Self-Supervised Cross-City Trajectory Representation Learning Based on Meta-Learning

Yanwei Yu[1,2*], Hong Xia[1], Shaoxuan Gu[1], Xingyu Zhao[1], Dongliang Chen[1], Yuan Cao[1]

[1]Faculty of Information Science and Engineering, Ocean University of China, China
[2]State Key Laboratory of Physical Oceanography, Ocean University of China, China
{yuyanwei}@ouc.edu.cn, {xhong, gushaoxuan, zhaoxingyu}@stu.ouc.edu.cn, {chendongliang, cy8661}@ouc.edu.cn

Abstract

Trajectory representation learning transforms complex spatio-temporal features of trajectories into dense, low-dimensional embeddings, enabling applications in intelligent transportation systems. With advances in this field and the availability of large-scale traffic data, intelligent urban systems have been widely deployed in major cities. However, existing methods heavily rely on large volumes of trajectory data, limiting their transferability to cities with sparse data, especially small or less-developed ones. Moreover, most current approaches learn representations within a single city, overlooking the shared travel patterns across regions and cities with similar geographic contexts. To address these issues, we propose *MetaTRL*, a self-supervised cross-city trajectory representation learning method based on meta-learning. Specifically, we introduce a Shared and Private Parameterized Cross-city Meta-learning Framework to support knowledge sharing and transfer across cities. We further design a Meta-knowledge Enhanced Road Segment Encoder and a Trajectory Encoder that integrates private and shared knowledge to learn and fuse spatio-temporal trajectory features. Extensive experiments on two real-world datasets and multiple downstream tasks demonstrate the significant superiority of MetaTRL over state-of-the-art baselines and achieves a remarkable average improvement of 134.66% in Macro-F1 on destination prediction task.

Introduction

With the rise of GPS-enabled devices and location-based services, large volumes of trajectory data are being generated, providing valuable insights into human and object mobility. As a key form of spatio-temporal data, trajectories support applications in urban planning (Bao et al. 2017; Dai et al. 2022; Alessandretti 2022), intelligent transportation (Feng et al. 2023; Liang et al. 2022)

Trajectory Representation Learning (TRL) aims to encode sparse, high-dimensional trajectories into compact and meaningful embeddings for diverse downstream tasks. Early TRL methods used task-specific sequence models trained end-to-end (Liu et al. 2019; Li et al. 2018), but they suffered from limited scalability, strong task dependency, and heavy reliance on labeled data (Ashukha et al. 2020; Ishida et al. 2020). To address these limitations, self-supervised TRL has gained popularity. Seq2seq-based models with reconstruction losses were first introduced (Fang et al. 2021; Li et al. 2018; Yao et al. 2017). To reduce redundancy and noise, some methods represent trajectories as sequences of road segments rather than raw GPS points (Chen et al. 2021; Yang et al. 2021; Jiang et al. 2023b). More recent two-stage approaches leverage graph neural networks to first learn road-segment embeddings from the road network, followed by trajectory embedding learning via sequence models with self-supervised objectives (Jiang et al. 2023b; Ma et al. 2024; Xia et al. 2025)

Nevertheless, there are still two main challenges to be solved in self-supervised TRL. (1) **Poor generalization and transferability in cross-city learning.** Current intelligent transportation systems have been extensively applied in large cities but seldom in smaller ones. One key reason is that large cities have the capacity to collect massive amounts of relevant data, while smaller cities often cannot afford the associated high costs. Unfortunately, most existing TRL methods generally adopt self-supervised learning, which effectively eliminates the need for labeled data, they typically rely on large amounts of trajectory data for training in order to fit the specific spatio-temporal features of trajectories. Furthermore, different cities usually have no direct interactions and often exhibit distinct geographic structures and travel preferences, so, current methods inherently exhibit poor generalization and limited transferability in cross-city learning. Studying how to leverage existing data from certain cities to effectively learn consistent spatio-temporal features across cities and achieve knowledge transfer is both a meaningful and challenging problem.

(2) **Underexplored correlations between travel patterns and road segment geographic context.** The road network fundamentally constrains human mobility, and a trajectory's spatial features are reflected in the relationships among road segments. These relationships depend not only on intrinsic segment attributes (e.g., length, road class) but also on their direct connections (e.g., connectivity), which most existing studies focus on. However, the features of a road segment are influenced not only by its directly connected neighbors but also by the geographic context of its surrounding region. For instance, congestion at an intersection is affected by the attributes of adjacent seg-

*Corresponding author: Yanwei Yu.

ments, the intersection's structure, and even nearby intersections. Road segments with similar geographic contexts often exhibit similar spatio-temporal correlations. Prior work overlooks these contextual commonalities of road segments across regions. More importantly, although road networks differ across cities, such correlations between geographic context and trajectory spatio-temporal features are consistent both across regions within a city and across different cities. Therefore, it is meaningful and necessary to fully explore these correlations to better support TRL.

To address the above challenges, we propose MetaTRL, a cross-city TRL method based on meta-learning, key contributions of this paper are summarized as follows:

- We propose a novel cross-city self-supervised trajectory representation learning framework. By a meta-learning process with both private and shared parameters, guided by a dual-modal trajectory recovery task, our MetaTRL facilitates effective cross-city spatio-temporal feature learning and knowledge transfer.
- We design a meta-knowledge-enhanced road-segment encoder and a transerferable temporal encoder to improve private and shared spatio-temporal features learning. Our proposed knowledge fusion trajectory encoder performs effective feature integration.
- The experiments conducted on two real-world datasets across three downstream tasks verify that our model exhibits strong capabilities in transferring spatio-temporal features across cities and achieves the best performance.

Related Work

Self-supervised Trajectory Representation Learning

Trajectory representation learning aims to encode raw trajectories into high-dimensional embeddings for downstream tasks such as classification and prediction. Early work treats GPS trajectories as sequential data (Fang et al. 2021) (Li et al. 2018; Yao et al. 2017, 2022), ignoring road network structure. Recent methods adopt a two-stage approach: first, learning road segment embeddings via skip-gram (Mikolov et al. 2013) or GNNs (Mao et al. 2022; Fu and Lee 2020; Yang et al. 2023; Wei et al. 2024; Jiang et al. 2023b; Xia et al. 2025); second, encoding trajectories with sequence models like RNNs or Transformers (Vaswani et al. 2017). For temporal features, existing methods mainly focus on encoding temporal periodicity and time intervals. In addition, START (Jiang et al. 2023b) incorporates temporal information into the attention mechanism, while TrajRL (Xia et al. 2025) performs multi-scale interval encoding. Recently, more multimodal information is incorporated into TRL(Ma et al. 2024; Wei et al. 2025).

To reduce reliance on labeled data, self-supervised learning is widely adopted. Typical techniques include sequence reconstruction (Yao et al. 2017; Li et al. 2018), masked language modeling (Chen et al. 2021), mutual information maximization (Yang et al. 2021), and contrastive learning (Chen et al. 2020; Chang et al. 2023). These approaches enhance generalization and robustness of trajectory embeddings across tasks.

Cross-city Transfer Learning

Knowledge transfer (Jin et al. 2022; Wei, Zheng, and Yang 2016; Finn, Abbeel, and Levine 2017) aims to address deep learning challenges in data-scarce scenarios. In intelligent urban system (Zou et al. 2025; Jin, Chen, and Yang 2022), achieving cross-city knowledge transfer (Lu et al. 2022) to reduce data costs and improve learning efficiency remains an ongoing research challenge. In the field of spatio-temporal prediction (Panagopoulos, Nikolentzos, and Vazirgiannis 2021; Wang et al. 2018; Gong et al. 2024), some methods have explored knowledge transfer within homogeneous domains. However, the cross-city TRL task requires learning shared spatio-temporal features from cities with significant heterogeneity in a self-supervised manner, making it a more challenging task.

Preliminaries

In this section, we first introduce the basic preliminaries used in this paper, and then we formally define the studied problem.

Definition 1 (Road Network) *A road network is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where $\mathcal{V}$ is the set of vertices, each vertex $v_i \in \mathcal{V}$ representing a road segment. $\mathcal{E}$ is the set of edges, each $e_{i,j} = (v_i, v_j)$ representing the intersection between road segments v_i and v_j. $\mathcal{F} \in \mathbb{R}^{|\mathcal{V}| \times f}$ is the feature matrix of road segments, where f is the feature dimension.*

Definition 2 (Road-network Constrained Trajectory) *A road-network constrained trajectory $\mathcal{T} =< r_1, r_2, \ldots, r_n >$ is a sequence of adjacent road segments ordered in time, where $r_i = \langle v_i, t_i \rangle$ and $v_i \in \mathcal{V}$ presents the i-th road segment in the sequence, t_i is the visit time for v_i, and $t_i < t_{i+1}$. A road-network constrained trajectory is usually obtained by performing network matching on the original trajectory.*

In this work, we focus on the road-network constrained trajectories. For simplicity, we use road and trajectory to refer to road segment and road-network constrained trajectory, respectively.

Next, we formally define our studied problem of cross-city trajectory representation learning as follows:

Problem 1 *Given the road networks $\mathcal{S}_\mathcal{G} = \{\mathcal{G}_i\}_{i=1}^{|\mathcal{S}_\mathcal{G}|}$ and datasets of trajectories $\mathcal{S}_\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{|\mathcal{S}_\mathcal{D}|}$, of multiple cities, where $\mathcal{D}_i = \{\mathcal{T}_{i,j}\}_{j=1}^{|\mathcal{D}_i|}$ is the dataset of trajectories of i-th city. Based on data from multiple cities, the cross-city trajectory representation learning aims to learn a generic low-dimensional vector $h_j \in \mathbb{R}^d$ that preserves the spatio-temporal correlations for each trajectory $\mathcal{T}_{t,j} \in \mathcal{D}_i$ of the specific target city.*

In this work, our goal is to achieve cross-city knowledge transfer for TRL, thereby obtaining robust and generalizable trajectory representation vector that achieves optimal performance on multiple downstream tasks.

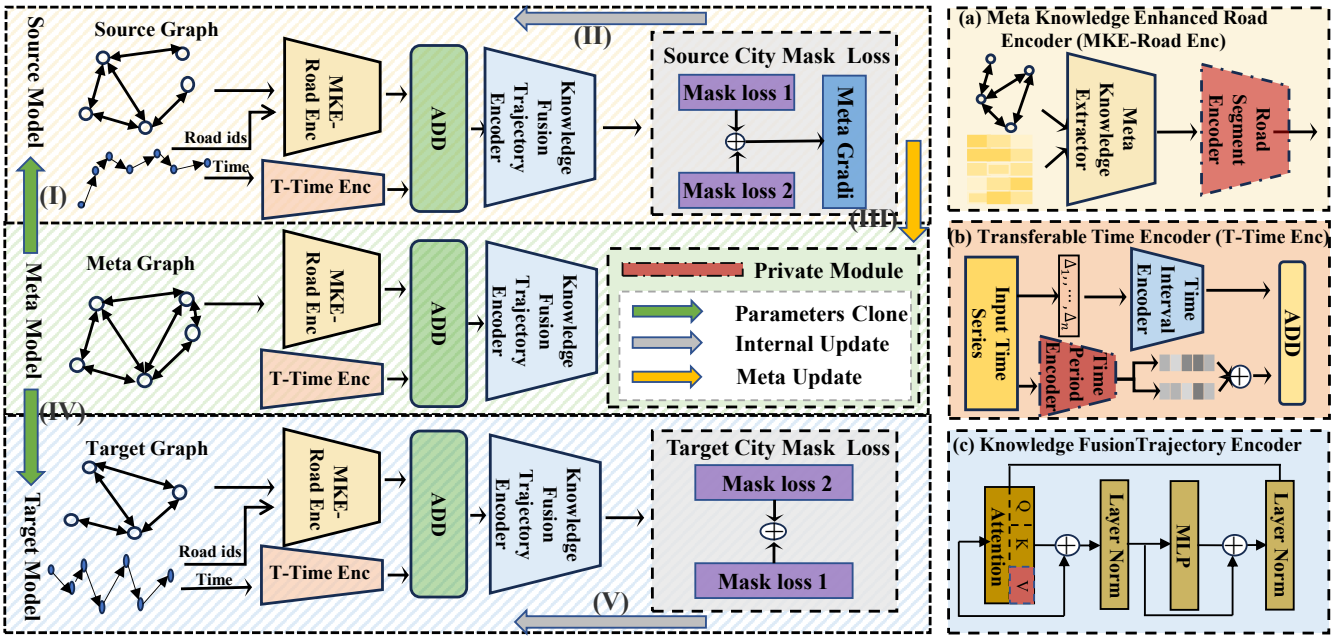


Figure 1: Overall Framework of the Proposed MetaTRL.

Methodology

In this section, we present the details of our proposed MetaTRL framework, as illustrated in Figure 1. Our proposed MetaTRL consists of four key components: (1) *Cross-city Meta-learning Framework*, (2) *Meta Knowledge Enhanced Road Encoder*, (3) *Transferable Temporal Encoder*(4) *Knowledge Fusion Trajectory Encoder*.

Cross-city Meta-learning Framework

To enable cross-city knowledge transfer, our MetaTRL adopts a meta-learning framework. The dataset is split into source and target cities, each with its own training and test sets. Unlike classical meta-learning that shares all parameters, we introduce a shared-private parameter mechanism to better capture both general and city-specific spatio-temporal features. Shared parameters are updated using the source model and transferred to initialize the target model, facilitating general feature transfer. Private parameters are updated separately within each model.

We now introduce the training process of the private and shared parameterized meta-learning framework. Let $M_{\Theta_{\text{src}}}$, $M_{\Theta_{\text{meta}}}$,$M_{\Theta_{\text{tar}}}$ be the source, meta and target models respectively, where Θ_{src}, Θ_{meta} and Θ_{tar} are their parameters. Firstly, the shared parameters of source model are initialized from the current meta model, which is named Parameters Clone. Subsequently, the source model performs internal learning through the self-supervised task:

$$\Theta_{\text{src}} = \Theta_{\text{src}} - \alpha_{\text{src}} \nabla_{\Theta_{\text{src}}} \mathcal{L}^{pre}(M_{\Theta_{\text{src}}}(\mathcal{D}^{src}_{\text{train}})), \quad (1)$$

where α_{src} is the learning rate for the source model, $\mathcal{L}^{pre}$ is the loss function and $\mathcal{D}^{\text{src}}_{\text{train}}$ is the source train set. After training, the source model captures the spatio-temporal features in the source city. Subsequently, the meta model updates its parameters based on the gradients of the source model on the source test set, enabling the meta model to rapidly learn knowledge from the source city.

$$\Theta_{\text{meta}} = \Theta_{\text{meta}} - \alpha_{\text{meta}} \nabla_{\Theta_{\text{src}}} \mathcal{L}^{pre}(M_{\Theta_{\text{src}}}(\mathcal{D}^{\text{src}}_{\text{test}})), \quad (2)$$

where where α_{meta} is the learning rate for the meta model and $\mathcal{D}^{\text{src}}_{\text{test}}$ is the test set of the source dataset. Subsequently, the target city clones the shared parameters from the meta-model, enabling effective cross-city knowledge transfer.

$$\Theta^{\text{shared}}_{\text{tar}} = \{\theta \mid \forall \theta \in \Theta^{\text{shared}}_{\text{meta}}\}. \quad (3)$$

Through the above process, the target model acquires useful prior knowledge and a better initialization for training. Finally, the target model undergoes self-supervised pretraining on the training set of the target city.

$$\Theta_{\text{tar}} = \Theta_{\text{tar}} - \alpha_{\text{tar}} \nabla_{\Theta_{\text{tar}}} \mathcal{L}^{pre}(M_{\Theta_{\text{tar}}}(\mathcal{D}^{\text{tar}}_{\text{train}})), \quad (4)$$

where α_{tar} is the learning rate for the target model and $\mathcal{D}^{\text{tar}}_{\text{train}}$ is the train set of the target dataset.

Meta-knowledge Enhanced Road Encoder

Meta knowledge extraction. To fully exploit the correlations between similar road geographic context and travel patterns across cities, thereby enabling effective knowledge transfer, we design a shared road-network meta-knowledge extraction module.

Considering the transferability across cities, we try to extract the intersection structure of the road network, which captures the mutual influences between roads as well as their joint influence on trajectories. The consistency of intersection structures across and within cities reflects shared spatial patterns, which we regard as meta-knowledge in road network. The following process is performed for each road, with subscripts omitted for clarity.

To characterize the intersection structure, we construct a structural feature vector for each road. Specifically, we first compute the distances among all its direct neighbors, and then construct a statistical feature vector $s_d \in \mathbb{R}^4$ over these distances, including the maximum, minimum, mean, and variance. In addition, the shape of a road is also part of the intersection structure. We describe the road shape by relative spatial position which is a vector $s_p = [\delta_s, \delta_e] \in \mathbb{R}^2$ representing the difference in latitude and longitude between its start and end points. The structural feature vector is constructed by combining all the above features as:

$$\mathbf{s} = concat(s_d, s_p) \in \mathbb{R}^6, \tag{5}$$

We employ a shared MLP to further encode and learn the meta knowledge,

$$\mathrm{MKF(s)} = \mathrm{MLP}^{shared}(s), \tag{6}$$

where MLP^{shared} is an MLP whose parameters are shared.

Private road encoder. In addition to the meta-knowledge of roads mentioned above, each road also possesses its own intrinsic attributes. Considering both types of features and the dynamic correlations among roads, we design a road encoder based on a graph attention network (GAT). In the following description of the graph neural network, all parameters are private, and we omit the corresponding notation for brevity. First, we employ a private MLP to encode and learn the attribute features of each road:

$$\mathrm{NAF}(u_i) = \mathrm{MLP}^{private}(u_i), \tag{7}$$

where u is the attribute vector of a road.

Then, the roads of each city perform message passing over their corresponding road-network graph $\mathcal{G}$, during which the meta-knowledge of roads is incorporated. The nodes in the graph G aggregate information from their neighboring nodes as:

$$h_i^{(l+1)} = \mathrm{ELU}\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l)} h_j^{(l)} \mathbf{W}_1^{(l)}\right), \tag{8}$$

where $h_i^{(0)} = \mathrm{NAF}(u_i)$, $\mathbf{W}_1^{(l)} \in \mathbb{R}^{d \times d}$ is learnable private parameters for the l-th layer, $\mathcal{N}_i$ is the set of neighbors of v_i in graph $\mathcal{G}$, ELU is the Exponential Linear Unit activation function, and $\alpha_{ij}^{(l)}$ is the attention weight at the l-th layer.

The attention score is related to both the hidden states of node i and node j and meta knowledge of nodes learned. In addition, inspired by previous work, we also incorporate the transition probabilities of roads derived from historical data. First, we obtain the meta knowledge on $e_{i,j}$ as follows:

$$\mathrm{MKF}_{(ij)} = \mathrm{MKF}(v_i) \parallel \mathrm{MKF}(s_j), \tag{9}$$

where $\parallel$ is the concatenation operator. The attention weight calculation for the l-th layer is as follows:

$$\begin{aligned}
\alpha_{ij}^{(l)} &= \frac{\exp(\mathrm{LeakyReLU}(e_{ij}^{(l)}))}{\sum_{k \in \mathcal{N}_i} \exp(\mathrm{LeakyReLU}(e_{ik}^{(l)}))}, \\
e_{ij}^{(l)} &= ((h_i^{(l)} \parallel h_j^{(l)}) \mathbf{W}_{i,j}^{(l)} + p_{ij} \mathbf{W}_3^{(l)}) \mathbf{W}_4^{(l)\mathsf{T}}, \\
\mathbf{W}_{i,j}^{(l)} &= \mathrm{MKF}_{(i,j)} \mathbf{W}_2^{(l)}, \\
p_{i,j} &= \mathrm{count}(e_{i,j}) / \mathrm{count}(v_i),
\end{aligned} \tag{10}$$

where $h_i^{(l)}, h_j^{(l)} \in \mathbb{R}^{d_l}$ are the embeddings of roads v_i and v_j in the l-th layer, $\mathbf{W}_2^{(l)}, \in \mathbb{R}^{1 \times d}$, $\mathbf{W}_3^{(l)}, \mathbf{W}_4^{(l)} \in \mathbb{R}^{1 \times d}$ are learnable parameters, d is the hidden vector dimension in GAT, LeakyReLU is the activation function, and $p_{i,j}$ is the transition probability between v_i and v_j, $\mathbf{W}_{i,j}^{(l)} \in \mathbb{R}^{2d \times d}$ is a learned weight matrix that incorporates meta knowledge of connected roads.

Transferable Temporal Encoder

The temporal features in a trajectory are as important as the spatial features. To learn similar temporal patterns across trajectories from different cities and enable knowledge transfer in the temporal domain, we design a shared time interval encoder beyond the private period encoder.

To extract weekly and daily periodicities, we extract the day of the week and the minute of the day, and transform them into the day-of-week index (1 to 7) and minute-of-day index (1 to 1440), respectively. For each visit timestamp t_i, we use two embeddings $t_i^d \in \mathbb{R}^d$ and $t_i^m \in \mathbb{R}^d$ to present these two periodic patterns. The time period embedding $T_{t_i}^P$ of time t_i is:

$$T_{t_i}^P = t_i^d + t_i^m. \tag{11}$$

More importantly, the shared time interval encoder is utilized to capture temporal variation patterns of trajectories that are transferable across cities. For Δ_i in the time interval series of a trajectory

$$\Delta_i = \begin{cases} 0 & \text{if } i = 1 \\ t_i - t_{i-1} & \text{if } i > 1 \end{cases}, \tag{12}$$

we transform it to a dense embedding to achieve a stronger representation capability. Specifically, following (Ma et al. 2024), we adopt a learnable embedding matrix consisting of N_B embedding vectors, each regarded as a virtual bucket. The time interval embedding for Δ_i is then constructed as a weighted combination of these N_B virtual buckets.

$$T_{t_i}^I = \mathrm{Softmax}(\mathrm{MLP}^{shared}(\Delta_i)) * W_{BE}, \tag{13}$$

where $W_{BE} \in \mathbb{R}^{N_B \times d}$ is learnable embedding matrix. After obtaining the time period embedding and time interval embedding, we add the two as the time embedding of time t_i:

$$T_{t_i} = T_{t_i}^P + T_{t_i}^I. \tag{14}$$

Knowledge Fusion Trajectory Encoder

After the spatial and temporal encodings, we obtain the embedding of a road v_i in a given trajectory $\mathcal{T}$ as:

$$x_i = h_i + T_{t_i} + pe_i, \tag{15}$$

where pe_i denotes the position encoding in $\mathcal{T}$. The initial representation of the trajectory $\mathcal{T}$ is obtained by concatenating the embeddings of roads in it as $\mathrm{X} = x_1 \| x_2 \| \ldots \| x_{|\mathcal{T}|} \in \mathbb{R}^{|\mathcal{T}| \times d}$.

Now, each road embedding integrates both private and shared spatio-temporal features. To build a generalized trajectory representation for the target city, the trajectory encoder must be capable of extracting knowledge that truly

align with the features of trajectories in the target city. We achieve this by leveraging the self-attention mechanism in the Transformer encoder. First, to distinguish between private knowledge and shared knowledge, we map trajectory embedding into two separate views:

$$\begin{aligned} \mathrm{X}^{private} &= \mathrm{MLP}^{private}(\mathrm{X}), \\ \mathrm{X}^{shared} &= \mathrm{MLP}^{shared}(\mathrm{X}). \end{aligned} \tag{16}$$

Considering that the Value in the attention module should contain the true features of the target city, while the Query and Key should capture general features of different cities and enable knowledge transfer through attention weights, we obtain them from the corresponding views.

$$\begin{aligned} \mathrm{Q} &= \mathrm{X}^{shared} W_Q^{shared}, \\ \mathrm{K} &= \mathrm{X}^{shared} W_K^{shared}, \\ \mathrm{V} &= \mathrm{X}^{private} W_V^{private}, \end{aligned} \tag{17}$$

where W_Q^{shared}, W_K^{shared} and W_V^{shared} are shaped learnable parameters.Then, self-attention–based feature fusion is performed as follows:

$$\mathrm{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{QK}^T}{\sqrt{d}}\right)\mathrm{V}. \tag{18}$$

Other components, such as residual connections, layer normalization, and the multi-head self-attention mechanism are also employed, but are not elaborated here. We denote the output of the trajectory encoder as $\mathbf{Z} \in \mathbb{R}^{|\mathcal{T}| \times d}$. Moreover, the final trajectory representation is obtained through average pooling as $z \in \mathbb{R}^d$.

Self-supervised Pre-training and Fine-Tuning

Pre-training. Our goal is to learn generalized trajectory representations that can support various downstream tasks. To this end, we pre-train MetaTRL using self-supervised signals from the trajectory itself, rather than relying on supervised labels. The effectiveness of trajectory recovery tasks based on masked language modeling (MLM) (Devlin et al. 2018) has been widely demonstrated. To enhance cross-city knowledge transfer, we design a dual-modal masking mechanism, enabling the model to learn and transfer both spatial and temporal features.

Specifically, we first perform full masking on randomly selected, non-contiguous sub-trajectories rather than on individual roads, masking all their features to increase prediction difficulty. Then, we apply temporal masking, where a subset of unmasked road segments is randomly selected, and their temporal features are masked with a fixed probability.

Finally, during the meta-learning pre-training phase, we append a linear predictor after the trajectory encoder to separately predict the masked roads and the masked timestamps.

$$\begin{aligned} \hat{\mathbf{Z}}^{fm} &= \mathbf{Z}\mathbf{W}_{fm} + b_{fm} \in \mathbb{R}^{\mathcal{T} \times |\mathcal{V}|}, \\ \hat{t}^{tm} &= \mathbf{Z}\mathbf{W}_{tm} + b_{tm} \in \mathbb{R}^{\mathcal{T} \times 1}, \end{aligned} \tag{19}$$

The former is a classification task and the latter a regression task. We use cross-entropy and mean squared error as their respective losses, combined via weighted summation.

$$\begin{aligned} \mathcal{L}_{\mathcal{T}}^{\mathrm{fm}} &= -\frac{1}{|\mathcal{FM}|} \sum_{i=1}^{|\mathcal{FM}|} \log \frac{\exp(\hat{\mathbf{Z}}_i^{fm})}{\sum_{j=1}^{|\mathcal{V}|} \exp(\hat{\mathbf{Z}}_j^{fm})}, \\ \mathcal{L}^{\mathrm{tm}} &= \frac{1}{\mathcal{TM}} \sum_{i=1}^{\mathcal{TM}} \|\hat{t}_i^{tm} - t_i\|^2, \\ \mathcal{L}^{pre} &= \gamma \mathcal{L}_{\mathcal{T}}^{\mathrm{fm}} + (1-\gamma)\mathcal{L}^{\mathrm{tm}}. \end{aligned} \tag{20}$$

Model Fine-tuning. We fine-tune MetaTRL on downstream tasks, formulating them as either classification or regression. A linear classifier or regressor followed target model is trained with cross-entropy or mean squared error loss, respectively.

Dataset	Porto	BJ
Time span	07/01/2013-07/01/2014	11/01/2015-11/30/2015
#Trajectories	695,085	1,018,312
#Users	435	1,677
#Road Segments	10,903	38,479

Table 1: Statistics of the two datasets after preprocessing.

Datasets

We conduct the experiments on two real-world datasets, Porto and BJ. All road-network and road attribute information is obtained from OpenStreetMap.

Compared Methods

To evaluate the effectiveness of our model, we compare MetaTRL with SOTA TRL methods, including **Trembr** (Fu and Lee 2020), **PIM** (Yang et al. 2021), **Toast** (Chen et al. 2021), **JCLRNT** (Mao et al. 2022), **LightPath** (Yang et al. 2023), **TrajCL** (Chang et al. 2023), **START** (Jiang et al. 2023b), **TrajRL** (Xia et al. 2025).

Experiment Settings

We implement our MetaTRL with PyTorch 2.6 and Libcity (Jiang et al. 2023a) framework. During training, another dataset outside the target city is used as the source dataset. To better align with the cross-city TRL application scenario, we limit the target training set to 50,000 trajectories. The code is available at https://github.com/Xfc30/MetaTRL.

Downstream task settings. We select three popular and important downstream tasks to evaluate the model's performance: trajectory classification (CLA), travel time estimation (TTE), and destination prediction (DP).

Evaluation Metrics. For the CLA task, we use Accuracy (ACC), F1-score (F1), Area Under Roc (AUC) to evaluate binary classification task, and Micro-F1, Macro-F1, Recall@5 to evaluate multi-classification task. For the TTE task, we use Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) to evaluate the regression results. For the DP task, we use Micro-F1, Macro-F1, Recall@5 to evaluate the multi-classification task.

Method	Trajectory Classification			Travel Time Estimation			Destination Prediction		
	Micro-F1 ↑	Macro-F1 ↑	Recall@5 ↑	MAE ↓	MAPE ↓	RMSE↓	Micro-F1 ↑	Macro-F1 ↑	Recall@5 ↑
Trembr [TIST-20]	0.0374	0.0167	0.1143	1.707	0.265	2.354	0.515	0.166	0.770
PIM [IJCAL-21]	0.0295	0.0121	0.0974	1.905	0.272	2.872	0.370	0.129	0.606
Toast [CIKM-21]	0.0308	0.0113	0.1070	1.995	0.288	2.923	0.390	0.131	0.637
JCLRNT [CIKM-22]	0.0355	0.0155	0.1016	2.014	0.265	2.761	0.360	0.119	0.592
TrajCL [ICDE-23]	0.0279	0.0841	0.0924	2.087	0.284	3.089	0.494	0.160	0.732
LightPath [KDD-23]	0.0324	0.0154	0.1069	1.802	0.259	2.784	0.417	0.139	0.657
START [ICDE-23]	0.0403	0.0194	0.1334	1.605	0.251	2.351	0.550	0.176	0.800
TrajRL [ICDE-25]	0.0425	0.0202	0.1466	1.607	0.241	2.368	0.563	0.191	0.818
MetaTRL	**0.0478**	**0.0252**	**0.1545**	**1.509**	**0.233**	**2.198**	**0.642**	**0.248**	**0.861**
Improvement	12.47%	24.75%	5.39%	5.98%	3.32%	6.51%	14.03%	29.84%	5.26%

The metric with "↑" means that a larger result is better, and the metric "↓" means that a smaller result is better.
The **bold** results are the best, and the underlined results are the second best. The time in travel time estimation is in minutes.

Table 2: Performance Comparison of All Models on Porto Dataset.

Method	Trajectory Classification			Travel Time Estimation			Destination Prediction		
	ACC ↑	F1 ↑	AUC ↑	MAE ↓	MAPE ↓	RMSE↓	Micro-F1 ↑	Macro-F1 ↑	Recall@5 ↑
Trembr [TIST-20]	0.685	0.735	0.715	11.539	0.516	38.363	0.131	0.027	0.281
PIM [IJCAL-21]	0.650	0.713	0.724	13.489	0.581	42.365	0.094	0.021	0.226
Toast [CIKM-21]	0.672	0.690	0.681	12.875	0.529	49.151	0.119	0.024	0.233
JCLRNT [CIKM-22]	0.709	0.725	0.738	12.616	0.533	48.650	0.107	0.020	0.215
LightPath [KDD-23]	0.649	0.690	0.690	12.153	0.605	45.006	0.108	0.023	0.245
TrajCL [ICDE-23]	0.584	0.608	0.623	14.110	0.592	50.714	0.121	0.027	0.269
START [ICDE-23]	0.737	0.819	0.784	10.851	0.489	36.317	0.140	0.029	0.295
TrajRL [ICDE-25]	0.732	0.816	0.781	10.698	0.468	36.185	0.173	0.038	0.343
MetaTRL	**0.784**	**0.847**	**0.845**	**10.217**	**0.400**	**34.279**	**0.396**	**0.129**	**0.561**
Improvement	6.38%	3.42%	7.78%	4.50%	14.53%	5.27%	128.90%	239.47%	63.56%

The symbols have the same meanings as in Table 2.

Table 3: Performance Comparison of All Models on BJ Dataset.

Setting	Method	Trajectory Classification			Travel Time Estimation			Destination Prediction		
		Micro-F1 ↑	Macro-F1 ↑	Recall@5 ↑	MAE ↓	MAPE ↓	RMSE↓	Micro-F1 ↑	Macro-F1 ↑	Recall@5 ↑
Porto	START	0.0403	0.0194	0.1334	1.605	0.251	2.351	0.550	0.176	0.800
	TrajRL	0.0425	0.0202	0.1466	1.607	0.241	2.368	0.563	0.191	0.818
	MetaTRL	**0.0433**	**0.0211**	**0.1493**	**1.578**	**0.238**	**2.309**	**0.594**	**0.217**	**0.835**
BJ→ Porto	STATR	0.0352	0.0157	0.1244	1.615	0.259	2.364	0.509	0.150	0.758
	TrajRL	0.0386	0.0173	0.1385	1.654	0.267	2.401	0.535	0.183	0.779
	MetaTRL	**0.0478**	**0.0252**	**0.1545**	**1.509**	**0.233**	**2.198**	**0.642**	**0.248**	**0.861**

Table 4: Evaluation of Model Transferability.

Performance Comparison

Table 2 and Table 3 show a comprehensive comparison of the performance of different baselines and our MetaTRL across the three tasks. Clearly, our MetaTRL achieves the best performance across all three downstream tasks on both datasets, with notable improvements over state-of-the-art baselines. This indicates that different cities share certain common spatio-temporal trajectory patterns, further demonstrating both the necessity of cross-city trajectory representation learning and the effectiveness of MetaTRL. In particular, on the important destination prediction task, our MetaTRL achieves a remarkable average improvement of 143.98% across three metrics on BJ dataset .

Specifically, for the CLA task, our MetaTRL achieves an average improvement of 10.03% across all metrics on both datasets, demonstrating its ability to transfer trajectory pattern from source to target cities. This effectively enhances travel pattern mining in the target city. For the TTE task, our MetaTRL achieves up to 14.53% improvement in the MAPE metric. Despite the significant differences in mobility behaviors across cities, our proposed meta-knowledge extraction and transferable time encoder enable the model to learn relative spatio-temporal variation patterns in trajec-

tories, leading to better performance. In the DP task, our MetaTRL achieves a significant average improvement of 80.18% across three metrics on both datasets, with particularly strong performance on the larger BJ dataset. Benefiting from meta-learning directly optimized through the trajectory reconstruction pretext task, as well as the carefully designed shared and private parameterization, our MetaTRL maximizes effective knowledge transfer while preserving city-specific information.

Model Transferability Study

To further analyze and demonstrate the superiority of our MetaTRL over existing SOTA methods in terms of cross-city transferability, we report in Table 4 the performance of two representative baseline models under different training settings. For the baseline methods, they are trained on the BJ dataset and fine-tuned on the Porto dataset. The results show that both START and TrajRL exhibit noticeable performance degradation across various downstream tasks when trained in a cross-city manner, indicating their limited transferability. In contrast, our MetaTRL, which is inherently designed to support cross-city training, achieves significant performance improvement in a cross-city manner, further validating the necessity and effectiveness of cross-city trajectory representation learning. Notably, the experimental results also demonstrate that our MetaTRL, when trained with self-supervised learning directly on the target city, can still achieve the best performance on downstream tasks.

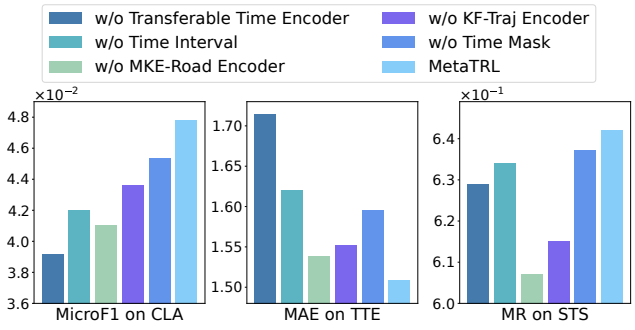
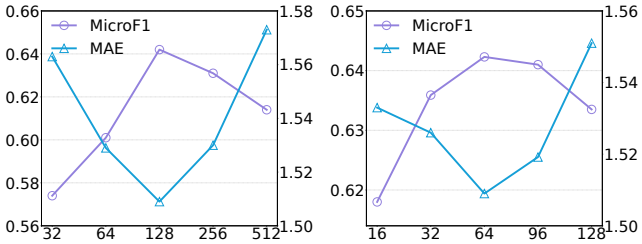


Figure 2: Results of Ablation Study on Porto.

Ablation Study

To evaluate the effectiveness of each module of our TrajRL, we compare the performance of the following variants and the complete MetaTRL: (1)**w/o MKE-Road Encoder**:this variant replaces the whole **Meta Knowlegde Enhanced Road Encoder** with a GAT encoder. (2) **w/o Transferable Time Encoder**: this variant removes the whole **Transferable Time Encoder**. (3) **w/o Time Interval**: this variant removes the shared time interval encoder. (4) **w/o KF-Traj Encoder**: this variant replaces the whole **Knowledge Fusion Trajectory Encoder** with a transformer encoder. (5) **w/o Time Mask**: this variant removes the time masking and recovery self-supervised task.

The experimental results in Figure 2 validate the effectiveness of each module. First, **w/o Transable Time Encoder** significantly reduces performance, which demonstrates the necessity of temporal encoding in TRL. Furthermore, the comparison between **w/o Time Interval** and **MetaTRL** demonstrates that the shared time interval encoding benefits multiple downstream tasks, particularly having a significant impact on travel time estimation. The results of **w/o Time Mask** demonstrate that the time mask and recovery task not only enhances temporal feature learning, but also contributes to spatial feature representation. In addition, thanks to the exploration and integration of shared meta knowledge across regions and cities, our **Meta Knowlegde Enhanced Road Encoder** outperforms the conventional GAT encoder significantly. **Knowledge Fusion Trajectory Encoder** also effectively improves model performance across various tasks through the fusion of cross-city trajectory patterns.



(a) Embedding Size d (b) Batch Size N_b

Figure 3: Impact of Hyper-parameters on TTE and DP on Porto.

Parameter Sensitivity Analysis

We further conduct the parameter sensitivity analysis for critical hyper-parameters, i.e., embedding size and batch size. We report the results of TTE (by MAE) and DP (by MicroF1) on Porto Dataset, the results on BJ are similar. From Figure 3(a), we observe that increasing the embedding size initially enhances model performance. However, excessively large embeddings lead to performance degradation, likely due to sparsity in the learned trajectory representations. Larger batch sizes typically improve convergence, but excessive sizes can harm training stability and cause the model to converge to inferior solutions. This trend is clearly reflected in Figure 3(b).

Conclusion

In this work, we proposed MetaTRL, a novel cross-city trajectory representation learning method. MetaTRL captures both private and shared spatio-temporal features through a self-supervised meta-learning process, enabling effective transfer and integration to target cities. Extensive experiments across three downstream tasks on two real-world datasets show that MetaTRL outperforms current SOTA baselines in both performance and transferability cross cities, which demonstrate the necessity of cross-city TRL and superiority of the proposed MetaTRL.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under grant No. 62176243, the Fundamental Research Funds for the Central Universities under Grant No. 202442005, and Qingdao Natural Science Foundation under Grant No. 24-8-4-zrjj-3-jch.

References

Alessandretti, L. 2022. What human mobility data tell us about COVID-19 spread. *Nature Reviews Physics*, 4(1): 12–13.

Ashukha, A.; Lyzhov, A.; Molchanov, D.; and Vetrov, D. 2020. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*.

Bao, J.; He, T.; Ruan, S.; Li, Y.; and Zheng, Y. 2017. Planning bike lanes based on sharing-bikes' trajectories. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 1377–1386.

Chang, Y.; Qi, J.; Liang, Y.; and Tanin, E. 2023. Contrastive trajectory similarity learning with dual-feature attention. In *2023 IEEE 39th International conference on data engineering (ICDE)*, 2933–2945. IEEE.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PmLR.

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

Dai, S.; Yu, Y.; Fan, H.; and Dong, J. 2022. Spatio-temporal representation learning with social tie for personalized POI recommendation. *Data Science and Engineering*, 7(1): 44–56.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fang, Z.; Du, Y.; Chen, L.; Hu, Y.; Gao, Y.; and Chen, G. 2021. E2dtc: An end to end deep trajectory clustering framework via self-training. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 696–707. IEEE.

Feng, T.; Yan, H.; Wang, H.; Huang, W.; Han, Y.; Liao, H.; Hao, J.; and Li, Y. 2023. ILRoute: A Graph-based Imitation Learning Method to Unveil Riders' Routing Strategies in Food Delivery Service. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4024–4034.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Fu, T.-Y.; and Lee, W.-C. 2020. Trembr: Exploring road networks for trajectory representation learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1): 1–25.

Gong, Y.; He, T.; Chen, M.; Wang, B.; Nie, L.; and Yin, Y. 2024. Spatio-Temporal Enhanced Contrastive and Contextual Learning for Weather Forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(8): 4260–4274.

Ishida, T.; Yamane, I.; Sakai, T.; Niu, G.; and Sugiyama, M. 2020. Do we need zero training loss after achieving zero training error? *arXiv preprint arXiv:2002.08709*.

Jiang, J.; Han, C.; Jiang, W.; Zhao, W. X.; and Wang, J. 2023a. LibCity: A Unified Library Towards Efficient and Comprehensive Urban Spatial-Temporal Prediction. *arXiv preprint arXiv:2304.14343*.

Jiang, J.; Pan, D.; Ren, H.; Jiang, X.; Li, C.; and Wang, J. 2023b. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *2023 IEEE 39th international conference on data engineering (ICDE)*, 843–855. IEEE.

Jin, X.; Park, Y.; Maddix, D.; Wang, H.; and Wang, Y. 2022. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, 10280–10297. PMLR.

Jin, Y.; Chen, K.; and Yang, Q. 2022. Selective Cross-City Transfer Learning for Traffic Prediction via Source City Region Re-Weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 731–741. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393850.

Li, X.; Zhao, K.; Cong, G.; Jensen, C. S.; and Wei, W. 2018. Deep representation learning for trajectory similarity computation. In *2018 IEEE 34th international conference on data engineering (ICDE)*, 617–628. IEEE.

Liang, Y.; Ouyang, K.; Wang, Y.; Liu, X.; Chen, H.; Zhang, J.; Zheng, Y.; and Zimmermann, R. 2022. TrajFormer: Efficient trajectory classification with transformers. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1229–1237.

Liu, H.; Wu, H.; Sun, W.; and Lee, I. 2019. Spatio-temporal GRU for trajectory classification. In *2019 IEEE International Conference on Data Mining (ICDM)*, 1228–1233. IEEE.

Lu, B.; Gan, X.; Zhang, W.; Yao, H.; Fu, L.; and Wang, X. 2022. Spatio-Temporal Graph Few-Shot Learning with Cross-City Knowledge Transfer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 1162–1172. ACM.

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

Mao, Z.; Li, Z.; Li, D.; Bai, L.; and Zhao, R. 2022. Jointly contrastive representation learning on road network and trajectory. In *Proceedings of the 31st ACM International Con-*

ference on Information & Knowledge Management, 1501–1510.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Panagopoulos, G.; Nikolentzos, G.; and Vazirgiannis, M. 2021. Transfer graph neural networks for pandemic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4838–4845.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, L.; Geng, X.; Ma, X.; Liu, F.; and Yang, Q. 2018. Cross-city transfer learning for deep spatio-temporal prediction. *arXiv preprint arXiv:1802.00386*.

Wei, T.; Lin, Y.; Lin, Y.; Guo, S.; Zhang, L.; and Wan, H. 2024. Micro-Macro Spatial-Temporal Graph-Based Encoder-Decoder for Map-Constrained Trajectory Recovery. *IEEE Transactions on Knowledge and Data Engineering*.

Wei, Y.; Lin, Y.; Gao, H.; Xu, R.; Yang, S. B.; and Hu, J. 2025. 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*, 2289–2298.

Wei, Y.; Zheng, Y.; and Yang, Q. 2016. Transfer knowledge between cities. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1905–1914.

Xia, H.; Zhang, X.; Cao, Y.; Cao, L.; Yu, Y.; and Dong, J. 2025. Self-supervised trajectory representation learning with multi-scale spatio-temporal feature exploration. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, 779–792. IEEE Computer Society.

Yang, S. B.; Guo, C.; Hu, J.; Tang, J.; and Yang, B. 2021. Unsupervised Path Representation Learning with Curriculum Negative Sampling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 3286–3292.

Yang, S. B.; Hu, J.; Guo, C.; Yang, B.; and Jensen, C. S. 2023. Lightpath: Lightweight and scalable path representation learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2999–3010.

Yao, D.; Hu, H.; Du, L.; Cong, G.; Han, S.; and Bi, J. 2022. TrajGAT: A graph-based long-term dependency modeling approach for trajectory similarity computation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2275–2285.

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

Zou, X.; Yan, Y.; Hao, X.; Hu, Y.; Wen, H.; Liu, E.; Zhang, J.; Li, Y.; Li, T.; Zheng, Y.; and Liang, Y. 2025. Deep learning for cross-domain data fusion in urban computing: Taxonomy, advances, and outlook. *Information Fusion*, 113: 102606.