From Moves to Paths: A Hierarchical Framework for Trajectory Representation Learning

Chundong, Wang[1, 3] **Xiangtian, Zheng**[1] **Qingbo, Hao**[1,2] **Yongxin, Zhao**[1] **Yixuan, Song**[1] **Jia, Li**[1]

[1]School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China
[2]Technical College for the Deaf, Tianjin University of Technology, Tianjin, China
[3]Tianjin Police Institute, Tianjin, China

Abstract

Trajectory representation learning (TRL) seeks to convert trajectory data into low-dimensional embeddings for various downstream tasks. Existing methods are limited to a single perspective: GPS-based approaches capture dynamic details but lack semantic context, while route-based methods preserve structure but lose fine-grained motion patterns. To address these limitations, we propose a trajectory representation model with multi-perspective fusion, MPH, where M denotes the Motion modality (GPS), P denotes the planned path modality (Route), andH signifies the Hierarchical encoding and fusion strategy. MPH first map-matches raw GPS trajectories to route sequences and designs dedicated encoders to extract dynamic behavioral patterns and static semantic features respectively. Based on this, MPH leverages a cross-attention mechanism for modality interaction and fusion, producing fused route representations that are hierarchically aggregated into trajectory-level embeddings. Furthermore, we introduce two self-supervised tasks to train the model. Contrastive learning is employed to align road segment representations across the two perspectives, while the dual-mask prediction task strengthens contextual modeling by jointly reconstructing masked road segment identities and their corresponding temporal information. Experiments on two real-world datasets show that MPH outperforms all baselines across different tasks.

1 INTRODUCTION

The proliferation of GPS technology and mobile Internet has given rise to massive trajectory datasets, which serve as a valuable foundation for intelligent transportation systems Lan et al. [2023], Wang et al. [2021], urban planning He et al. [2020], and related applications Wang et al. [2024], while also revealing rich movement patterns and behavioral signals Deng et al. [2025]. However, trajectories are inherently high-dimensional, long, and noisy sequences. Transforming raw GPS point streams into low-dimensional, dense, and semantically rich vector embeddings thus remains a critical bottleneck for unlocking their value in downstream tasks.

The core challenge in trajectory representation learning(TRL) lies in effectively embedding complex trajectory data into a low-dimensional vector space. In recent years, significant progress has been made in this field. Early work treat trajectories as simple sequences and used RNNs or LSTM variants to model temporal dependencies Yao et al. [2017], Fu and Lee [2020], Mikolov et al. [2013], but redundancy and noise in GPS data limited their effectiveness. To address this, map matching is introduced to convert raw GPS points into ordered road-segment sequences that better reflect real routes Lou et al. [2009], Yang and Gidofalvi [2018]. Building on road-segment representations, Graph Neural Networks(GNNs) model the road network as a graph to capture topological constraints and spatial relationships among segments Kipf and Welling [2017]. At the same time, inspired by the natural language processing (NLP) pretraining Devlin et al. [2019], Yan et al. [2021], trajectory pretraining models based on masked language models (MLMs) have emerged Vaswani et al. [2017], Devlin et al. [2019], using self-supervised tasks like masked prediction to learn general trajectory patterns from unlabeled data, reducing reliance on labeled data.

Despite the significant progress made by the aforementioned methods, trajectory representation still faces two major challenges:

- First, most models rely on a single data perspective, focusing either on the static topology of the road network or the dynamic GPS point sequence of trajectories, failing to integrate of dynamic behaviors and static semantics. Taking the trajectory shown in Figure 1 as an

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:7121–7136.

example, its motion can be decomposed into several stages: from p1 to p2 is the starting phase, p2 to p3 involves a turn, p3 to p4 is steady straight movement, p4 to p5 involves another turn, and p5 to p6 includes acceleration. These micro-motion features can be directly extracted from the raw GPS sequence. while its origin in a residential area and destination in a business district reveal the commuting intent. Both are essential for tasks like destination inference.

- Second, multi-source information fusion remains simplistic and lack sophisticated interaction mechanisms to adaptively weigh different attributes across contexts, which can even cause feature conflicts. For instance, when a vehicle moves slowly on a highway, naive fusion may reconcile the static high-speed road attribute with the actual low-speed dynamic feature poorly, weakening the model's discriminative power and generalization. This demands context-aware weighting mechanisms to resolve such conflicts.

To tackle these challenges, we propose MPH, an efficient TRL framework for joint learning of GPS trajectories and road-segment trajectories. Since GPS trajectories and route trajectories are highly complementary in information content, we draw inspiration from the computer vision domain, where different modalities such as RGB images and textual descriptions are integrated to achieve comprehensive scene understanding Li et al. [2023, 2021], and apply a similar approach to deeply fuse the two modalities. Specifically, MPH designs dedicated encoders for GPS and road-segment trajectories to extract features from both perspectives, and employs a cross-modal interaction mechanism for synchronized representation learning. To enhance MPH's representational capacity, contrastive loss is introduced to bring representations of the same road segment across modalities closer, while a mask prediction task encourages the model to capture the contextual dependencies of trajectories. This integrated approach effectively mines latent cross-modal correlations, delivering superior performance across multiple downstream tasks.

The main contributions of this work are as follows:

- Most existing trajectory representation methods rely on a single modality and thus underutilize the rich semantic and dynamic information embedded in trajectories. To address this limitation, we propose MPH, a unified framework that jointly encodes GPS trajectories for motion dynamics and route trajectories for travel semantics, enabling more comprehensive trajectory representations.
- To achieve effective cross-modal fusion and complementarity, we introduce a hierarchical encoding strategy. At the road-segment level, a cross-attention mechanism aligns each GPS sub-trajectory with its corresponding route segment, adaptively integrating dynamic motion features with static semantic information while preserving fine-grained locality. The resulting segment-level representations are then aggregated into a holistic trajectory embedding.
- Extensive experiments on two real-world datasets, including thorough ablation studies and comparisons with state-of-the-art baselines, demonstrate that MPH achieves superior performance across multiple downstream tasks, validating its effectiveness and practicality.

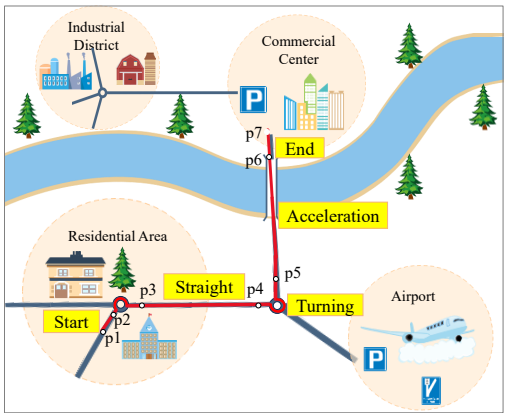


Figure 1: An example trajectory for demonstrating semantic information and motion information.

2 RELATED WORK

2.1 SEQUENCE-TO-SEQUENCE TRAJECTORY REPRESENTATION METHODS

Early trajectory representative works are as follows: Word2Vec Mikolov et al. [2013] models trajectories as raw GPS point sequences to produce vector embeddings, while DeepWalk Perozzi et al. [2014] and Node2Vec Grover and Leskovec [2016] generate road-segment sequences via random walks for learning embeddings. Subsequently, sequence-to-sequence architectures gained prominence. Traj2Vec Yao et al. [2017] uses LSTM for temporal dependency capture via sequence reconstruction. T2Vec Li et al. [2018] employs Skip-gram to reconstruct low-sampling-rate trajectories for similarity computation. GM-VSAE Liu et al. [2020] adopts a Gaussian Mixture Variational Sequence Autoencoder for anomaly detection. Trembr Fu and Lee [2020] introduces refined spatiotemporal modeling with an RNN encoder-decoder. Meanwhile, GNNs have gained popularity for their ability to adapt to the road network's topological structure. GAE Kipf and Welling [2016] reconstructs the adjacency matrix in a graph autoencoder for topology-aligned node embeddings. RFN Jepsen et al. [2020] proposes relational fusion graph convolution to incorporate multi-relational information. HRNR Wu et al. [2020]

adopts hierarchical GNNs for multi-scale structural modeling, and PIM Yang et al. [2021] together with its enhanced version PIM-TF effectively aligning road network and trajectory representations through mutual information maximization.

2.2 SELF-SUPERVISED TRAJECTORY REPRESENTATION METHODS

However, sequence-to-sequence models suffer from information loss when dealing with long sequences and are heavily dependent on large-scale labeled data for supervised training. Subsequent research has shifted towards self-supervised representation learning, aiming to learn general representations from the inherent structure of the data. Models trained on pretext tasks without manual labeling can capture deep patterns from the data. CTLE Lin et al. [2021] uses a bidirectional Transformer encoder with mask position and mask time prediction as two core self-supervised tasks to learn trajectory location embedding features. CSTRM Liu et al. [2022] models GPS trajectories using a lightweight BERT model, combining mask prediction and contrastive learning for training. START Jiang et al. [2023] also employs a BERT model, incorporating data augmentation techniques and mask strategies for model training, and integrates travel semantics together with temporal patterns to enhance trajectory representation capabilities. JCLRNT Mao et al. [2022] utilizes the Transformer model and is trained through three-level contrastive learning of road-road, track-trajectory, and road-trajectory. TrajCL Chang et al. [2023] uses a dual-feature sub-attention backbone encoder to capture the spatial and structural patterns of trajectories, optimizing the representations through trajectory contrastive learning. LighPath Yang et al. [2023] uses a sparse Transformer encoder, training with a path reconstruction task and cross-view, cross-network contrastive learning tasks. JGRM Ma et al. [2024] uses dual encoders to embed both route and GPS trajectories, integrates information through modality interactors, and designs mask prediction and cross-modal contrastive tasks for training the model.

Despite the significant progress achieved by the aforementioned methods, existing models still fail to adequately capture the complex motion patterns within road segments and overlook the rich semantic information inherent in trajectories.

3 PRLIMINARIES

Definition 1. ***(GPS Trajectory)*** *GPS Trajectory $T_{gps} = \{p_1, p_2, ..., p_\tau\}$ is an ordered sequence collected by the GPS positioning system in the continuous time dimension, composed of a series of spatiotemporal points. Each spatiotemporal point $p_i =< lat_i, lng_i, t_i >$ consists of latitude, longitude, and timestamp.*

Definition 2. ***(Road Network Graph)*** *A road network is formally represented as a directed graph $\mathcal{G} = (\mathcal{V}, A)$. Each element $v_i \in \mathcal{V}$ corresponds to a physical road segment. This choice aligns with trajectory data characteristics—trajectories are inherently sequences of road segments mapped from GPS points, enabling direct alignment between trajectory structure and graph vertices. $A \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ is a binary matrix where $A[i][j] = 1$ only if road segments v_i and v_j share a common intersection in the real road network; otherwise, $A[i][j] = 0$.*

Definition 3. ***(Route Trajectory)*** *Route Trajectory $T_{route} = \{r_1, r_2, ..., r_\tau\}$ is an ordered sequence represented by road segments, obtained from the GPS trajectory T_{gps} through map matching techniques. Each point $r_i =< s_i, t_i >$ consists of a road segment ID and a timestamp.*

Problem Statement 1. *The core task of trajectory representation is to learn a mapping function $f_\Theta(T_{gps}, T_{route}, \mathcal{G}) \rightarrow \mathbf{h} \in \mathbb{R}^d$, given the GPS sequence of the trajectory T_{gps}, its associated road segment sequence T_{route}, and the corresponding city's road network structure $\mathcal{G}$. This function maps the trajectory to a d-dimensional unified vector embedding, which captures the intrinsic features of the trajectory and can be directly applied to various downstream tasks.*

4 METHODOLOGY

The overall framework of the model is shown in Figure 2, and it mainly consists of three components: the GPS encoder, the Route encoder, and the modality fusion module. We employ two self-supervised tasks to train MPH, namely the contrastive loss and the mask loss.

4.1 GPS ENCODER

To capture fine-grained dynamic behavior features within road segments, we design a dynamic road segment encoder based on bidirectional gated recurrent units (Bi-GRU) to extract motion pattern features with temporal dependencies from the raw GPS point sequences.

First, we construct trajectory sequences containing multi-dimensional kinematic feature vectors. Each GPS point's feature vector includes longitude, latitude, traveled distance, acceleration, instantaneous speed, jerk, and motion direction angle, providing a complete description of the vehicle's motion state. Then, each trajectory is divided into several sub-trajectories according to the road segments it passes through. The allocation matrix $\mathbf{B}_\tau$ records the number of GPS points in each sub-trajectory of trajectory τ. A trajectory τ is divided into K sub-trajectories according to road segments, denoted as $\tau = \{s_1, s_2, \ldots, s_K\}$, where each sub-trajectory s_j corresponds to the trajectory segment on road segment r_j ($r_j \in \mathcal{V}$). The j-th sub-trajectory s_j

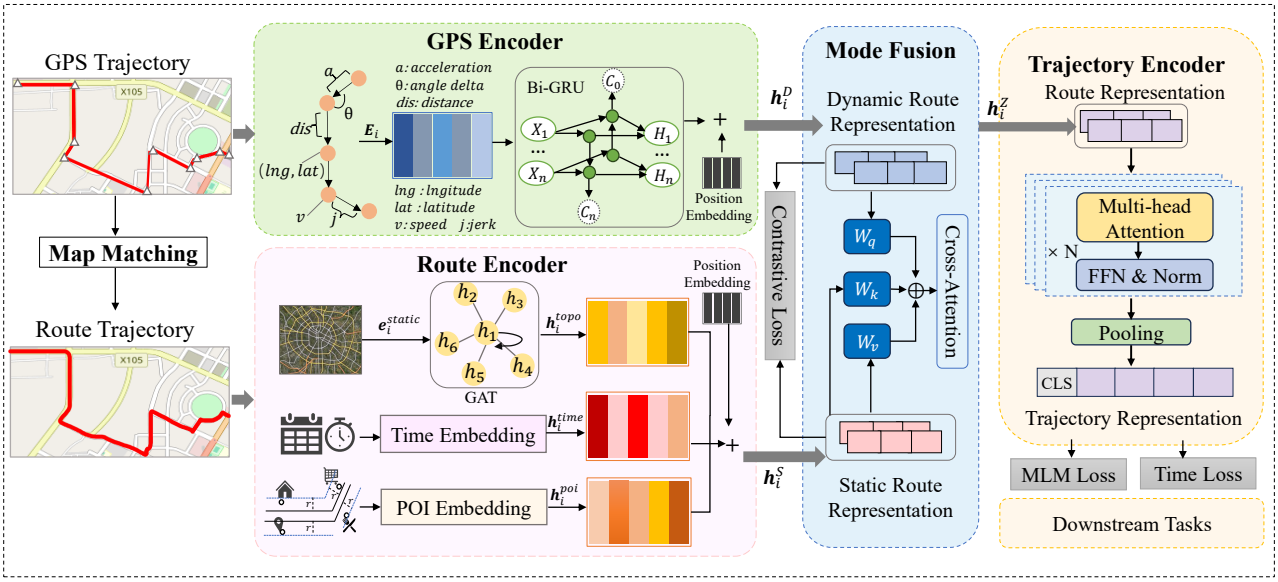


Figure 2: Overall framework of MPH, of which the original GPS trajectory is processed by map matching to obtain the road segment trajectory, these two types of trajectories are encoded by two encoders respectively, then undergo mode fusion, and finally pass through a trajectory encoder to obtain the final trajectory representation.

contains b_j^τ GPS points, which form a subset of the original GPS trajectory. The raw feature of sub-trajectory s_j is represented as $\mathbf{X}_j \in \mathbb{R}^{b_j^\tau \times 7}$, where b_j^τ is the number of GPS points on s_j, and 7 denotes the dimension of per-point features. We first apply a linear transformation layer for nonlinear mapping and enhancement:

$$\mathbf{E}_i = \text{LayerNorm}(\text{ReLU}(\mathbf{W}_{feat} \cdot s_i + \mathbf{b}_{feat})), \quad (1)$$

where $\mathbf{W}_{feat}$ and $\mathbf{b}_{feat}$ are the linear layer parameters. The enhanced sequence features are then input into the Bi-GRU network to capture both forward progression and backward contextual dependencies of motion patterns. Specifically, the forward GRU processes the sequence in temporal order to capture motion trends from start to end, while the backward GRU processes the sequence in reverse to model context from end to start:

$$\begin{aligned} \overrightarrow{\mathbf{h}_i} &= \text{GRU}_{\text{forward}}(E_i; \boldsymbol{\theta}_{\overrightarrow{gru}}) \in \mathbb{R}^H, \\ \overleftarrow{\mathbf{h}_i} &= \text{GRU}_{\text{backward}}(E_i; \boldsymbol{\theta}_{\overleftarrow{gru}}) \in \mathbb{R}^H, \end{aligned} \quad (2)$$

where H is the hidden dimension, $\boldsymbol{\theta}_{\overrightarrow{gru}}$ represents the forward GRU parameters, and $\boldsymbol{\theta}_{\overleftarrow{gru}}$ denotes the backward GRU parameters. The final hidden states from both directions are concatenated to integrate bidirectional temporal information, forming the dynamic representation of the road segment $\mathbf{h}_i^{road}$, which contains local motion details within the segment:

$$\mathbf{h}_i^{road} = [\overrightarrow{\mathbf{h}_r} \oplus \overleftarrow{\mathbf{h}_r}] \in \mathbb{R}^{2H}. \quad (3)$$

This representation is then projected into the model's unified feature dimension d and incorporates the segment's temporal position within the trajectory:

$$\mathbf{h}_i^{proj} = \text{LayerNorm}(\mathbf{W}_{proj} \cdot \mathbf{h}_i^{road} + \mathbf{b}_{proj}) \in \mathbb{R}^d, \quad (4)$$

where $\mathbf{W}_{proj} \in \mathbb{R}^{d \times 2H}$ and $\mathbf{b}_{proj} \in \mathbb{R}^d$ are the projection layer parameters.

Finally, the segment positional encoding $\mathbf{pe}_i$ is added to obtain the dynamic road segment representation:

$$\mathbf{h}_i^D = \mathbf{h}_i^{proj} + \mathbf{pe}_i, \quad (5)$$

where $\mathbf{h}_i^D$ contains both fine-grained motion patterns within the segment and trajectory-level contextual information, providing a high-quality dynamic feature foundation for subsequent cross-modal fusion.

4.2 ROUTE ENCODER

Traditional road network encoding methods are usually limited to static features and topological structures, often ignoring semantic contextual information. In fact, the distribution of Points of Interest (POIs) around a road segment profoundly reflects the socio-economic functions of the area and significantly affects traffic flow and mobility patterns. To address this, we propose a road network encoder that integrates multiple sources of features, systematically combining topology, static attributes, and semantic information.

For each road segment r_i, categorical features like road type, lane type are mapped to dense vectors through embedding layers, while numerical features like length and speed limit are nonlinearly transformed via a multi-layer perceptron (MLP), and ultimately fused into a unified static feature embedding $\mathbf{e}_i^{static}$.

Based on the constructed road network graph $\mathcal{G} = (\mathcal{V}, A)$, we adopt a Graph Attention Network (GAT) to model spatial dependencies between road segments. The multi-head self-attention mechanism dynamically learns the importance weights of neighboring nodes, achieving selective aggregation of topological context. For each road segment r_i, using $\mathbf{e}_i^{static}$ as input, the GAT encoder outputs a topology-aware enhanced representation $\mathbf{h}_i^{topo}$:

$$\mathbf{h}_i^{topo} = \mathrm{GAT}(\mathbf{e}_i^{static}, A). \tag{6}$$

In terms of semantic enhancement, to efficiently capture the functional semantics of road segments, we select 12 representative POI categories: dining services, shopping services, lifestyle services, healthcare services, accommodation services, scenic spots, business and residential areas, government institutions and social organizations, educational and cultural services, transportation facilities, financial and insurance services, companies and enterprises. This classification system comprehensively covers the main destination types of urban residents, effectively characterizing regional functional features. For each road segment r_i, we take its geometric center as the origin and construct a buffer with a preset radius σ. The function of this buffer zone is to filter out POIs strongly related to r_i , aggregating all POIs within this range:

$$POI_{s_i} = \{p \mid p \in AllPOIs, \mathrm{distance}(p, \mathrm{center}(s_i)) \leq \sigma. \tag{7}$$

The POIs around the geometric center can better reflect the core semantics of the section such as commercial areas, residential areas, enhancing the discrimination of the representation. The details about radius can refer to Appendix A.3. By counting the occurrence frequency of each POI category on the road, we construct a 12-dimensional multi-hot vector $\mathbf{v}_i$ and project it through a dedicated embedding layer to encode the functional context of the segment, obtaining the vector $\mathbf{h}_i^{poi}$:

$$\begin{gathered}\mathbf{v}_i = \left[\textstyle\sum_{p \in POI_{s_i}} \mathbb{I}(p \in \mathrm{c}_1), \ldots, \sum_{p \in POI_{s_i}} \mathbb{I}(p \in \mathrm{c}_{12})\right]^{\mathrm{T}}, \\ \mathbf{h}_i^{poi} = \sigma(x * \mathrm{Embedding}(\mathbf{v}_i) + \mathbf{b}).\end{gathered} \tag{8}$$

To effectively capture the temporal periodic patterns of trajectories, we extract minute-level and week-level temporal information for the timestamps of each road segment r_i traversed by the trajectory Kazemi et al. [2019], combining them into a feature representing the hour within the week. This is then mapped into a unified temporal representation vector $\mathbf{h}_i^{time}$ to more accurately capture the joint periodic patterns:

$$\mathbf{h}_i^{time} = \mathbf{t}_{week_i} \times 24 + \mathbf{t}_{hour_i}, \tag{9}$$

where $\mathbf{t}_{week_i}$ and $\mathbf{t}_{hour_i}$ represent the week encoding and hour encoding for the i-th road segment, respectively.

To characterize relative spatiotemporal distances between segments, we further introduce a positional encoding $\mathbf{h}_i^{pos}$ for each road segment.

Finally, we fuse the above features—topology-aware representation, semantic vector, positional information, and temporal context—to construct a comprehensive static road segment representation:

$$\mathbf{h}_i^S = \mathrm{Linear}(\mathrm{Concat}(\mathbf{h}_i^{topo}, \mathbf{h}_i^{poi}, \mathbf{h}_i^{time}, \mathbf{h}_i^{pos})). \tag{10}$$

4.3 MODE INTERACTOR

To achieve deep fusion of dynamic motion patterns and static road network semantics, we design a bimodal fusion module based on cross-attention mechanism. This module takes the dynamic motion representation $\mathbf{h}_i^D$ from the GPS encoder as the query vector ($\mathbf{Q}$) and the comprehensive static representation $\mathbf{h}_i^S$ from the Route encoder as the key ($\mathbf{K}$) and value ($\mathbf{V}$) vectors, producing the fused road segment representation $\mathbf{h}_i^Z$.

We first linearly project the input features into a unified attention space and then compute cross-attention as follows:

$$\begin{gathered}\mathbf{Q} = \mathbf{h}_i^D \mathbf{W}^Q, \mathbf{K} = \mathbf{V} = \mathbf{h}_i^S \mathbf{W}^K, \\ \mathbf{h}_i^Z = \mathrm{CrossAttention}(\mathbf{h}_i^S, \mathbf{h}_i^D) = \mathrm{Softmax}(\tfrac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V},\end{gathered} \tag{11}$$

where $\mathbf{W}^Q \in \mathbb{R}^{s_1 \times d_k}$ and $\mathbf{W}^K \in \mathbb{R}^{d_2 \times d_k}$ are learnable matrices, and d_k is the dimension of the query set.

By using motion features as $\mathbf{Q}$ and static features as $\mathbf{K}$-$\mathbf{V}$, our design is grounded in human spatial cognition: moving agents typically perceive and understand their environment based on current motion states such as speed and direction changes—drivers adjust attention to road slopes and curves based on real-time speed. Our design simulates this "perceive environment through behavior" cognition: using dynamic motion states as queries to retrieve the most relevant semantic information from rich static environment features, thereby understanding not only "what is happening" but also"in what environment this behavior occurs".

In contrast, using static features as $\mathbf{Q}$ and motion features as $\mathbf{K}$-$\mathbf{V}$ implies a mechanical "environment determines behavior" assumption, which contradicts the autonomous decision-making nature of moving agents in real traffic and limits the model's ability to capture complex interactions between behavior and environment. The experiment results can refer to the Appendix B.3.

Through cross-attention, we generate a fused representation $\mathbf{h}_i^Z$ for each road segment that retains original motion details while selectively integrating the most relevant static and semantic features. These token sequences are then input into a multi-layer Transformer Vaswani et al. [2017] encoder. Its self-attention mechanism learns the global context dependencies of the trajectory, ultimately producing a unified trajectory representation $\mathbf{H}\tau \in \mathbb{R}^d$ that integrates local fine-grained dynamics and global trajectory patterns:

$$\mathbf{H}_\tau = \text{TransEncoder}(\text{concat}(\,[\text{CLS}]\,, \mathbf{h}_1^Z, ...\mathbf{h}_i^Z)), \quad (12)$$

where [CLS] is the trajectory start token. Before being fed into the Transformer, the road segment representations are concatenated with the start token to form the complete trajectory representation. The encoder output yields trajectory embeddings with contextual awareness, and this hierarchical encoding framework—from point-level to segment-level fusion, and then trajectory-level integration—ensures the model effectively understands and represents complex mobility behavior across multiple spatiotemporal scales.

4.4 SELF-SUPERVISED TASKS

4.4.1 Contrastive Loss

In the research of multimodal learning research in computer vision and natural language processing, a core consensus is that effective fusion often relies on good alignment Radford et al. [2021]. Therefore, we introduce a contrastive learning task aimed at aligning the dynamic and static views of the same road segment. The basic assumption is that for any road segment r_i , its dynamic representation from the GPS encoder $\mathbf{h}_i^D$ and its static representation from the Route encoder $\mathbf{h}_i^S$ should correspond to the same semantic entity, and thus should be close to each other in the embedding space. We treat the dynamic-static representation pair of the same road segment $(\mathbf{h}_i^D, \mathbf{h}_i^S)$ as a positive sample pair, and representation pairs of different road segments $(\mathbf{h}_i^D, \mathbf{h}_j^S)$ (where $i \neq j$) as negative sample pairs.

To avoid the influence of dimensional and scale inconsistencies on similarity computation, we first map both types of representations to a shared semantic space:

$$\begin{aligned} \mathbf{h}_i^{D'} &= \text{Linear}(\mathbf{h}_i^D), \\ \mathbf{h}_i^{S'} &= \text{Linear}(\mathbf{h}_i^S). \end{aligned} \quad (13)$$

Based on this, we apply the InfoNCE loss function:

$$\mathcal{L}_{cont} = -\log \frac{\exp(\text{sim}(\mathbf{h}_i^{D'}, \mathbf{h}_i^{S'})/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(\mathbf{h}_i^{D'}, \mathbf{h}_i^{S'})/\tau)}, \quad (14)$$

this loss encourages the model to pull positive sample pairs closer in the embedding space while pushing negative sample pairs farther apart in the representation space.

4.4.2 Mask Loss

The design of the mask loss is inspired by the MLM in NLP Devlin et al. [2019], which has been adapted for trajectory data. Considering that trajectory data exhibit significant spatial and behavioral continuity, a simple single-segment mask is insufficiently challenging for the model and fails to capture complex mobility patterns. Therefore, we adopt a continuous masking strategy: the start position of the mask in a trajectory is determined with probability *p*, and a consecutive sequence of road segments is masked with an average length of *l*. The masked road segment IDs are replaced with a special [MASK] token, and the model is required to reconstruct the IDs and travel intervals of the masked segments based on contextual information. The loss is computed as follows:

$$\mathcal{L}_{mask} = \mathcal{L}_{mask}^{road} + \mathcal{L}_{mask}^{time}, \quad (15)$$

$$\mathcal{L}_{mask}^{road} = -\frac{1}{|M_S|} \sum_{s_i \in M_S} \log \frac{\exp(\hat{y}_{s_i})}{\sum_{s_j \in S} \exp(\hat{y}_{s_j})}, \quad (16)$$

$$\mathcal{L}_{mask}^{time} = \frac{1}{|M_T|} \sum_{t_i \in M_T} |\hat{y}_{t_i} - t_i|. \quad (17)$$

The model's final objective function is the weighted sum of the two self-supervised tasks:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{cont} + \lambda_2 \mathcal{L}_{mask}, \quad (18)$$

where λ_1 and λ_2 are balancing hypeparameters.

5 EXPERIMENTS

5.1 EXPERIMENT SETTINGS

We evaluate the model on two real-world datasets to address the following questions:

- **RQ1**: How does MPH compare to other state-of-the-art TRL methods in terms of performance on different downstream tasks?
- **RQ2**: How does each module in MPH contribute to the overall model performance?
- **RQ3**: How effective is the pre-training process of MPH?
- **RQ4**: How well does MPH perform in terms of transferability?

5.2 PERFORMANCE COMPARISON(RQ1)

We systematically compare our model with six existing representative methods which is depicted in Appendix A.5 on two real trajectory datasets, based on the overall performance metrics across three downstream tasks. Note that we

Table 1: Performance comparison on three downstream tasks on Chengdu dataset. **Bold** denotes the best result, and underline denotes the second-best result.

Models	Destination Prediction		Travel Time Estimation		Top k Similar Trajectory Search		
	Mi-F1↑	Ma-F1↑	MAE↓	RMSE↓	HR@1↑	HR@5↑	HR@10↑
Word2Vec	0.2260	0.0352	1.7322	2.7322	0.4814	0.5126	0.6074
Node2Vec	0.3167	0.0724	1.6056	2.0591	0.4503	0.5032	0.6370
Traj2Vec	0.0230	0.0100	2.0486	3.0303	0.4002	0.5217	0.6218
GVAE	0.3500	0.0900	2.0675	2.5747	0.4976	0.5301	0.6575
JCLRNT	0.4062	0.1459	1.5339	2.5626	0.7600	0.8860	0.8980
START	0.7993	0.2671	1.2861	1.6995	0.8806	0.9390	0.9698
JGRM	0.7538	0.2209	1.3842	1.8463	0.6993	0.8889	0.9342
MPH	**0.7996**	**0.2712**	**0.4386**	**0.6823**	**0.9096**	**0.9856**	**0.9928**

Table 2: Performance comparison on three downstream tasks on Xi'an dataset. **Bold** denotes the best result, and underline denotes the second-best result.

Models	Destination Prediction		Travel Time Estimation		Top k Similar Trajectory Search		
	Mi-F1↑	Ma-F1↑	MAE↓	RMSE↓	HR@1↑	HR@5↑	HR@10↑
Word2Vec	0.1180	0.0056	1.7098	2.7168	0.5776	0.7986	0.8756
Node2Vec	0.5426	0.1769	1.5111	1.9393	0.5500	0.7780	0.8660
Traj2Vec	0.0250	0.0208	2.0190	2.5129	0.0190	0.0712	0.1054
GVAE	0.3238	0.2120	2.1343	2.6969	0.5960	0.6574	0.7358
JCLRNT	0.3740	0.1264	1.6076	2.0710	0.7700	0.9300	0.9560
START	0.8229	0.3008	1.3494	1.7772	0.8598	0.9096	0.9396
JGRM	0.7748	0.3739	1.4567	1.9827	0.4911	0.7748	0.8596
MPH	**0.8367**	**0.2881**	**0.9111**	**1.2772**	**0.865**	**0.9728**	**0.9804**

introduce the exact same POI semantic information as MPH to the most representative baseline models (JGRM, START) to strip away the data advantage and purely evaluate the contribution of the model architecture itself.

Our model achieves superior performance across all three downstream tasks, demonstrating its effectiveness in capturing comprehensive trajectory semantics. In destination prediction, our model outperforms START and JGRM by effectively integrating movement patterns with road segment semantic information, enabling accurate identification of travel intentions and urban functional areas. For travel time estimation, our model exhibits a distinct advantage, particularly on the Chengdu dataset, by leveraging both inter-segment temporal correlations and intra-segment fine-grained dynamic features to precisely quantify time costs. In top-k similar trajectory search, our model surpasses START, which incorporates travel semantics but lacks understanding of travel purposes, by integrating road-level functional semantic information that captures deeper trajectory intentions. In contrast, sequence-based methods (Word2Vec, Node2Vec, Traj2Vec) struggle with complex spatiotemporal dependencies, while GVAE, though considering road network topology, lacks optimization for trajectory-specific characteristics and real-time dynamics.

5.3 ABLATION STUDY (RQ2)

In order to evaluate the effectiveness of each component of the model, we conducted ablation experiments with the following specific variants:

- **w/o Cont Loss(CL)**: Removes the contrastive learning in the self-supervised task.
- **w/o Mask Loss(ML)**: Removes the mask prediction in the self-supervised task.
- **w/o GPS Encoder(GE)**: Removes the GPS encoder branch.
- **w/o Route Encoder(RE)**: Removes the route encoder branch.
- **w/o Cross Attention(CA)**: Removes the cross-attention module and directly concatenates different view representations for fusion.

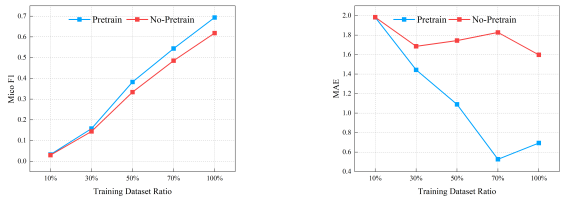
Table 3: MPH performance under different ablations on Chengdu dataset. **Bold** denotes the best result, and underline denotes the second-best result.

	Destination Prediction		Travel Time Estimation		Similar Trajectory Search		
	Mi-F1 ↑	Ma-F1 ↑	MAE ↓	RMSE ↓	HR@1 ↑	HR@5 ↑	HR@10 ↑
MPH	**0.7996**	**0.2712**	0.4386	0.6823	0.9096	0.9856	0.9928
w/o CL	0.7961	0.2269	1.7320	2.1294	0.9290	0.9750	0.9926
w/o ML	0.7376	0.1823	1.6157	2.1068	0.8098	0.8890	0.9514
w/o GE	0.7649	0.2042	1.5295	2.0711	**0.9510**	**0.9892**	**0.9938**
w/o RE	0.7121	0.1647	1.9045	2.3334	0.7978	0.9602	0.9784
w/o CA	0.7714	0.2118	**0.3188**	**0.5550**	0.8586	0.9530	0.9716

From Table 3, it can be seen that although some model variants perform better on specific tasks, the complete model demonstrates the best overall performance across all downstream tasks. This indicates that the effective collaboration between components is crucial for achieving balanced and robust model performance. 1) The **w/o CL** variant leads to a sharp decline in travel time estimation performance, as the two representations are not adequately aligned, resulting in the model's failure to establish an accurate time mapping relationship. 2) The **w/o ML** variant shows a decline in performance across all tasks, indicating that the mask prediction task is essential for the model's sequence understanding and generalization ability. 3) Notably, the **w/o GE** variant performs best on the most similar trajectory search task. This is attributed to the fact that this task prioritize the overall semantic similarity of trajectories; thus, removing GE allows the model to focus more on functional semantics, ultimately yielding better performance. 4) The **w/o RE** variant performs the worst on both the travel time estimation and most similar trajectory search tasks, highlighting the core role of path information in trajectory sequence modeling. 5)The **w/o CA** variant performs best on the travel time estimation task. Since this task is sensitive to GPS noise, the full cross-attention mechanism tends to introduce such noise and degrade performance; consequently, the modified variant proves to be more effective. These results suggest that while simplifying the model may work well in certain specific scenarios, the complete model achieves the best overall multi-task balance , confirming our design targets generalization, not single-task overfitting.

5.4 EFFECT OF PRE-TRAINING (RQ3)

We design comparative experiments to examine the effect of pretraining on downstream performance for destination prediction and travel time estimation. One group follows a "pretraining–fine-tuning" paradigm and the other trains the model from scratch. We progressively increase the training data scale from 10% to 100%. The results shown in Figure 3 implies that performance steadily improves for both approaches as data increases, while the pretrained model's curve rises more gently, indicating that pretraining enables the model to learn general features and reduces its dependence on data volume. Notably, fine-tuning a pretrained model consistently outperforms direct training at the same data scale, a benefit attributable to the rich trajectory priors learn during pretraining which enhance adaptability to downstream tasks. Moreover, as training data grows, overall performance continues to improve, demonstrating that our masked-segment prediction and contrastive self-supervised tasks effectively guide the model to capture generalizable trajectory representations; even with limited data, the pretrained model maintains strong performance, highlighting its practical value.



(a) Micro F1 in Destination Prediction.

(b) MAE in Travel Time Estimation.

Figure 3: The effect of pretrain on Chengdu dataset.

5.5 TRANSFERABILITY STUDY (RQ4)

We designed cross-dataset transfer experiments to evaluate the model's generalization ability and cross-city adaptability. Specifically, we will train a complete model on City A, and then directly or fine-tune it for transfer to City B, evaluating its performance on downstream tasks in the target city. The experiments include two transfer methods: 1) Direct Transfer: This approach directly applies the model trained on the source city to the target city. 2) Fine-tuning Transfer: In this approach, we select a subset of samples from the target city's training data and fine-tune the source city model before applying it to the downstream tasks of the target city. Considering the differences in road network structures between cities, during the fine-tuning process, we will unfreeze and reinitialize certain parameters, such as the GAT embedding dimensions of road segments, while retaining the weights of the other parts of the model to balance adaptation to the city-specific structures and the reuse of general knowledge.

Table 4: Experimental results for Chengdu→Xi'an transfer.

Chengdu→Xi'an	DP		TTE	
	Mi-F1 ↑	Ma-F1↑	MAE↓	RMSE↓
w/ Transfer	0.7673	0.2077	1.1018	1.5327
w/o Transfer	0.7635	0.2105	1.4938	1.8000

Table 5: Experimental results for Xi'an→Chengdu transfer.

Xi'an→Chengdu	DP		TTE	
	Mi-F1 ↑	Ma-F1 ↑	MAE ↓	RMSE ↓
w/ Transfer	0.7095	0.1457	0.5927	0.8571
w/o Transfer	0.7345	0.1655	1.1018	1.5327

From the above Tables 4 and 5, it can be seen that the direct transfer approach demonstrates significant advantages in both tasks, outperforming most baseline methods. Due to the differences in road networks and mobility patterns between cities, the fine-tuned results show task-related variability. Travel time estimation depends on movement patterns, which exhibit good transferability across cities of different scales. However, the destination prediction task heavily relies on city-specific semantic contexts, and its transfer performance is closely related to the similarity between cities. Overall, MPH demonstrates robust cross-city transferability and reduced reliance on city-dependent structures.

6 CONCLUSION

MPH consists of a hierarchical road segment-trajectory collaborative encoding strategy and a dual-branch heterogeneous feature encoding architecture. The former achieves progressive modeling from local to global by aligning and fusing road segment-level features and aggregating them at the trajectory level. The latter employs modality-specific encoders combined with a cross-attention mechanism to realize deep adaptive fusion of heterogeneous features, significantly enhancing the robustness and semantic discriminative capability of trajectory representations. Our model relies on high-density trajectory data and may degrade under sparse sampling. Future work will explore data generation and augmentation techniques to improve applicability in real-world sparse scenarios.

Acknowledgements

This work was partially supported by Tianjin Science and Technology Major Special Project 25ZXSFSN00070 and the National Key R&D Program of China 2023YFB2703900.

References

Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65: 126–139, 2017.

Yanchuan Chang, Jianzhong Qi, Yuxuan Liang, and Egemen Tanin. Contrastive trajectory similarity learning with dual-feature attention. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2933–2945. IEEE, 2023.

Bangchao Deng, Ling Ding, Lianhua Ji, Chunhua Chen, Xin Jing, Bingqing Qu, and Dingqi Yang. Marionette: Fine-grained conditional generative modeling of spatiotemporal human trajectory data beyond imitation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 463–473. ACM, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Tao-Yang Fu and Wang-Chien Lee. Trembr: Exploring road networks for trajectory representation learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–25, 2020.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM, 2016.

Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4): 12–18, 2008.

Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Li Song, Hui He, and Yu Zheng. What is the human mobility in a new city: Transfer mobility knowledge across cities. In *Proceedings of The Web Conference 2020*, pages 1355–1365. ACM, 2020.

Tobias Skovgaard Jepsen, Christian S Jensen, and Thomas Dyhre Nielsen. Relational fusion networks: Graph convolutional networks for road networks. *IEEE Transactions on Intelligent Transportation Systems*, 23 (1):418–429, 2020.

Jiawei Jiang, Dayan Pan, Houxing Ren, Xiaohan Jiang, Chao Li, and Jingyuan Wang. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 843–855. IEEE, 2023.

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. arXiv preprint arXiv:1907.05321, 2019.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. In *NIPS 2016 Workshop on Bayesian Deep Learning*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

Tian Lan, Ziyue Li, Zhishuai Li, Lei Bai, Man Li, Fugee Tsung, Wolfgang Ketter, Rui Zhao, and Chen Zhang. Mm-dag: Multi-task dag learning for multi-modal data-with application for traffic congestion analysis. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1188–1199. ACM, 2023.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pages 9694–9705, 2021.

Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S Jensen, and Wei Wei. Deep representation learning for trajectory similarity computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 617–628. IEEE, 2018.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400. IEEE, 2023.

Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4241–4248, 2021.

Xiang Liu, Xiaoying Tan, Yuchun Guo, Yishuai Chen, and Zhe Zhang. Cstrm: Contrastive self-supervised trajectory representation model for trajectory similarity computation. *Computer Communications*, 185:159–167, 2022.

Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. Online anomalous trajectory detection with deep generative sequence modeling. In *2020 IEEE 36th International*

Conference on Data Engineering (ICDE), pages 949–960. IEEE, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 352–361. ACM, 2009.

Zhipeng Ma, Zheyan Tu, Xinhai Chen, Yan Zhang, Deguo Xia, Guyue Zhou, Yilun Chen, Yu Zheng, and Jiangtao Gong. More than routing: Joint gps and route modeling for refine trajectory representation learning. In *Proceedings of the ACM Web Conference 2024*, pages 3064–3075. ACM, 2024.

Zhenyu Mao, Ziyue Li, Dedong Li, Lei Bai, and Rui Zhao. Jointly contrastive representation learning on road network and trajectory. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1501–1510. ACM, 2022.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 Workshop Track*, 2013.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710. ACM, 2014.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.

Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. Libcity: An open library for traffic prediction. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, pages 145–148. ACM, 2021.

Pu Wang, Jingya Sun, Wei Chen, and Lei Zhao. Towards effective urban region-of-interest demand modeling via graph representation learning. *Data Mining and Knowledge Discovery*, 38(6):3503–3530, 2024.

Ning Wu, Xin Wayne Zhao, Jingyuan Wang, and Dayan Pan. Learning effective road network representation with hierarchical graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 6–14. ACM, 2020.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, 2021.

Can Yang and Gyozo Gidofalvi. Fast map matching, an algorithm integrating hidden markov model with precomputation. *International Journal of Geographical Information Science*, 32(3):547–570, 2018.

Sean Bin Yang, Chenjuan Guo, Jilin Hu, Jian Tang, and Bin Yang. Unsupervised path representation learning with curriculum negative sampling. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021*, pages 3286–3292. IJCAI, 2021.

Sean Bin Yang, Jilin Hu, Chenjuan Guo, Bin Yang, and Christian S Jensen. Lightpath: Lightweight and scalable path representation learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2999–3010. ACM, 2023.

Di Yao, Chao Zhang, Zhihua Zhu, Jianhui Huang, and Jingping Bi. Trajectory clustering via deep representation learning. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3880–3887. IEEE, 2017.

From Moves to Paths: A Hierarchical Framework for Trajectory Representation Learning (Supplementary Material)

Chundong, Wang[1, 3] **Xiangtian, Zheng**[1] **Qingbo, Hao**[1,2] **Yongxin, Zhao**[1] **Yixuan, Song**[1] **Jia, Li**[1]

[1]School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China
[2]Technical College for the Deaf, Tianjin University of Technology, Tianjin, China
[3]Tianjin Police Institute, Tianjin, China

A EXPERIMENTAL SETTINGS

A.1 DATASETS

The experiments are conducted on two real-world datasets including Chengdu and Xi'an[1]. Each of these includes GPS trajectories, route trajectories. Both dataset contain 100000 trajectories collected in China, from 2018/01/01 to 2018/01/15. The road network data are downloaded from OpenStreetMap (OSM)Haklay and Weber [2008], Boeing [2017], and the POI data are collected via the Amap API[2]. We split each dataset into training, validation, and test sets in a 6:2:2 ratio.

A.2 IMPLEMENTATION

We run MPH using PyTorch 2.5.1. We uniformly deploy all experimental environments on the Ubuntu 22.04 system equipped with NVIDIA RTX GeForce 3090 GPU. The model settings are as follows: the GAT has 2 layers with attention heads [8, 1]; the Transformer encoder has 8 attention heads and a hidden layer dimension of 256; dropout rate is set to 0.1; the temperature parameter for contrastive loss δ is 0.07; λ_1=λ_2=0.5, and the mask ratio is 15%. During pretraining of the Chengdu and Xi'an models, we use the AdamW Loshchilov and Hutter [2019] optimizer with a learning rate of 3e-3 and train for 30 epochs.

A.3 POI AGGREGATION RADIUS CONFIGURATION

Table 6: POI aggregation radius for different road types

Road type	radius (m)	Road type	radius (m)
motorway	200	motorway_link	200
trunk	150	trunk_link	150
primary	100	primary_link	100
secondary	80	secondary_link	80
tertiary	60	tertiary_link	60
residential	50	unclassified	40
living_street	30		

To effectively capture the functional semantics surrounding each road segment, we assign differentiated buffer radii based on road hierarchy and service scope. Roads at different levels serve distinct functions within the urban transportation network

[1]Dataset:https://github.com/mamazi0131/JGRM
[2]POI Data:https://lbs.amap.com/

and exhibit varying influence ranges over surrounding Points of Interest (POI). In general, higher-grade roads with greater traffic capacity (e.g., motorways, urban arterials) have larger service radii and can cover broader areas, thus warranting larger buffer radii. In contrast, lower-grade roads (e.g., residential streets, living streets) primarily serve adjacent local areas, where smaller buffer radii are more appropriate.

The detailed radius configuration is presented in Table6. This design references urban road classification standards and relevant empirical studies, ensuring that POI aggregation adequately reflects the functional context of each road segment while avoiding the introduction of excessive irrelevant information.

A.4 BASELINES

To comprehensively evaluate the performance of MPH, we select four categories of representative TRL models for comparison and analysis:

1. Joint Representation Models
    - **JCLRNT Mao et al. [2022]**: This model constructs separate encoders for road segments and trajectory data and performs joint optimization using a triple contrastive loss.
    - **JGRM Ma et al. [2024]**: It designs dual encoders to process GPS trajectories and route trajectories, using mask reconstruction loss and cross-view matching loss to achieve representation learning.
2. Graph-Based Models
    - **START Jiang et al. [2023]**: This model uses a graph encoder to fuse the spatiotemporal features of trajectories and combines contrastive learning and mask language modeling tasks for pretraining.
    - **GVAE Kipf and Welling [2016]**: Based on the graph variational autoencoder architecture, this model learns node representations that align with road network topologies by reconstructing the adjacency matrix.
3. Trajectory Point-Based Sequence Models
    - **Traj2Vec Yao et al. [2017]**: Using a seq2seq architecture, this model encodes GPS trajectory sequences with an LSTM network to learn their dynamic features.
4. Route-Based Embedding Models
    - **Node2Vec Grover and Leskovec [2016]**: This model generates node sequences through random walks and learns road segment embeddings based on graph structures.
    - **Word2Vec Mikolov et al. [2013]**: Using the Skip-gram model, this approach learns distributed representations of road segments based on their co-occurrence relationships.

A.5 DOWNSTREAM TASKS

We designed the following three downstream tasks to evaluate the performance of the trajectory representation model:

- **Destination Prediction**: This task aims to predict the road segment ID of the destination based on the trajectory sequence, which is a classification task. We use a linear layer for prediction and employ micro-averaged (Mi-F1) and macro-averaged (Ma-F1) F1 scores as evaluation metrics, which measure the overall classification accuracy and the model's ability to identify sparse classes, respectively.
- **Travel Time Estimation**: In this task, the model is required to predict the total travel time from the start to the end of the trajectory, which is treated as a regression problem. We use a fully connected layer as the regression head and employ Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as evaluation metrics, assessing the accuracy and stability of the time predictions.
- **Top k Similar Trajectory Search**: This task evaluates the performance of trajectory representations in similarity-based retrieval. To rigorously test the discriminative power of the learned embeddings, we construct the candidate set with two types of samples: 1) original trajectories; 2) generated detour trajectories (spatially altered but semantically consistent with the original). We then retrieve the Top-k most similar samples and evaluate using Hit Rate (HR@1, HR@5, HR@10).

The detour trajectory generation follows a spatially-temporally feasible strategy:

1. Select a random continuous sub-trajectory from the original trajectory (accounting for r% of the original length);
2. Based on the directed road network graph, use the depth-first search (DFS) algorithm to find an alternative path between the start and end points of the selected sub-trajectory;
3. Assign timestamps to road segments in the detour path using historical average travel time or observed data, ensuring the detour sample meets both spatial feasibility (compliant with road network constraints) and temporal consistency (matching real travel dynamics).

B ADDITIONAL EXPERIMENTS

B.1 HYPERPARAMETER ANALYSIS

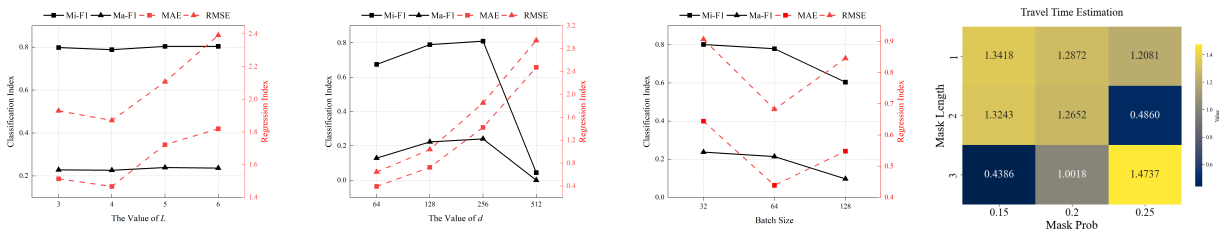
To investigate the impact of key model parameters on performance, this study systematically evaluates parameters such as the number of encoder layers, embedding dimensions, batch size, using the Chengdu dataset. Besides, mask length and mask probability, as key settings for the mask prediction task, have a significant impact on model training. We also test different combinations of mask probability and mask length. The experiments are evaluated using the Mi-F1 and Ma-F1 scores for the classification task, and the MAE and RMSE for the regression task. Retrieval metrics, which are part of the downstream extension validation, are not introduced at this stage. Experimental results are shown in Figure 4.

The number of encoder layers MPH achieves optimal regression performance at 4 layers, while classification metrics remain stable across settings in figure 4a. This is because the destination prediction task primarily relies on semantic information, while the travel time prediction task requires sufficient expressive capacity to capture long-term dependencies in the sequence. Once the number of encoder layers exceeds the optimal value, the performance of the regression task continuously declines on account of excessive model complexity.

Embedding dimension The results in Figure 4b indicate that a moderate embedding dimension (128) achieves the best trade-off between the two tasks. This is due to that smaller dimensions limit expressive power and semantic capture, hurting classification; larger dimensions introduce excessive parameters and task-irrelevant noise, causing severe overfitting—particularly detrimental to regression performance.

Batch size From the figure 4c, it can be seen that a batch size of 64 yields the best overall performance for both the classification and regression tasks. Smaller batches lead to underfitting and elevated regression errors due to insufficient gradient stability, whereas larger batches inject excessive noise, impairing classification accuracy. Therefore, moderate batch sizes thus best support stable convergence across both tasks.

Mask settings As shown in Figure 4d, the darker the grid color, the smaller the MAPE error value. We can observe that when the mask probability is 0.15 and the mask length is 3, the performance on the regression task is optimal. This is because a lower mask probability reduces information loss, while a longer mask length better aligns with the model's requirements for the regression task.



(a) The effect of the number of encoder layers on MPH.
(b) The effect of embedding size on MPH.
(c) The effect of batch size on MPH.
(d) The effect of mask settings on regression task.

Figure 4: Hyperparameter analysis on Chengdu dataset.

(a) query (b) Top-1 by MPH (c) Top-2 by MPH (d) Top-3 by MPH

(e) query (f) Top-1 by START (g) Top-2 by START (h) Top-3 by START

Figure 5: Comparison of the top-3 similar trajectories by MPH and START on Chengdu Dataset.

B.2 CASE STUDY

To visually evaluate MPH in the most similar trajectory search task, we compare a representative trajectory with the best-performing baseline, START, as shown in Figure 5. Both models retrieve trajectories largely similar to the query, the difference between the retrieved results and the query trajectory gradually increasing as the ranking moves lower. In the Top-1 result, the trajectory returned by MPH has the highest degree of spatial consistency with the query trajectory, while START shows some deviation. In Top-2 and Top-3, MPH's retrieved trajectories better match the path shape and spatial direction, while START exhibits noticeable deviations. This demonstrates that MPH captures global semantic and spatial features more effectively, yielding stronger discriminative ability in similarity retrieval.

B.3 COMPARISON OF DIFFERENT FUSION METHODS

To verify the rationality of the cross-attention fusion mechanism (treating motion features as Q and static route features as K-V), we supplement ablation experiments comparing three alternative fusion strategies: 1) Concatenation fusion(CF): directly concatenating GPS and route feature vectors; 2) Inverted cross-attention fusion(ICF), using static route features as Q and dynamic GPS motion features as K-V. Experimental results in Table 7 demonstrate our method's clear superiority in two downstream tasks. This empirically validates that treating dynamic GPS as queries to retrieve relevant static route context is the most effective strategy, aligning with our design principle inspired by human spatial cognition.

B.4 MODEL EFFICIENCY STUDY

The comparison of model parameters is shown in Table 8, MPH remains lightweight structure under a multi-module design while achieving superior efficient compared to baselines. Table 9 shows MPH achieves the lowest average inference latency across tasks, satisfying real-time ITS requirements and confirming practical deployability.

Table 7: Experimental results for the comparison of different fusion methods on chengdu dataset.

	Destination Prediction		Travel Time Estimation	
	Mi-F1 ↑	Ma-F1↑	MAE↓	RMSE↓
MPH	0.7996	0.2712	0.4386	0.6823
CF	0.7361	0.1771	1.5074	2.0015
ICF	0.7345	0.1832	1.5093	1.9304

Table 8: Comparison of Model Parameter Size and Training Time

Model Name	Parameter Size (MB)	Training Time (Hour)
MPH	3.05	1.4
START	15.28	2.3
JGRM	6.48	2

Table 9: Inference Latency Comparison

Model Name	Downstream Task	Inference Latency (ms per trajectory)
MPH	Destination Prediction	0.78
	Travel Time Estimation	0.54
	Similar Trajectory Search	0.57
JGRM	Destination Prediction	14.65
	Travel Time Estimation	2.72
	Similar Trajectory Search	11.4
START	Destination Prediction	0.89
	Travel Time Estimation	0.54
	Similar Trajectory Search	1.27

B.5 MODEL INTERPRETABILITY EVALUATION

To understand how MPH fuses dynamic GPS features with static route semantics, we visualize the cross-attention weights produced by the model during inference. Figure 6 visualizes the attention heatmap for a single trajectory, the darker the color, the greater the attention weight here. Attention concentrates around the diagonal, showing the model primarily focuses on the current and neighboring road segments, consistent with spatial locality. At the end of the trajectory, attention shifts rightward, indicating the model attends to future road segments, demonstrating temporal predictive capability.

Figure 7 compares three trajectories at different speeds: Figure 7a corresponds to a low-speed scenario, where the attention forms a narrow band tightly concentrated near the diagonal. Figure 7b represents normal medium-speed driving, where the attention appears as a wider band with moderate diffusion. Figure 7c shows a high-speed scenario, where the attention is more uniformly distributed without prominent peaks. This shows MPH dynamically adjusts attention to static route semantics based on GPS motion signals, highlighting cross-attention's adaptive capability in fusing heterogeneous features.

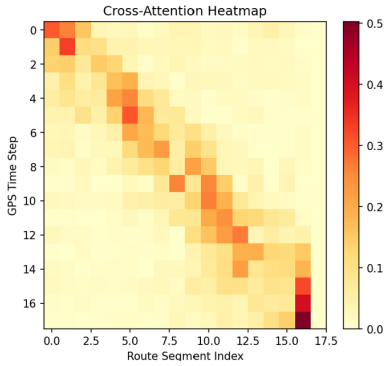
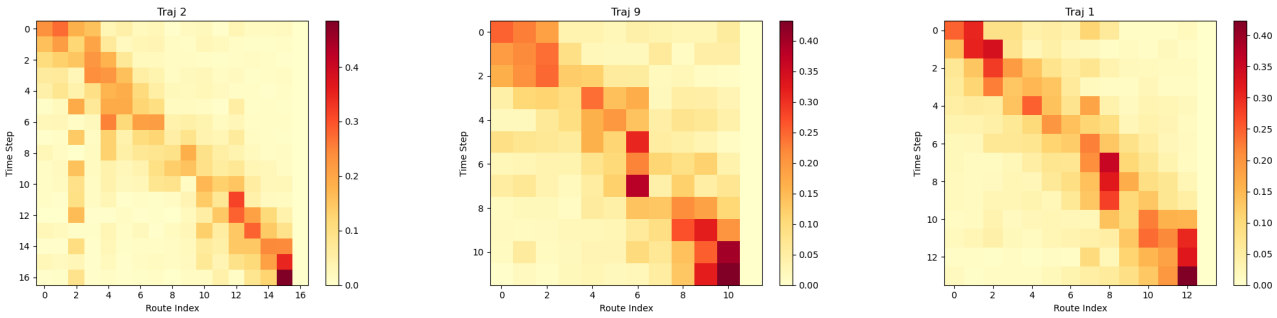


Figure 6: Attention heatmap of a sampled trajectory



(a) Low speed trajectory (b) Medium speed trajectory (c) High speed trajectory

Figure 7: Attention heatmaps of different speed trajectories