and latent multi-scale route structures. During fine-tuning, dynamic context information is incorporated through gated cross-attention for ETA prediction. Extensive experiments on controlled transformed-coordinate benchmarks and a real-world obfuscated logistics benchmark demonstrate that SHAPE achieves the best overall performance against representative ETA baselines.**

***Index Terms*—Estimated Time of Arrival, Trajectory Representation Learning, Hyperbolic Geometry, Contrastive Learning**

I. Introduction

Estimated Time of Arrival (ETA) prediction is a fundamental problem in intelligent transportation systems [1]–[8]. Accurate ETA predictions not only directly enhance user experience but also optimize system resource allocation and reduce overall operational costs. Existing ETA paradigms are typically categorized into origin-destination (OD)-based methods [9]–[11] and ROUTE-based methods [12]–[14]. OD-based methods offer rapid responses but ignore intermediate paths, thereby limiting their accuracy. In contrast, ROUTE-based methods demonstrate high accuracy by modeling fine-grained trajectory sequences. However, their success often hinges on the assumption that high-precision GPS trajectories are directly accessible and aligned with the road network.

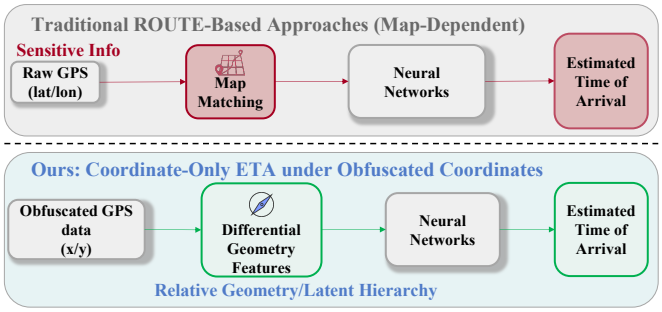

Fig. 1. Paradigm shift from map-dependent route modeling to coordinate-only geometric representation under obfuscated spatial reference systems.

However, this assumption may not hold in practical logistics and mobility scenarios. Due to privacy regulations such as GDPR [15], commercial confidentiality [16]–[18], and cross-organization data sharing constraints, trajectory data may be processed before being exposed to downstream models. As a result, ETA models may only observe coordinate sequences represented in obfuscated spatial reference systems. Since the underlying processing procedure is usually unavailable, these coordinates cannot be reliably aligned with public maps, and road-segment identities or topological relations cannot be directly recovered. Therefore, map-dependent methods based on map matching have limited applicability in this setting.

Despite the absence of explicit map information, obfuscated trajectories may still preserve useful geometric and sequential cues. Specifically, patterns such as successive high-curvature turns and extended low-curvature straight segments may implicitly reflect complex intersections and arterial road sections, respectively. Meanwhile, urban routes are not arbitrary point sequences; they are shaped by multi-level road systems, including local streets, arterial roads, and expressways. When only obfuscated coordinate sequences are available, these map-derived cues are inaccessible and the route hierarchy must be inferred from trajectories themselves. Although Euclidean representations remain effective for local geometric modeling,

* Corresponding author.

their flat geometry may provide limited inductive bias for branching and hierarchical structures [19]–[21]. This motivates us to explore a hybrid representation space that preserves Euclidean modeling capacity while introducing a non-Euclidean bias for latent hierarchical route organization. Meanwhile, ETA labels supervise outcomes rather than route structure, motivating self-supervised pre-training.

These observations raise three key questions for ETA prediction under restricted spatial information:

(Q1) Can accurate ETA prediction be achieved using only spatially obfuscated trajectory data?

(Q2) How can motion patterns be characterized from geometric and sequential trajectory structures?

(Q3) Can latent route hierarchy be modeled without explicit road-network priors?

To address these questions, we propose **SHAPE**[1], a Spatial Hyperbolic Embedding Augmented Pre-training model for ETA on Obfuscated Trajectories. SHAPE first constructs a **physics-informed** differential geometric feature representation from coordinate-only route sequences to capture rich motion semantics, and further conducts self-supervised pre-training through masked autoencoding, next-point prediction, and Euclidean–Hyperbolic hybrid-space contrastive learning to jointly model local geometry, sequential dynamics, and latent route hierarchy. Finally, the learned route representation is fused with departure context via gated cross-attention for ETA prediction. Overall, we make the following contributions:

- **Differential route representation:** We design a map-agnostic differential feature scheme that represents obfuscated trajectories via relative geometric and motion patterns, rather than absolute locations or map-aligned road segments.
- **Hybrid Space Contrastive Learning:** We design a hybrid Euclidean-Hyperbolic contrastive learning mechanism to implicitly induce the latent hierarchical structures of urban routes. By synergizing the complementary properties of both geometries, SHAPE captures both local kinematic variations and latent hierarchical route organization, leading to more discriminative route representations.
- **Comprehensive evaluation:** We conduct extensive experiments on multiple real-world trajectory datasets under controlled coordinate transformations and industrial obfuscated-coordinate settings. The results demonstrate its effectiveness in practical map-unavailable ETA scenarios.

II. Related Work

Estimated Time of Arrival (ETA) Prediction. ETA prediction, also known as TTE (Travel Time Estimation), has evolved over decades, transitioning from early model-driven approaches based on queuing theory [22] or cell transmission models [23] to data-driven deep learning methodologies. Existing mainstream methods are categorized into two types: OD-based and ROUTE-based. OD-based methods focus on the statistical correlation between origin and destination. For instance, MURAT [9] combines spatio-temporal priors via multi-task representation learning; TEMP [10] aggregates weighted averages of neighboring trips; and DeepOD [24] reconstructs hidden representations of historical trajectories to compensate for missing intermediate data. While computationally efficient, these methods struggle to capture microscopic traffic variations. Conversely, ROUTE-based methods model the complete path sequence. DeepTTE [12] introduces geo-convolution and LSTMs to learn spatio-temporal dependencies; WDR [13] jointly trains Wide-Deep-Recurrent networks; HetETA [25] further exploit heterogeneous road-network information; MetaTTE [26] leverages meta-learning for generalization; HierETA [27] models segment-level structures via multi-view representation; and MulT-TTE [28] incorporates multi-modal sequences with self-supervised learning. Despite their higher accuracy, most of these methods rely on standard coordinates. In contrast, our work focuses on ETA prediction under a more restrictive setting, where trajectories are spatially obfuscated and explicit map-aligned resources are inaccessible.

Trajectory Representation Learning. Trajectory representation learning aims to encode variable-length sequences into dense vectors to support downstream tasks such as similarity computation, ETA and so on. Existing works fall into GPS-based and segment-based approaches. GPS-based methods embed raw points directly; for example, traj2vec [29] uses RNNs to capture spatio-temporal dependencies, while T3S [30] combines self-attention for long-range correlations. Segment-based methods convert GPS trajectories into road segments via map-matching. GNN-based methods such as Toast [31] incorporate road-network topology to capture richer spatial semantics. Recent universal trajectory pre-training methods, such as JGRM [32] and UVTM [33], learn transferable representations from large-scale mobility data. Some of these methods include ETA prediction as a downstream task, demonstrating the usefulness of general trajectory representations. LLM-based trajectory or path representation methods, such as TrajCogn [34] and Path-LLM [35], further introduce semantic or textual knowledge to enhance mobility understanding. In contrast, SHAPE learns directly from obfuscated coordinates without map-aligned segments or external semantics.

Trajectory Privacy and Restricted Spatial Access. Privacy protection in trajectory mining has become a hotspot, given that individuals can be re-identified from anonymous data using only a few spatio-temporal points [36]. To mitigate these risks, extensive research has investigated privacy-preserving techniques for trajectory data. Traditional publishing-oriented methods like de-identification or K-anonymity are used to reduce risks, but they often struggle to defend against linkage attacks [37]. Differential privacy mechanisms inject calibrated noise into locations or trajectories, but excessive perturbation may distort geometric structures and harm downstream utility [38], [39]. Cryptographic protocols like Homomorphic Encryption [40] allow collaborative mining without leaking local details but incur high computational costs. More recently, CRATE [41] generates dummy routes to mask real user paths. These studies focus primarily on designing privacy mechanisms or protected data-sharing protocols. Unlike these

[1]The code is available at https://github.com/spring4869/SHAPE.

studies, our method focuses on ETA for geometrically obfuscated trajectories. Under this setting, our goal is to recover useful geometric, sequential, and latent hierarchical signals for accurate travel time prediction.

III. Preliminaries

In this section, we formalize the relevant concepts and the problem addressed in this paper.

Definition 1 (Coordinate-Obfuscated Route): A coordinate-obfuscated route is an ordered sequence of spatial points represented in an obfuscated or transformed coordinate system, denoted as $R = \langle p_1, p_2, \ldots, p_L \rangle$. Here, $p_i = (x_i, y_i)$ is a coordinate point and L denotes the length of the trajectory. Unlike standard GPS coordinates, the points in R cannot be reliably aligned with public maps or road segments by the downstream ETA model.

Definition 2 (Historical Trip): Let U be a user. A historical trip is defined as a tuple $P = (R, T_{\text{start}}, T_{\text{end}})$, which represents the observed coordinate-obfuscated route R traversed by user U during the time interval from T_{start} to T_{end}. The ground-truth travel time is defined as $Y = T_{\text{end}} - T_{\text{start}}$.

Definition 3 (Estimated Time of Arrival): Given a coordinate-obfuscated route R, a departure timestamp t_{dep}, and optional contextual information such as a user identifier u_{id}, the objective is to construct a predictive model $\mathcal{F}$ that outputs the estimated travel time $\hat{Y}$. The formal definition is as follows:

$$\hat{Y} = \mathcal{F}_\theta(R, t_{\text{dep}}, u_{\text{id}}), \tag{1}$$

where θ represents the learnable parameters. The corresponding estimated arrival timestamp can be obtained as $t_{\text{dep}} + \hat{Y}$. During training, $\mathcal{F}_\theta$ is optimized on historical trips by fitting the ground-truth travel time Y.

Constraints and Assumptions. We study ETA prediction under restricted spatial information. The downstream model only observes coordinate-obfuscated routes and departure context, without access to raw GPS coordinates. The underlying coordinate processing procedure is unknown or unavailable to the model, so the observed coordinates cannot be reliably mapped back to the original geographic space. SHAPE does not assume lossless obfuscation or propose a new privacy mechanism; instead, it predicts ETA directly from transformed or obfuscated coordinate sequences.

If the processing procedure severely destroys local geometry or temporal consistency, the remaining trajectory signals may become less informative, leading to degraded predictive utility.

IV. Methodologies

In this section, we describe the architecture of SHAPE. The proposed framework consists of two distinct phases: *(i) Self-Supervised Pre-training Phase,* where we adopt a multi-task learning paradigm to jointly train the route encoder, incorporating three pretext objectives: hybrid space contrastive learning, masked autoencoder, and next point prediction. These objectives are designed to jointly capture complementary structural properties of routes; *(ii) Fine-tuning Phase,* where the route representations extracted from the route encoder are fused with external dynamic contexts. The resulting fused representations are then fed into a lightweight regression head to predict the travel time.

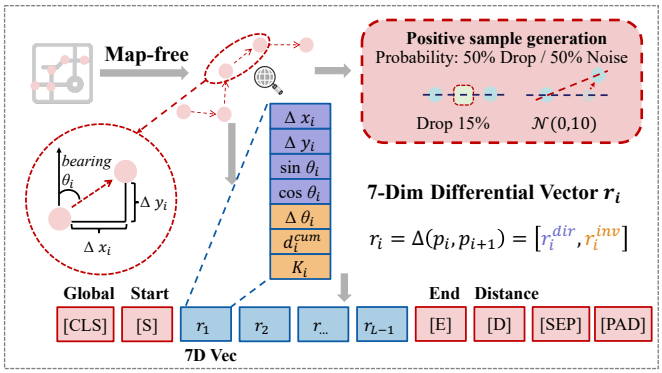

Fig. 2. Differential feature decomposition and Route Structure.

A. Differential Features and Route Structure

To decouple the model's dependence on absolute coordinates, we transform the input route into a sequence of differential features, as illustrated in Figure 2. Specifically, we transform the obfuscated route R into a differential feature sequence $R_{\text{diff}} =< r_1, r_2, \ldots, r_{L-1} >$. We decompose each differential route point r_i into two complementary feature subsets: a direction-sensitive subset r_i^{dir} and a strictly rotation-invariant subset r_i^{inv}:

$$r_i^{\text{dir}} = (\Delta x_i, \Delta y_i, \sin(\theta_i), \cos(\theta_i)) \in \mathbb{R}^4, \tag{2}$$

$$r_i^{\text{inv}} = (d_i^{\text{cum}}, \Delta\theta_i, \mathcal{K}_i) \in \mathbb{R}^3, \tag{3}$$

where Δx_i and Δy_i denote the coordinate offsets relative to the predecessor point. θ_i represents the heading angle of the current segment. d_i^{cum} indicates the cumulative distance from the starting point, calculated as $d_i^{\text{cum}} = \sum_{j=2}^{i} \|p_j - p_{j-1}\|_2$. $\Delta\theta_i$ represents the heading change rate. $\mathcal{K}_i$ denotes the curvature, defined as $\mathcal{K}_i = \frac{|\Delta\theta_i|}{\|p_i - p_{i-1}\|_2 + \epsilon}$, with ϵ being a small constant to prevent division by zero.

The invariant subset r_i^{inv} is invariant to global translation and rotation. The direction-sensitive subset r_i^{dir} is translation-invariant but not rotation-invariant. However, under a shared transformed coordinate system, these directional components can still encode useful macroscopic movement patterns, such as dominant travel directions and spatial layout biases. Therefore, the full differential feature vector is defined as

$$r_i = \left[r_i^{\text{dir}}; r_i^{\text{inv}}\right] \in \mathbb{R}^7. \tag{4}$$

Building upon this, we augment the differential sequence R_{diff} to obtain $R' =< S, r_1, r_2, \ldots, r_{L-1}, E, D >$, where S and E represent the obfuscated start and end points of the route, respectively, and are encoded in the same differential feature space. D denotes the total length of the route. To facilitate batch processing, we fix all route lengths to T, padding shorter sequences with a specific `[PAD]` token and employing a mask to indicate valid positions.

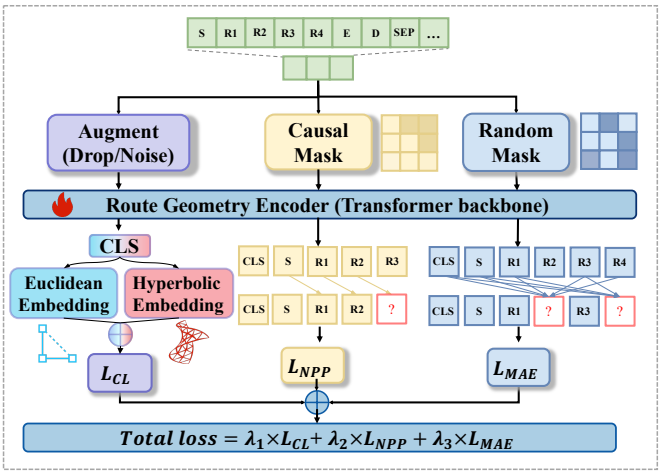

Fig. 3. Schematic illustration of the multi-task pre-training framework. The Route Encoder serves as a shared backbone to process inputs under different views. The global semantic feature z_{CLS} is utilized for Contrastive Learning (L_{CL}), and simultaneously serves as the context for Next Point Prediction (L_{NPP}) and Masked Auto-Encoding (L_{MAE}). The entire framework is optimized via a weighted sum of these three losses.

B. Route Encoder Architecture

The route encoder serves as the shared backbone for both pre-training and fine-tuning. As shown in Figure 3, we adopt a Transformer-based architecture [42] to capture long-range trajectory dependencies. Given the differential feature sequence R', each feature vector is first projected into a latent embedding space via a linear layer. A learnable special token `[CLS]` is then prepended to the head of the sequence, followed by the addition of positional encodings, denoted as z_0. The encoder takes the initial sequence $z_0 \in \mathbb{R}^{(T+1)\times d}$ as input. After processing through N encoder layers, it generates the contextualized hidden representations Z:

$$Z = \text{Encoder}(z_0) = (z_{\text{CLS}}, z_1, \ldots, z_T), \tag{5}$$

where $z_i \in \mathbb{R}^d$ represents the hidden representation of the i-th route point. Notably, the hidden state corresponding to the `[CLS]` token, denoted as z_{CLS}, is regarded as the global semantic feature vector of the entire route. It aggregates both local motion patterns and global structural information, serving as the core input for the subsequent hybrid space contrastive learning task and the downstream ETA fine-tuning. The three pretext tasks will be introduced below.

C. Hybrid Space Contrastive Learning

To better capture latent hierarchical structures and branching patterns in routes, we extend our representation beyond the confines of a single Euclidean space. To this end, we introduce a hybrid Euclidean-hyperbolic contrastive objective.

Motivation. Hyperbolic geometry has been widely used as a non-Euclidean inductive bias for data with latent hierarchical or tree-like structures. In our setting, explicit road-network topology and map-derived hierarchical priors are unavailable. Therefore, we expect the hyperbolic space to capture latent hierarchical relationships among routes, particularly multi-scale structures. Intuitively, when many routes share local subpaths and then branch into diverse route combinations, hyperbolic space can better organize such route-level relationships. Different from LH-Plugin [43], which employs hyperbolic geometry for trajectory similarity modeling, we incorporate the hybrid similarity into a contrastive objective.

Hyperbolic Projection. Given the sequence-level representation $z_{\text{CLS}} \in \mathbb{R}^d$, we split it along the feature dimension into two equal-sized components:

$$z_{\text{CLS}} = [z^E \parallel z^H], \quad z^E \in \mathbb{R}^{d/2}, \quad z^H \in \mathbb{R}^{d/2}, \tag{6}$$

where z^E is used as the Euclidean component and z^H is further projected onto the Lorentz model as the hyperbolic component. Following standard constructions in hyperbolic representation learning, we map the hyperbolic component z^H into the Lorentz model using a radial parameterization based on hyperbolic cosine and sine functions, as adopted in [43]. Specifically, a radial scaling function $\gamma(r)$ is adopted to regulate the embedding radius:

$$\begin{aligned} \Phi(z^H) &= \left(\Phi_0(z^H), \Phi_s(z^H)\right), \\ \Phi_0(z^H) &= \sqrt{\beta}\cosh(\gamma(r)), \\ \Phi_s(z^H) &= \sqrt{\beta}\sinh(\gamma(r))\frac{z^H}{r}, \quad r = \|z^H\|_2. \end{aligned} \tag{7}$$

where β is a curvature-related constant and $\gamma(r) = r^c$ controls the growth of the embedding radius, where c represents the radial compression factor. This parameterization provides increasing geometric capacity as embeddings move towards the boundary of the hyperbolic space, which helps alleviate representation crowding and improves numerical stability.

Hybrid Similarity Function. Given representations z_i and z_j, the Euclidean similarity is the standard cosine similarity $s_{ij}^E = \cos(z_i^E, z_j^E)$. For the hyperbolic component, we define the similarity metric as the negative squared Lorentz distance, ensuring that closer embeddings yield higher similarity scores:

$$s_{ij}^H = -d_L^2(\Phi(z_i^H), \Phi(z_j^H)). \tag{8}$$

This formulation aligns with the energy-based perspective of contrastive learning [44], where the distance function serves as an energy term, and minimizing the energy corresponds to maximizing similarity.

To adaptively fuse information from both geometries, we adopt the magnitude-based dynamic weighting mechanism [43], adapted to our normalized similarity metric. For any sample z_i, its Euclidean weight $w_i^E = \frac{\|z_i^E\|_2}{\|z_i^E\|_2 + \|z_i^H\|_2}$, with the corresponding hyperbolic weight $w_i^H = 1 - w_i^E$. For a pair (i, j), the hybrid weight is $\alpha_{ij}^E = (w_i^E + w_j^E)/2$. The final hybrid similarity is:

$$S_{\text{hybrid}}(z_i, z_j) = \alpha_{ij}^E s_{ij}^E + (1 - \alpha_{ij}^E) s_{ij}^H. \tag{9}$$

This design allows the model to automatically balance geometries in a data-driven manner, avoiding a manually fixed contribution from either geometry.

Contrastive Objective. Our hybrid similarity is directly embedded into the contrastive learning objective, allowing the

geometric inductive bias to actively shape the representation space during training. Based on this hybrid metric, we reformulate the contrastive loss. For a mini-batch of B routes, we construct a positive pair for each route consisting of the *original sequence* and its *stochastically augmented view*. Specifically, the augmented view is generated by applying one of two strategies with equal probability ($p = 0.5$): (1) Random Point Dropping, which removes a randomly sampled subset of trajectory points along the temporal axis; or (2) Gaussian Noise Injection, where small Gaussian noise is injected into the obfuscated coordinates and the corresponding differential features are recomputed. For any anchor representation z_i, we identify its specific positive pair $z_{p(i)}$ as the other augmented view originating from the same raw route. All other $2B - 2$ samples in the batch are treated as negative samples. The contrastive loss for the i-th sample is defined as:

$$\mathcal{L}_{\text{CL}} = -\frac{1}{2B}\sum_{i=1}^{2B}\log\frac{\exp\left(S_{\text{hybrid}}(z_i, z_{p(i)})/\tau\right)}{\sum_{\substack{j=1\\ j\neq i}}^{2B}\exp\left(S_{\text{hybrid}}(z_i, z_j)/\tau\right)}. \quad (10)$$

where τ is the temperature parameter, and $p(i)$ denotes the index of the positive view paired with sample i.

D. Masked Autoencoder (MAE)

To empower the route encoder to capture fine-grained local geometric dependencies within trajectory sequences, we introduce the Masked Autoencoder (MAE) task as one of our self-supervised pre-training objectives.

Given an input differential feature sequence R', we randomly sample a subset of indices $\mathcal{M}$ for masking with a ratio ρ. Specifically, the features of the selected trajectory points r_i are replaced with a `[MASK]` token, while the unselected points remain unchanged, and all special tokens (including `[CLS]`, start point S, end point E, and distance D) are excluded from the masking operation. The objective of the task is to reconstruct the physical differential features at the masked positions based on the remaining visible context. This mechanism effectively prevents the model from degenerating into a trivial identity mapping, ensuring that the global representation aggregated by the `[CLS]` token is built upon a profound understanding of local motion semantics. Formally, the MAE objective function is defined as the Mean Squared Error (MSE) between the predicted and ground-truth features at the masked positions: $\mathcal{L}_{\text{MAE}} = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}}\|\hat{r}_i - r_i\|_2^2$.

E. Next Point Prediction (NPP)

To further endow the encoder with the capability to capture route directionality and sequential dynamic evolution, we introduce the Next Point Prediction (NPP) task as an auxiliary self-supervised pre-training objective.

Specifically, given a trajectory point sequence R' composed of differential feature, we incorporate a causal attention mask matrix M_{causal} into the encoder to enforce unidirectional information flow along the temporal dimension. When predicting the state of the $(t + 1)$-th trajectory point, the model is permitted access only to the `[CLS]` token, the start token S, and the preceding trajectory points $r_1, \ldots, r_t$.

As a result, the encoder is compelled to capture the directional consistency and temporal dependencies of the entire route. Since the `[CLS]` token participates in modeling as accessible context at every time step, its corresponding hidden state z_{CLS} is implicitly injected with global sequence order information. This task helps the resulting route representation to distinguish between routes that are geometrically similar but traversed in opposite directions (e.g., round trips). Formally, the objective function for the NPP task is defined as the Mean Squared Error (MSE) between the predicted and ground-truth values: $\mathcal{L}_{\text{NPP}} = \frac{1}{L-1}\sum_{t=1}^{L-1}\|\hat{r}_{t+1} - r_{t+1}\|_2^2$.

Overall Pre-training Objective. To summarize, the route pre-training phase integrates three complementary objectives: Contrastive Learning ($\mathcal{L}_{\text{CL}}$) for global structural discrimination, Masked Autoencoder ($\mathcal{L}_{\text{MAE}}$) for local geometric understanding, and Next Point Prediction ($\mathcal{L}_{\text{NPP}}$) for sequential dynamic modeling. The overall loss function $\mathcal{L}_{\text{pre}}$ is a weighted sum of these components:

$$\mathcal{L}_{\text{pre}} = \lambda_1\mathcal{L}_{\text{CL}} + \lambda_2\mathcal{L}_{\text{MAE}} + \lambda_3\mathcal{L}_{\text{NPP}}, \quad (11)$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters.

F. Fine-tuning for ETA Prediction

As shown in Figure 4, the goal of the fine-tuning phase is to transfer the learned static route representations to the downstream task via external dynamic contexts, thereby achieving precise Estimated Time of Arrival (ETA) prediction.

Dynamic Context Modeling. The external context consists of departure time and optional user identifiers. For departure time, we model it as a set of cyclical temporal features, including month, day, second, and day-of-week. These attributes are encoded using sinusoidal functions, namely sine and cosine transformations, to capture temporal periodicity. For user identifiers, we employ a simple learnable embedding layer. These two components are concatenated and linearly transformed to form a unified context vector, denoted as C, which is further projected to dimension d via a linear layer with ReLU and LayerNorm, yielding $\tilde{C}$.

Gated Cross-Attention Fusion. To facilitate effective interaction between dynamic contexts and static route features, we design a Gated Cross-Attention Fusion module. Let $Z_{\text{seq}} = [z_1, z_2, \ldots, z_T] \in \mathbb{R}^{T\times d}$ denote the token-level route representations extracted from the frozen route encoder, excluding the `[CLS]` token, and let $\tilde{C} \in \mathbb{R}^d$ denote the projected dynamic context vector obtained from the context encoding layer. We employ $\tilde{C}$ as a single query token and Z_{seq} as both keys and values. Formally, the cross-attention output is computed by a multi-head attention layer with H heads:

$$Q = \tilde{C}W_Q, \quad K = Z_{\text{seq}}W_K, \quad V = Z_{\text{seq}}W_V, \quad (12)$$

$$\text{head}_h = \text{softmax}\left(\frac{Q_h K_h^\top}{\sqrt{d_h}}\right)V_h, \quad h = 1, \ldots, H, \quad (13)$$

$$z_{\text{attn}} = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\,W_O, \quad (14)$$

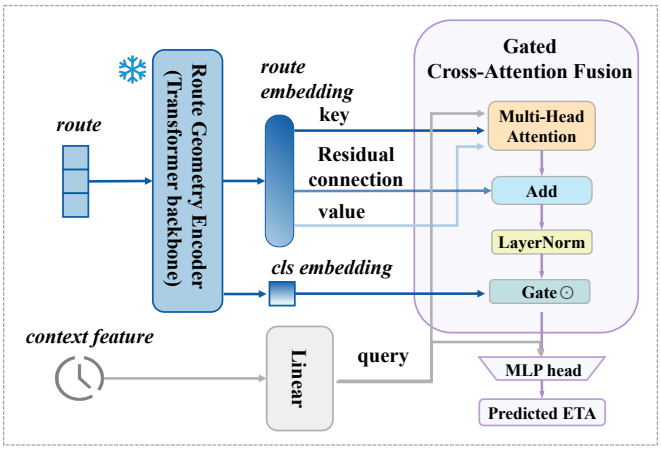

Fig. 4. Workflow of Fine-tuning. The frozen route encoder generates route embeddings, which are fused with contextual features via gated attention.

where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$, $W_O \in \mathbb{R}^{d\times d}$, and $d_h = d/H$. Since the query contains only one token, $z_{\text{attn}} \in \mathbb{R}^d$ aggregates context-dependent information from the entire route sequence. Padding positions in Z_{seq} are masked out in the computation.

To adaptively balance the context-aware token-level route representation with the global route representation encoded by z_{CLS}, we introduce a feature-wise gating mechanism. The gate vector $g \in \mathbb{R}^d$ is adaptively calculated based on the concatenation of both representations, regulating the information flow as follows:

$$g = \sigma\left(W_g[z_{\text{attn}} \parallel z_{\text{CLS}}] + b_g\right), \tag{15}$$

$$h_{\text{final}} = \text{LayerNorm}(g \odot z_{\text{attn}} + (1-g) \odot z_{\text{CLS}}), \tag{16}$$

where $\parallel$ denotes concatenation, $\sigma(\cdot)$ is the sigmoid activation function, and $\odot$ denotes element-wise multiplication. $W_g \in \mathbb{R}^{d\times 2d}$ and $b_g \in \mathbb{R}^d$ are learnable weight and bias parameters. The resulting h_{final} serves as the final context-aware route representation for the ETA prediction.

Prediction and Objective. Finally, the context-aware route representation h_{final} is concatenated with the original context vector C and fed into a Multi-Layer Perceptron (MLP) prediction head to yield the estimated arrival time $\hat{Y}$. The objective function for the fine-tuning phase is defined as the Mean Absolute Error (MAE) between the predicted and ground-truth values, i.e., $\mathcal{L}_{\text{ETA}} = \frac{1}{N}\sum_{i=1}^{N}|\hat{Y}_i - Y_i|$.

V. Experiments

In this section, we conduct experiments to evaluate the effectiveness of SHAPE from multiple perspectives. We consider both real-world obfuscated coordinates and controlled transformed coordinates to examine whether SHAPE can learn useful route representations. We aim to answer the following research questions (RQs):

RQ1: How does SHAPE perform in ETA prediction under obfuscated and controlled transformed coordinate settings?

RQ2: Can SHAPE learn radial patterns that match route scale and composition complexity?

TABLE I
Statistics of the Real-World Datasets used in Experiments.

Statistic	LaDe	Chengdu-G	Porto-G
Valid Trajectories	381,391	897,567	900,694
Number of Users	2,076	–	–
Sampling interval (s)	22.73	12.66	14.92
Avg. Sequence Length(pts)	33.43	57.93	42.89
Avg. Travel Distance (m)	1220.75	4447.99	3866.02
Avg. Travel Time (s)	709.00	682.00	583.00

RQ3: How does each component of SHAPE contribute to the final prediction accuracy?

To answer these questions, we first conduct an overall comparison on three datasets: LaDe, Chengdu-G, and Porto-G. LaDe [45] is a real-world logistics benchmark with processed coordinates, while Chengdu-G and Porto-G are controlled transformed-coordinate benchmarks constructed from public trajectory datasets. Together, these datasets allow us to evaluate ETA prediction under both real-world obfuscated coordinates and controlled coordinate transformations where explicit map resources are unavailable. Particularly, we provide both quantitative results and visual analyses to evaluate the contribution of hybrid Euclidean-hyperbolic representation learning. Furthermore, we conduct comprehensive ablation studies to investigate the contribution of each key module.

A. Experimental Setup

1) Datasets: We evaluate SHAPE on three datasets, including one real-world obfuscated-coordinate dataset and two controlled transformed-coordinate datasets. The statistics of the processed datasets are summarized in Table I.

LaDe. LaDe is a large-scale industrial logistics trajectory dataset containing courier trajectories from five cities in China. The released trajectories are mixed across cities without city labels and the available coordinates cannot be directly aligned with standard geographic coordinates or public road networks. As the processing procedure is not disclosed, we treat LaDe as a real-world obfuscated-coordinate benchmark reflecting practical map-unavailable logistics scenarios.

Chengdu-G and Porto-G. We construct controlled transformed-coordinate benchmarks from the Chengdu and Porto taxi datasets released by Wang et al. [26]. Since their GPS coordinates are available, we first project the coordinates onto a local planar coordinate system and then apply a globally consistent rigid transformation, including a rotation and a translation shared by all trajectories in the dataset. The resulting datasets are denoted as Chengdu-G and Porto-G.

2) Baselines: We compare our proposed method against the following representative baselines, ranging from traditional statistical models to state-of-the-art deep learning approaches, whose details are provided in **Appendix VIII-A**:

- Statistical methods: **LR** [46] and **GBM** [46];
- Deep Learning ETA models: **DeepTTE** [12], **WDR** [13], **HetETA** [25], **HierETA** [27], **MetaTTE** [26], **MulT-TTE** [28], **JGRM** [32] and **UVTM** [33].

TABLE II
OVERALL ETA PREDICTION PERFORMANCE. THE BEST PERFORMANCE IS IN BOLD FONT AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Type	Method	Year	LaDe			Chengdu-G			Porto-G		
			MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Statistical Methods	LR	2017	213.4843	353.3443	33.22	131.1637	165.3633	20.95	108.2190	134.7182	19.78
	GBM	2015	181.5103	314.9345	26.45	108.9888	142.0220	16.66	89.2937	116.4410	15.75
Task-specific ETA Methods	DeepTTE	2018	309.6924	409.1786	21.97	91.7728	191.6279	11.75	26.8383	33.2479	4.23
	WDR	2018	235.0062	867.5094	49.78	118.6680	156.0480	20.78	64.062	87.246	10.95
	HetETA	2020	326.6825	487.5338	47.20	113.4618	152.6373	16.85	104.1081	138.1618	17.78
	HierETA	2022	194.8958	676.0271	8.65	73.5497	100.9811	10.89	52.0040	70.3324	8.97
	MetaTTE	2022	117.8939	253.6116	20.80	357.3230	830.8262	43.51	30.3492	52.7738	5.00
	MulT-TTE	2024	377.0214	997.7171	30.46	274.4764	813.1220	17.51	72.5255	117.4901	10.65
Universal Trajectory Models	JGRM	2024	227.52	377.94	35.31	187.0982	222.1656	31.07	136.7231	163.4572	25.59
	UVTM	2024	132.6174	326.7528	19.01	354.0771	460.4489	56.15	496.7463	612.4344	65.80
Ours	**SHAPE**	–	**69.4590**	**245.1232**	**7.31**	**41.1617**	**65.3812**	**6.05**	**24.2720**	**32.0946**	**3.75**
Improv.	–	–	41.08%	3.35%	15.49%	44.04%	35.25%	44.44%	9.56%	3.60%	11.35%

3) Evaluation Metrics: We adopt three standard metrics to evaluate the performance: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). MAE and RMSE are reported in seconds, while MAPE is reported as a percentage.

Implementation details are provided in **Appendix VIII-B**.

B. RQ1: Overall Comparisons

The overall performance of SHAPE and the baseline methods on the LaDe dataset is summarized in Table II. As shown in Table II, SHAPE achieves the best overall performance across all three settings. The results on Chengdu-G and Porto-G demonstrate the advantage of SHAPE. Under globally consistent coordinate transformations, trajectories are expressed in a shared transformed coordinate system, while explicit map matching and road-segment attributes remain unavailable. The consistent performance gains in these controlled settings suggest that SHAPE can effectively exploit relative geometric and sequential route patterns from coordinate sequences alone. On LaDe, SHAPE outperforms both statistical methods and deep ETA baselines, indicating its effectiveness on real-world obfuscated logistics trajectories where the original coordinate processing procedure is unknown. Among the baselines, methods that rely on structured road-segment representations require adaptation under our coordinate-only setting. Although pseudo-segment construction enables these methods to run, their performance is generally limited by the absence of explicit road-network topology and semantic attributes.

C. RQ2: Hyperbolic Representation Analysis

For each test route, we compute the Euclidean norm $R^E = \|z^E\|_2$. The hyperbolic radius is defined as:

$$R^H = \sqrt{\beta}\,\mathrm{arccosh}\left(-\beta^{-1}\langle \Phi(z^H), \mathbf{o}\rangle_{\mathcal{L}}\right). \tag{17}$$

Here, $\mathbf{o}$ denotes the hyperbolic origin and $\langle\cdot,\cdot\rangle_{\mathcal{L}}$ is the Minkowski inner product. This allows us to examine whether radial organization emerges within route embeddings without explicit hierarchical supervision.

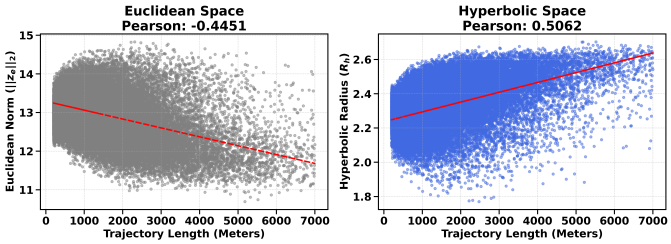

Fig. 5. The correlation between length and latent radial metrics.

As shown in Figure 5, the hyperbolic radius exhibits a positive correlation with route length, while Euclidean radius exhibits a negative correlation. This suggests that the hyperbolic radial dimension captures multi-scale route organization, placing shorter routes near the origin and longer, more complex routes near the boundary. This observation is consistent with the intended non-Euclidean inductive bias of the hyperbolic space, suggesting that it can organize routes with different spatial scales more effectively. Although route length is not a complete description of road-network hierarchy, it reflects the spatial scale and compositional complexity of a route to a certain extent. Therefore, the hyperbolic component organizes route representations more effectively in terms of spatial scale than the Euclidean component.

D. RQ3: Ablation Studies

Unless otherwise specified, all subsequent experiments are conducted on the LaDe dataset, which serves as the real-world obfuscated-coordinate benchmark in our experiments. We use this setting to examine the contribution of each key component under realistic obfuscated coordinate inputs. Specifically, we design a series of model variants from four aspects:

- **Differential Feature Decomposition.** To examine the effect of different feature subsets, we evaluate an **Invariant-only features** variant that uses only the rotation-invariant geomet-

ric components and removes direction-sensitive components such as displacement and heading features.

- **Self-supervised Pre-training Objectives.** We investigate the role of different self-supervised signals by removing the Masked Autoencoder task (**w/o MAE**), the Next-Point Prediction task (**w/o NPP**), and the Contrastive Learning objective (**w/o CL**), respectively.
- **Hybrid Geometric Representation.** To evaluate the contribution of different geometric spaces in the contrastive learning module, we compare the full hybrid design with two single-space variants: only in Euclidean space (**w/o Hyperbolic**) and only in hyperbolic space (**w/o Euclidean**).
- **Fine-tuning Fusion Module.** To verify the effectiveness of the gated cross-attention fusion module, we compare it with a simpler **Concat Fusion** variant that directly concatenates route and context representations for ETA prediction.

The experimental results are presented in Table III.

TABLE III
ABLATION RESULTS ON THE LADE DATASET.

Model	MAE	RMSE	MAPE (%)
SHAPE	**69.4590**	**245.1232**	**7.31**
Invariant-only features	72.5345	251.4103	7.69
w/o MAE	74.8960	250.2013	8.10
w/o NPP	75.7187	250.1112	8.16
w/o CL	121.5311	274.4131	14.92
w/o Hyperbolic	88.4748	255.4375	10.05
w/o Euclidean	75.3360	250.7542	8.26
Concat Fusion	72.5483	248.9240	7.80

As shown in Table III, using only the rotation-invariant feature subset leads to performance degradation compared with the full differential representation. This result supports our feature decomposition design: invariant geometric features provide stable shape-related cues, while direction-sensitive components, such as displacement and heading features, still carry useful information in the real-world obfuscated-coordinate setting. Removing any self-supervised pre-training objective degrades the performance of SHAPE. Among them, removing contrastive learning causes the largest performance drop, indicating that route-level representation learning is important for distinguishing different travel patterns. The MAE and NPP objectives also contribute to the accuracy by enhancing local geometric reconstruction and sequential dependency modeling, respectively. For the hybrid geometry design, both single-space variants perform worse than the full model. The hyperbolic-only variant outperforms the Euclidean-only variant, while the hybrid model achieves the best overall performance. These results suggest that Euclidean and hyperbolic components provide complementary inductive biases for ETA-oriented route representation learning. For the fine-tuning module, removing the gated cross-attention fusion also degrades the performance. By concatenating h_{final} with C, the prediction head can jointly exploit the context-aware route representation and the raw contextual signals, leading to more robust ETA estimation.

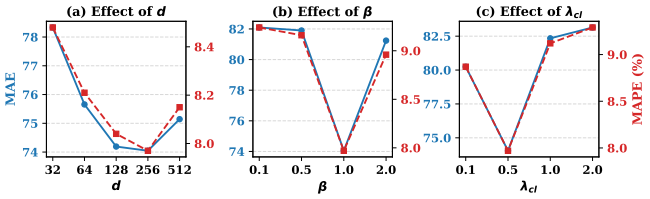

Fig. 6. Parameter Sensitivity Analysis

E. Parameter Sensitivity Analysis

We further analyze the sensitivity of SHAPE to several key hyperparameters, including the embedding dimension d, the hyperbolic curvature parameter β, and the contrastive learning weight λ_{CL}, as shown in Figure 6. Overall, SHAPE shows stable performance across a reasonable range of configurations. Across all three hyperparameters, the performance generally follows a rise-then-fall trend as the hyperparameter value increases. The best validation performance is obtained with $d = 256$, $\beta = 1.0$, and $\lambda_{\text{CL}} = 1.0$, which are used as the default settings in our experiments.

VI. CONCLUSION

In this work, we proposed **SHAPE**, a pre-trained model for ETA prediction under obfuscated-coordinate and transformed-coordinate settings. SHAPE does not require standard GPS coordinates, explicit map matching or road-network information. Instead, it learns ETA-oriented route representations from differential geometric features and sequential patterns in available coordinate sequences. We further introduced a multi-task self-supervised pre-training framework with three tasks. Finally, a gated cross-attention fusion module is employed during fine-tuning to integrate route representations with departure context. Experiments on controlled transformed-coordinate benchmarks and a real-world obfuscated-coordinate logistics benchmark show that SHAPE outperforms representative statistical and deep ETA baselines. Ablation studies verify that all proposed components contribute effectively to the final performance. These results demonstrate the potential of learning ETA-oriented route representations directly from transformed or obfuscated coordinate sequences when explicit map resources are unavailable.

VII. ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (62472125), Key Technologies Research and Development Program of Guangdong Province (2026B0909060001), Shenzhen Science and Technology Programs (ZDCY20250901111705007), and Guangdong Basic and Applied Basic Research Foundation (2025A1515011258). This work was also supported by the National Natural Science Foundation of China (62506097), Shenzhen Basic Research Program Natural Science Foundation (JCYJ20250604145542055), and Guangdong Basic and Applied Basic Research Foundation (2026A1515010223).

REFERENCES

[1] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, "Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data," in *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, 2020, pp. 3074–3082.

[2] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, pp. 3656–3663.

[3] B. Hui, D. Yan, H. Chen, and W.-S. Ku, "Trajnet: A trajectory-based deep learning model for traffic prediction," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 716–724.

[4] K. Li, L. Chen, S. Shang, P. Kalnis, and B. Yao, "Traffic congestion alleviation over dynamic road networks: Continuous optimal route combination for trip query streams," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021, pp. 3656–3662.

[5] H. Liu, Y. Li, Y. Fu, H. Mei, J. Zhou, X. Ma, and H. Xiong, "Polestar: An intelligent, efficient and national-wide public transportation routing engine," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2321–2329.

[6] H. Qin, X. Zhan, Y. Li, X. Yang, and Y. Zheng, "Network-wide traffic states imputation using self-interested coalitional learning," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 1370–1378.

[7] Z. Xu, Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, and J. Ye, "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 905–913.

[8] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[9] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1695–1704.

[10] 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 Transactions on Intelligent Systems and Technology (TIST)*, pp. 1–22, 2019.

[11] Y. Lin, H. Wan, J. Hu, S. Guo, B. Yang, Y. Lin, and C. S. Jensen, "Origin-destination travel time oracle for map-based services," *Proceedings of the ACM on Management of Data*, vol. 1, no. 3, pp. 1–27, 2023.

[12] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[13] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 858–866.

[14] J. Han, H. Liu, S. Liu, X. Chen, N. Tan, H. Chai, and H. Xiong, "ieta: A robust and scalable incremental learning framework for time-of-arrival estimation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4100–4111.

[15] P. Voigt and A. v. d. Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, 1st ed. Springer, 2017.

[16] H. Li, L. Yu, and W. He, "The impact of gdpr on global technology development," *Journal of Global Information Technology Management*, vol. 22, no. 1, pp. 1–6, 2019.

[17] J. P. Albrecht, "How the gdpr will change the world," *Eur. Data Prot. L. Rev.*, vol. 2, p. 287, 2016.

[18] C. Tankard, "What the gdpr means for businesses," *Network Security*, vol. 2016, no. 6, pp. 5–8, 2016.

[19] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 10 023–10 044, 2021.

[20] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[21] N. He, H. Madhu, N. Bui, M. Yang, and R. Ying, "Hyperbolic deep learning for foundation models: A survey," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 6021–6031.

[22] S. Takaba, T. Morita, T. Hada, T. Usami, and M. Yamaguchi, "Estimation and measurement of travel time by vehicle detectors and license plate readers," in *Vehicle Navigation and Information Systems Conference, 1991*, vol. 2. IEEE, 1991, pp. 257–267.

[23] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation research part B: methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[24] H. Yuan, G. Li, Z. Bao, and L. Feng, "Effective travel time estimation: When historical trajectories over road networks matter," in *Proceedings of the 2020 acm sigmod international conference on management of data*, 2020, pp. 2135–2149.

[25] H. Hong, Y. Lin, X. Yang, Z. Li, K. Fu, Z. Wang, X. Qie, and J. Ye, "Heteta: Heterogeneous information network embedding for estimating time of arrival," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 2444–2454.

[26] C. Wang, F. Zhao, H. Zhang, H. Luo, Y. Qin, and Y. Fang, "Fine-grained trajectory-based travel time estimation for multi-city scenarios based on deep meta-learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 716–15 728, 2022.

[27] Z. Chen, X. Xiao, Y.-J. Gong, J. Fang, N. Ma, H. Chai, and Z. Cao, "Interpreting trajectories from multiple views: A hierarchical self-attention network for estimating the time of arrival," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2771–2779.

[28] T. Liao, L. Han, Y. Xu, T. Zhu, L. Sun, and B. Du, "Multi-faceted route representation learning for travel time estimation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 11 782–11 793, 2024.

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

[30] P. Yang, H. Wang, Y. Zhang, L. Qin, W. Zhang, and X. Lin, "T3s: Effective representation learning for trajectory similarity computation," in *2021 IEEE 37th international conference on data engineering (ICDE)*. IEEE, 2021, pp. 2183–2188.

[31] Y. Chen, X. Li, G. Cong, Z. Bao, C. Long, Y. Liu, A. K. Chandran, and R. Ellison, "Robust road network representation learning: When traffic patterns meet traveling semantics," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 211–220.

[32] Z. Ma, Z. Tu, X. Chen, Y. Zhang, D. Xia, G. Zhou, Y. Chen, Y. Zheng, and J. Gong, "More than routing: Joint gps and route modeling for refine trajectory representation learning," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3064–3075.

[33] Y. Lin, J. Hu, S. Guo, B. Yang, C. S. Jensen, Y. Lin, and H. Wan, "Uvtm: Universal vehicle trajectory modeling with st feature domain generation," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[34] Z. Zhou, Y. Lin, H. Wen, S. Guo, J. Hu, Y. Lin, and H. Wan, "Trajcogn: Leveraging llms for cognizing movement patterns and travel purposes from trajectories," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025, pp. 3698–3706.

[35] Y. Wei, Y. Lin, H. Gao, R. Xu, S. B. Yang, and J. Hu, "Path-llm: A multi-modal path representation learning by aligning and fusing with large language models," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 2289–2298.

[36] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 1241–1250.

[37] X. Meng, S. Sun, X. Zhang, Q. Leng, and J. Fang, "A survey on trajectory representation learning methods," *Frontiers of Computer Science*, vol. 19, no. 12, p. 1912379, 2025.

[38] L. Yao, Z. Chen, H. Hu, G. Wu, and B. Wu, "Privacy preservation for trajectory publication based on differential privacy," *ACM Transactions on Intelligent Systems and Technology (TIST)*, pp. 1–21, 2022.

[39] G. Mai, Y. Xie, X. Jia, N. Lao, J. Rao, Q. Zhu, and Z. Liu, "The evolution of geospatial artificial intelligence," in *Geoai and human geography: the dawn of a new spatial intelligence era*. Springer, 2025, pp. 13–27.

[40] P. Singh, J. Rathi, and P. B. Patel, "Sadhe: secure anomaly detection for gps trajectory based on homomorphic encryption," in *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, 2024, pp. 139–141.

[41] J. J. Yu, "Crate: privacy-preserving travel time estimation," *IEEE Transactions on Knowledge and Data Engineering*, 2025.
[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[43] J. Si, H. Yuan, N. Jiang, M. Chen, X. Ma, and S. Wang, "Towards robust trajectory embedding for similarity computation: When triangle inequality violations in distance metrics matter," in *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, 2025, pp. 1–14.
[44] B. Kim and J. C. Ye, "Energy-based contrastive learning of visual representations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4358–4369, 2022.
[45] L. Wu, H. Wen, H. Hu, X. Mao, Y. Xia, E. Shan, J. Zheng, J. Lou, Y. Liang, L. Yang, R. Zimmermann, Y. Lin, and H. Wan, "Lade: The first comprehensive last-mile express dataset from industry," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, p. 5991–6002.
[46] W. Lan, Y. Xu, and B. Zhao, "Travel time estimation without road networks: an urban morphological layout representation approach," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 1772–1778.
[47] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 308–324, 2015.

VIII. Appendix

A. Details of baselines

- **LR (Linear Regression) [46]:** A statistical method models travel time through linear regression on statistical features.
- **GBM (Gradient Boosting Machine) [47]:** A non-linear regression method based on gradient boosting decision trees.
- **DeepTTE:** An end-to-end framework that captures spatio-temporal dependencies via geo-convolutional layers and LSTMs. It employs multi-task learning to simultaneously estimate the travel time of the entire path and local segments.
- **WDR:** A Wide-Deep-Recurrent learning model that jointly trains a wide linear model, deep neural networks, and recurrent neural networks for ETA prediction.
- **HetETA:** An approach to combining heterogeneous information networks with GNNs for ETA prediction.
- **HierETA:** The first method to employ multi-view trajectory representation, explicitly modeling trajectory structure from the perspectives of segments, links, and intersections.
- **MetaTTE:** A meta-learning-based framework designed to address complex spatio-temporal dependencies across multi-city scenarios.
- **MulT-TTE:** A multi-dimensional route representation learning framework that decomposes routes into trajectory, attribute, and semantic sequences.
- **JGRM:** The first method that models GPS trajectories and routes jointly and leverages self-supervised techniques to enhance the representation capability of trajectory data.
- **UVTM:** A universal trajectory pre-training framework designed to handle sparse and feature-incomplete trajectories via domain-wise masking and generative reconstruction, where ETA prediction is formulated as an OD-based travel time estimation task.

Note that some baselines (e.g., HierETA, MulT-TTE) originally rely on fine-grained road segments or semantic attributes, which are unavailable in our setting. To ensure a fair comparison, we adapt these methods by partitioning raw trajectories into pseudo-segments via stay-point detection, and padding missing attribute fields with zeros.

B. Implementation Details

Data Preprocessing. All datasets used in this paper are publicly available. For LaDe, we follow the official 6:2:2 train/validation/test split [45], and filter out invalid trajectories with duration < 300s, travel distance < 200m, or trajectory length outside $[14, 50]$. For Chengdu-G and Porto-G, we use the predefined splits released by MetaTTE [26], which follow an approximate 7:1:2 protocol. All compared methods use exactly the same split on each dataset. All numerical features are standardized using Z-score normalization.

Model Configuration. In the pre-training phase, the route encoder consists of 4 stacked Transformer blocks, with a hidden dimension of 256 and 4 attention heads. The dropout rate is set to 0.1. For the hyperbolic geometry settings, the curvature parameter β is fixed at 1.0, and the radial compression factor c is set to 0.25. For positive pair construction in contrastive learning, we apply random point dropping with a probability of 30%, and inject Gaussian noise sampled from $\mathcal{N}(0, 10)$ (in meters) to the obfuscated coordinates, followed by recomputation of differential features. The weights for the pre-training loss components are set to $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, and $\lambda_3 = 0.5$. We employ a linear warm-up strategy for contrastive learning, which lasts for the first 5 epochs. In the fine-tuning phase, the hidden dimension remains 256, and the cross-attention module also utilizes 4 heads.

Training Setup. We use the Adam optimizer for model training. The initial learning rate is set to 1×10^{-4} for both pre-training and fine-tuning, with a batch size of 64. For both pre-training and fine-tuning phases, the maximum number of epochs is set to 100, accompanied by an early stopping mechanism with a patience of 10 epochs to prevent overfitting. Each experiment was conducted once due to the computational cost of large-scale pre-training and fine-tuning, and we report the test performance from this single run.

Environment. All experiments are implemented using the PyTorch framework and conducted on a server running Ubuntu 22.04.3 LTS, equipped with an Intel® Xeon® Silver 4310 CPU @ 2.10GHz and an NVIDIA A800 PCIe 80GB GPU.

Complexity Analysis. Let B, T, d, L, and N denote the batch size, maximum trajectory length, hidden dimension, number of Transformer layers, and number of trajectories, respectively. The differential feature extraction costs $O(NT)$ time and $O(NTD_f)$ storage, where D_f is the feature dimension. For each mini-batch, the Transformer-based route encoder has time complexity $O(BL(T^2d+Td^2))$ and training memory complexity $O(BL(Td+T^2))$. The contrastive objective additionally introduces $O(B^2d)$ time and $O(B^2)$ memory for pairwise similarity computation, while the remaining pre-training and fine-tuning modules are dominated by the encoder. The context-route cross-attention in fine-tuning adds at most $O(BTd^2)$ time and is dominated by the Transformer encoder.