Spatial Hierarchical Meta-Learning for Single-Point Map Matching

Sijie Ruan
sjruan@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Yiqing Zou
zouyiqing@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Qianyu Yang
yangqy@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Haoyu Han
hanhy@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Yeting Zhang
ytzhang@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Ziqiang Yuan
ziqiangy@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Hanning Yuan*
yhn6@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Shuliang Wang
slwang2011@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Abstract

Inferring the actual road segment purely based on one positioning point, known as single-point map matching (SMM), is vital for many urban applications, e.g., ride-hailing and geo-tagging. However, it is challenging due to inherent positioning errors and extrinsic heterogeneous environments. Existing methods either overlook the heterogeneity of different regions, or do not exploit the commonality of different matching tasks. In this paper, we treat each region as an individual SMM task to tackle the heterogeneity, and propose Spatial Hierarchical Meta-Learning for SMM (SHSMM) to learn the shared knowledge across tasks. SHSMM is equipped with a Dual-view Map Matcher to perform the matching, which can perceive the knowledge of road segments globally. To learn the task-specific model parameters, SHSMM modulates initial parameters and scales the local update learning rate based on hierarchical geographical and semantic knowledge about spatial tasks. A local update learning rate scheduling strategy is further proposed to facilitate the meta-training. Extensive experiments as well as case studies based on two real-world datasets demonstrate the effectiveness of the proposed method.

CCS Concepts

• **Information systems → Spatial-temporal systems**.

Keywords

location-based service; map matching; meta-learning

*Hanning Yuan is the corresponding author.

KDD '25, Toronto, ON, Canada.

ACM ISBN 979-8-4007-1454-2/25/08
https://doi.org/10.1145/3711896.3737133

ACM Reference Format:
Sijie Ruan, Yiqing Zou, Qianyu Yang, Haoyu Han, Yeting Zhang, Ziqiang Yuan, Hanning Yuan, and Shuliang Wang. 2025. Spatial Hierarchical Meta-Learning for Single-Point Map Matching. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3711896.3737133

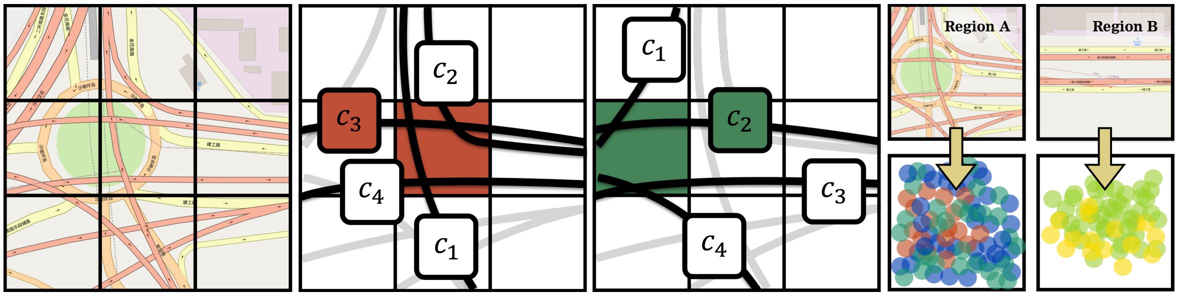
1 Introduction

The development of positioning technology has given rise to many location-based services (LBS). Most LBS applications rely on a fundamental step to process positioning data, i.e., single-point map matching (SMM), which infers the actual road segment on which the mobile object is located based on only one positioning point. This function is equipped in many urban applications, e.g., ride hailing [16, 40], geo-tagging [2, 13], and POI recommendation [30]. For example, in ride-hailing, a user usually stands at a fixed location, and the platform needs to locate her on the correct road so that a more appropriate vacant taxi can be assigned.

Nevertheless, SMM is challenging for two reasons: **(1) Inherent Positioning Errors.** GPS devices have inevitable positioning errors, so we cannot simply adopt the nearest road segment as the matching result. **(2) Extrinsic Heterogeneous Environments.** The positioning error varies from place to place given heterogeneous environments. For example, the positioning error in urban canyons is significantly greater than that in open areas.

Existing work is mainly composed of two categories: 1) universal modeling [14, 20], which trains a universal model to learn the deviation pattern based on all training data in the study region; and 2) separate modeling [1, 33], which trains different models for different road segments. The former sacrifices the modeling of spatial heterogeneity to ensure data sufficiency for the universal model, while the latter allows data to be modeled in a heterogeneous way yet faces the problem of data scarcity.

Recently, meta-learning [12] has shown its superiority in modeling heterogeneous tasks, which allows common knowledge to be shared across tasks. Therefore, if we divide the study region



(a) Task Division (b) Candidate Roads (Brown Task). (c) Candidate Roads (Green Task). (d) Effect of Task Semantics.

Figure 1: Motivating Examples.

into different subregions, as shown in Figure 1(a), treat SMM for data falling in different subregions as different tasks, and introduce meta-learning, we can not only tackle the spatial heterogeneity, but also mitigate the data scarcity via task knowledge sharing.

However, simply applying existing meta-learning methods, e.g., MAML [9], is still insufficient for SMM: **Firstly, constructing meta-learning tasks will disrupt the integrity of the road network.** For example, Figure 1(b) and 1(c) show all candidate road segments in the brown task, and the green task, respectively. We find road segment c_3 in the brown task, and road segment c_2 in the green task are essentially the same from the global view. However, they would be separately considered if we treat each subregion as a separate SMM task. **Secondly, most existing work overlooks the explicit prior knowledge in spatial tasks, they only derive beneficial task dependencies from the meta-training set itself [10, 15, 37, 38].** The spatial prior knowledge is composed of geographical knowledge and semantic knowledge. The former reveals spatial closeness and spatial hierarchy according to the First Law of Geography [25], while the latter indicates the environmental discrepancy of task regions. For example, the top row in Figure 1(d) shows the environments of two regions. We can find the road condition in Region A is rather complex, while that is simpler in Region B. The bottom row in Figure 1(d) shows the positioning points in those regions. Same-colored points denote that their corresponding road segments are the same. As can be observed, SMM for Region A is much more difficult than that for Region B, which is strongly related to the environment.

To this end, in this paper, we propose Spatial Hierarchical Meta-Learning for SMM (SHSMM). SHSMM is a gradient-based meta-learning method that considers the spatial properties of tasks and is dedicated to SMM. We also devise a variant to tackle the "with Destination" (SMMD) setting, which is also useful in some applications, e.g., ride-hailing. To tackle the disruption of the road network, we design a base model, i.e., Dual-view Map Matcher, which incorporates a globally shared knowledge bank to store the global knowledge of each road segment. To consider the explicit prior knowledge of spatial tasks in meta-learning, we use the geographical knowledge to modulate the globally shared initial parameters into region-specific initial parameters, and then leverage the semantic knowledge to scale the local update learning rate to a task-specific one. Both types of knowledge are considered hierarchically to maintain the spatial properties. We further propose a local update learning rate scheduling strategy, i.e., Incremental Dropout, during the meta-training process, to mitigate the overfitting caused by the introduction of parameters that would not be locally updated. We summarize our contributions as follows:

- We present the first attempt to leverage meta-learning to tackle SMM, which not only considers the region heterogeneity but also preserves the common matching knowledge across regions.
- We propose SHSMM, which is equipped with a Dual-view Map Matcher and two spatial knowledge fusion modules, i.e., Initial Parameter Modulation (IPM) and Learning Rate Scaling (LRS). Incremental Dropout is further proposed to avoid overfitting.
- Extensive experiments as well as case studies on two real-world datasets demonstrate the effectiveness of SHSMM. We also have released our code for public use[1].

2 Preliminaries

2.1 Problem Formulation

Definition 1 (Positioning Point). A positioning point p is a spatial point, denoted as $p = \langle lng, lat \rangle$. It captures the longitude lng and latitude lat of an object. The positioning point might deviate slightly from the actual location of the object due to positioning errors.

Definition 2 (Road Network). A road network is a directed graph, denoted as $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where each vertex $v \in \mathcal{V}$ is associated with a location $\langle lng, lat \rangle$, and each road segment $e \in \mathcal{E}$ is a triplet $\langle eid, u, v \rangle$, denoting its id eid and the connectivity from u to v.

Definition 3 (Spatial Knowledge). The spatial knowledge is composed of the geographical knowledge $\mathcal{K}^{Geo}$ and semantic knowledge $\mathcal{K}^{Sem}$ of the study region. $\mathcal{K}^{Geo}$ is just the location information, and $\mathcal{K}^{Sem}$ can be arbitrary semantics describing the region, e.g., digital map or remote sensing images.

Problem Statement (Single-point Map Matching). *Given a positioning point p generated by an object and a road network $\mathcal{G}$, as well as the spatial knowledge, infer the actual road segment $e \in \mathcal{E}$ where the object is located on.*

In the **Single-point Map Matching with Destination (SMMD)** setting, together with the inputs of SMM, the destination p^{dest}, which is another spatial point, is also given.

[1] https://github.com/zouyiqing-221/SHSMM

2.2 Model-Agnostic Meta-Learning

Model-Agnostic Meta-Learning (MAML) [9] is one of the most prevalent methods, which takes initial parameters $\boldsymbol{\theta}_0$ of the base model as the shared knowledge. It has two key processes:

Knowledge Utilization. Assuming we already have the globally shared initial parameters $\boldsymbol{\theta}_0$, for a task $\mathcal{T}_i$, MAML adapts to the task-specific parameter $\boldsymbol{\theta}_i$ based on several steps of gradient descent, i.e., local updates, over support set $\mathcal{D}_i^s$ generated from $\mathcal{T}_i$. For example, when using one-step gradient descent, we have

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_0 - \alpha \nabla_{\boldsymbol{\theta}_0} \mathcal{L}(f_{\boldsymbol{\theta}_0}, \mathcal{D}_i^s) \tag{1}$$

where α is the local update learning rate, and $\mathcal{L}$ is the task loss.

Knowledge Extraction. To learn the initial parameters $\boldsymbol{\theta}_0$, MAML aims to learn a globally shared $\boldsymbol{\theta}_0$ that can minimize the prediction error over $\mathcal{D}_i^q$ after local updates are performed based on $\mathcal{D}_i^s$ for every $\mathcal{T}_i$ in $\mathscr{T}_{tr}$. Formally, the training objective is given as follows, which can be optimized via stochastic gradient descent:

$$\boldsymbol{\theta}_0^* = \arg\min_{\boldsymbol{\theta}_0} \sum_{\mathcal{T}_i \in \mathscr{T}_{tr}} \mathcal{L}(f_{\boldsymbol{\theta}_0 - \alpha \nabla_{\boldsymbol{\theta}_0} \mathcal{L}(f_{\boldsymbol{\theta}_0}, \mathcal{D}_i^s)}, \mathcal{D}_i^q) \tag{2}$$

3 Methodology

3.1 Overview

In this section, we present our proposed Spatial Hierarchical Meta-Learning for SMM (SHSMM). We first divide the study region into several uniform grid regions and treat SMM in each grid as a task. For each task $\mathcal{T}_i$, data within the grid region will be further split into support set $\mathcal{D}_i^s$ and query set $\mathcal{D}_i^q$. SHSMM follows the paradigm of gradient-based meta-learning, which is depicted in Figure 2(a). We first introduce a base model Dual-view Map-Matcher, which contains some global-update-only parameters to learn common knowledge of road segments across tasks. For each task, the parameters $\boldsymbol{\theta}_0$ that would be locally updated in Dual-view Map-Matcher would be sent into an Initial Parameter Modulation (IPM), which gives region-specific parameters $\boldsymbol{\theta}_{r_i}^0$ based on the geographical knowledge $\mathcal{K}_i^{Geo}$. Then, we derive a task-specific local update learning rate α_i via Learning Rate Scaling (LRS) based on the semantic knowledge $\mathcal{K}_i^{Sem}$. After that, K-step local updates would be conducted to transform $\boldsymbol{\theta}_{r_i}^0$ into task-specific parameters $\boldsymbol{\theta}_i^K$ based on $\mathcal{D}_i^s$ and α_i. Finally, we can use the base model with $\boldsymbol{\theta}_i^K$ and $\mathcal{D}_i^q$ to perform the global updates for $\boldsymbol{\theta}_0$ and learnable parameters in IPM and LRS. We further design an Incremental Dropout mechanism to postpone the overfitting. Next, we elaborate on each design in detail.

3.2 Dual-view Map Matcher

Dual-view Map Matcher serves as the base model. Given the road network $\mathcal{G}$ and a position p, it takes a set of nearby road segments as candidates C, and infers the actual road segment for p.

Main Idea. A straightforward way is to adapt an existing trajectory map matching model [3, 7, 19] as the base model, in which the dimension of the outputs is fixed and equal to the size of the road segment set. Such a design is reasonable when all data are considered as a single set, but is not efficient as we divide the study region into grid-based tasks when using meta-learning.

Instead, here we take SMM as a candidate selection problem, in which the dimension of the outputs is equal to the size of the candidate set, which is not fixed for all samples and is much smaller than the size of the whole road segment set.

The basic idea is to renumber IDs for the subset of road segments for each task, i.e., local edge IDs, and use a local embedding layer to embed them. However, a side effect is that nearby tasks may share the same road segments, but the knowledge of the same road segments can not be shared given that they may have different local edge IDs in different tasks. To this end, we propose the Dual-view Map Matcher, which not only considers road segments from the local view but also leverages their knowledge from the global view, i.e., across tasks.

Implementation. To implement, for each positioning point p, we construct the candidate road segments C, which is a set of road segments within D meters from p, i.e., $C = \{c | \forall c \in \mathcal{G}.\mathcal{E} \wedge dist(p, c) \leq D\}$. Following [3, 17, 19, 33], the matching distance $dist(p, c)$ is the perpendicular distance if the projection of p is on c, otherwise, it is the distance to the closest endpoint of c. For each candidate road segment $c_j \in C$, we calculate its distance d_j to p, retrieve its local edge ID eid_j^l, which is renumbered by the associated task, as well as its global edge ID eid_j^g, i.e., $c.eid$.

Those features would be fed into the proposed Dual-view Map Matcher, which is shown in Figure 2(b). For each candidate c_j, we first use a local embedding layer $\mathtt{Emb}_l$ to embed eid_j^l, which is then concatenated with d_j, followed by a fully connected (FC) layer to generate candidate local representation, i.e., $\mathbf{h}_j = \mathtt{FC}([\mathtt{Emb}_l(eid_j^l) \| d_j])$, where $\|$ is the concatenation operation. In candidate selection, it is beneficial to model the inter-dependency among candidates as reported in [20, 27, 32]. Therefore, we further feed all candidate local representations into a Transformer encoder [26] to obtain context-aware local representations $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n$:

$$\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n = \mathtt{TransEnc}(\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}) \tag{3}$$

Next, each context-aware local representation $\mathbf{z}_j$ would be concatenated with the global knowledge of each road segment via a global embedding layer $\mathtt{Emb}_g$, and fed into a FC layer to obtain the final representation of each candidate:

$$\tilde{\mathbf{z}}_j = \mathtt{FC}([\mathbf{z}_j \| \mathtt{Emb}_g(eid_j^g)]) \tag{4}$$

Note that, the parameters in $\mathtt{Emb}_g$ would not be updated during the local update process, which only serves as a knowledge bank to store the global knowledge about road segments. Its parameters are only updated during the global update process. In this way, the knowledge about road segments is globally learned across tasks.

Finally, another FC layer is applied to generate logits, followed by a softmax activation over all candidates to obtain the matching probabilities:

$$\hat{y}_1, \hat{y}_2, ..., \hat{y}_n = \mathtt{softmax}(\{\mathtt{FC}(\tilde{\mathbf{z}}_1), \mathtt{FC}(\tilde{\mathbf{z}}_2), ..., \mathtt{FC}(\tilde{\mathbf{z}}_n)\}) \tag{5}$$

The candidate road segment with the maximum probability is selected as the inferred matched road segment.

As for SMMD, to incorporate destination, we calculate the orientation of the destination to each candidate road segment $c_j \in C$ via $o_j = \frac{< c_j.v - c_j.u, p^{dest} - c_j.v >}{|c_j.v - c_j.u| \cdot |p^{dest} - c_j.v|}$, where $< \cdot, \cdot >$ is the inner product, and then replace the aforementioned candidate local representation with $\mathbf{h}_j = \mathtt{FC}([\mathtt{Emb}_l(eid_j^l) \| d_j \| o_j])$.

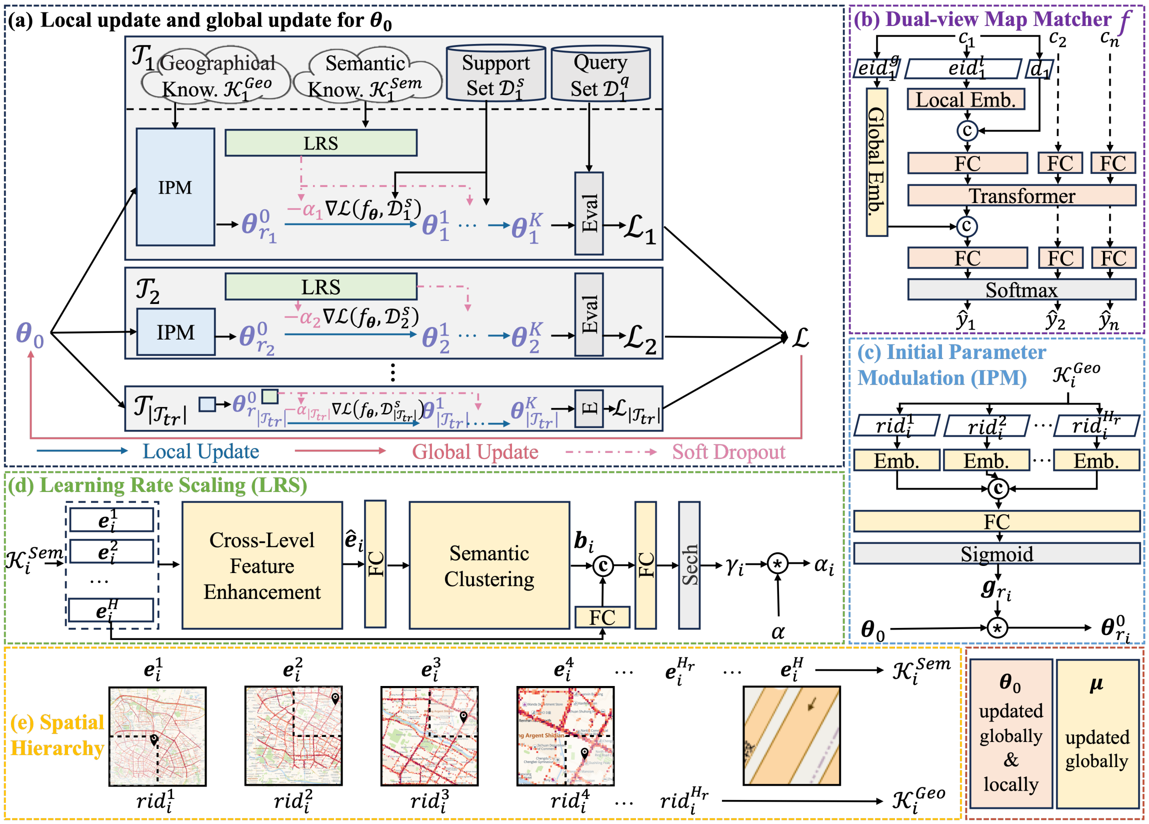


Figure 2: Framework of SHSMM.

3.3 Initial Parameter Modulation

After we have designed the base model, now we turn to learn its parameters based on meta-learning. One important step is to determine the starting point of local update, i.e., initial parameters, for each task. In SHSMM, we use Initial Parameter Modulation to obtain region-specific initial parameters based on geographical knowledge to give each task a better start.

Main Idea. In MAML [9], all tasks share the same starting point. Recent studies [28, 37] found it may be difficult to adapt to massive tasks due to task diversity. Therefore, they proposed to organize the tasks by customizing the initial parameters. The core idea here is to transform the global initial parameters to task-specific [28] or cluster-specific [37] initial parameters for better adaption. However, these methods attempt to organize the tasks based on samples in the support set, which are not designed for spatial tasks. Yet, for a spatial task, we aim to consider natural geographical properties to ensure a more reasonable organization of meta-tasks. Tobler's First Law (TFL) [25] indicates that "near things are more related than distant ones". Inspired by this, we use a hierarchical manner to encode the tasks, so that tasks in the same region will share same region-specific initial parameters, and tasks in closer regions will have more commonalities in their initial parameters. In this way, we managed to organize the spatially related tasks using the geographical knowledge.

Implementation. We first define a hierarchy level of the study region, i.e., H_r, and then use quad-tree [8] to recursively divide the study region into four subregions until H_r is reached, as shown in Figure 2(e). We aim to modulate the global initial parameters for each region at level H_r.

For each task $\mathcal{T}_i$, we assume its affiliated region at level H_r is r_i. The region code of r_i also follows a quad-tree coding scheme, which is a sequence of numbers (range from 0-3) indicating the traversing path from the root region to the current region. We transform it back to a sequence of numbers, denoted as $< rid_i^1, rid_i^2, ..., rid_i^{H_r} >$, and treat each of them as the position indicator at a certain granularity. We regard them as the geographical knowledge of $\mathcal{T}_i$, i.e., $\mathcal{K}_i^{Geo}$. In this way, not only tasks in the region with the finest granularity can share the same geographical knowledge, but tasks in adjacent regions can also have similar knowledge as well.

Given $\mathcal{K}_i^{Geo}$, the modulation process is shown in Figure 2(c). We first use an embedding layer to transform them into dense representations, and then concatenate and fuse those representations to obtain the hidden geographical knowledge $\mathbf{r}_i$:

$$\mathbf{r}_i = \mathtt{FC}(\|_{h=1}^{h=H_r} \mathtt{Emb}(rid_i^h)) \tag{6}$$

After that, we use a fully connected layer followed by a sigmoid activation to transform the hidden geographical knowledge $\mathbf{r}_i$ into a parameter gate $\mathbf{g}_{r_i} = \mathtt{Sigmoid}(\mathtt{FC}(\mathbf{r}_i))$.

Finally, the region-specific initial parameters $\boldsymbol{\theta}^0_{r_i}$ are obtained by modulating global initial parameters $\boldsymbol{\theta}_0$ with $\mathbf{g}_{r_i}$:

$$\boldsymbol{\theta}^0_{r_i} = \boldsymbol{\theta}_0 \circ \mathbf{g}_{r_i} \tag{7}$$

where $\circ$ is the Hadamard product.

3.4 Learning Rate Scaling

Given region-specific initial parameters $\boldsymbol{\theta}^0_{r_i}$, the local update for each task can be performed. We further use Learning Rate Scaling to scale the local update learning rate of each task based on the semantic knowledge to achieve customized parameter updating.

Main Idea. To obtain a good map matcher for each task, good initial parameters and effective local updates are both important. In MAML [9], the local update for each task is performed based on its support set and a shared local update learning rate. However, for different tasks, their adaptive ability may differ from each other due to the great diversity even though they are in the same region. For example, some tasks may have more support samples while others may have fewer, and some tasks may have a large set of candidate road segments while the size of others may be rather limited. This motivates us to use different learning rates for different tasks. Since the diversity caused by the above phenomenon is mainly due to the difference in artificial and natural environments, we aim to leverage the semantic knowledge of tasks to scale the local update learning rate. However, if we only use specific semantics of a task to module the local update learning rate, the knowledge-sharing across tasks would be hurt. To further integrate cross-task knowledge, for each task, we first extract the semantic features from different granularities so that the learning rate scaling considers not only the semantics of a task, but also its semantic context in the spatial hierarchy, and then we cluster semantic features of different tasks to obtain task context-aware local update learning rates.

Implementation. Since the images of digital maps are easily accessible, and can reflect the built and natural environments of task regions, we can use the digital map images as semantic knowledge. We assume the region of a task $\mathcal{T}_i$ is at spatial level H, i.e., r^H_i, then we can extract features from $< r^1_i, r^2_i, ..., r^H_i >$ to capture the hierarchical semantics, which is also shown in Figure 2(e). To extract features at level H, we use a pre-trained image encoder [23] to transform the region image into a hidden representation $\mathbf{e}^H_i$. For upper levels, level h for example, we average over all hidden representations of level-H subregions within r^h_i, i.e., mean-pooling, to obtain level-h hidden representation $\mathbf{e}^h_i$, i.e., $\mathbf{e}^h_i = \frac{1}{|\{r^H_j | \forall r^H_j \in r^h_i\}|} \sum_{r^H_j \in r^h_i} \mathbf{e}^H_j$. In this way, we represent the semantic knowledge of a task $\mathcal{K}^{Sem}_i$ as $< \mathbf{e}^1_i, \mathbf{e}^2_i, ..., \mathbf{e}^H_i >$.

The scaling process is shown in Figure 2(d). To fuse knowledge at different levels and enhance the representation, we use an attention mechanism [34] to obtain cross-level semantic representations $< \tilde{\mathbf{e}}^1_i, \tilde{\mathbf{e}}^2_i, ..., \tilde{\mathbf{e}}^H_i >$:

$$\begin{aligned} \tilde{\mathbf{e}}^{h_a}_i &= \sum_{h=1}^{H} \alpha^{ab}_i \mathbf{e}^{h_b}_i \mathbf{W}_e \\ \alpha^{ab}_i &= \frac{exp(< \mathbf{e}^{h_a}_i \mathbf{W}_e,\ \mathbf{e}^{h_b}_i \mathbf{W}_e >))}{\sum_{h_o=1}^{H} exp(< \mathbf{e}^{h_a}_i \mathbf{W}_e, \mathbf{e}^{h_o}_i \mathbf{W}_e >))} \end{aligned} \tag{8}$$

where $\mathbf{W}_e$ is the learnable matrix for common knowledge extraction. After that, we concatenate cross-level semantic representations of different levels to obtain the hierarchical semantics of task $\mathcal{T}_i$, i.e., $\tilde{\mathbf{e}}_i = \|_{h=1}^{H} \tilde{\mathbf{e}}^h_i$.

Then, to share the semantic knowledge of different tasks, we transform the hierarchical semantics $\tilde{\mathbf{e}}_i$ to a task context-aware semantics $\mathbf{b}_i$ by exchanging the hierarchical semantic features of different tasks via soft clustering [37]. The basic idea of soft clustering is to generate the representation of the input via the weighted sum of cluster centers, which are hidden vectors learned during the global update process among all tasks, such that all tasks can sufficiently share semantic knowledge.

Next, based on the task context-aware semantics $\mathbf{b}_i$, we use it to generate the task-specific scaling factor γ_i, which will be used to scale the local update learning rate α. More specifically, we first use a shortcut mechanism [11] to fuse $\mathbf{b}_i$ and the original semantic representation $\mathbf{e}^H_i$, then transform the concatenated vector to a scalar, and finally use an activation function to normalize it into (0,1]:

$$\gamma_i = \mathtt{Sech}(\mathtt{FC}([\mathtt{FC}(\mathbf{e}^H_i) \| \hat{\boldsymbol{b}}_i)]) \tag{9}$$

where $\mathtt{Sech}(\cdot)$ is the Sech function, i.e., $\mathtt{Sech}(\cdot) = \frac{2}{exp(\frac{\cdot}{\sigma}) + exp(-\frac{\cdot}{\sigma})}$ (σ is a scaling constant), which serves as the activation. The reason that we do not use common activations, e.g., `Sigmoid`, is to prevent the output from converging to the value 1, which will lead this module to be redundant and useless.

Eventually, we get the task-specific local update learning rate by multiplying γ_i to the original learning rate α, which is aware of the task context:

$$\alpha_i = \alpha \cdot \gamma_i \tag{10}$$

3.5 Optimization with Incremental Dropout

Given the above illustration, several sets of parameters should be learned during the meta-training, i.e., the global initial parameters and the global embedding layer in the base model, as well as parameters in IPM and LRS modules.

To optimize those parameters, we need to choose the form of loss function of a learning task. Here, we use the cross-entropy loss, which is widely used for selection problems [20, 27, 32]:

$$\mathcal{L}(\phi_{(\boldsymbol{\theta},\boldsymbol{\mu})}, \mathcal{D}) = - \sum_{(C,\mathbf{y}) \in \mathcal{D}} \sum_{i=1}^{n} y_i \log \phi_{(\boldsymbol{\theta},\boldsymbol{\mu})}(C) \tag{11}$$

where $\mathcal{D}$ is the support set or query set of a learning task depending on the update stage (local or global update). $\phi_{(\boldsymbol{\theta},\boldsymbol{\mu})}$ denotes the inference process of SHSMM to predict the matched road segment, which contains parameters $\boldsymbol{\theta}$ that can be locally updated, and parameters $\boldsymbol{\mu}$ that would only be updated during the meta-training stage. $\mathbf{y}$ is the one-hot representation of the ground-truth matching result, i.e., only the index of the actual matched road segment in candidates C is set to 1. Given $\mathcal{L}$, the parameters in SHSMM can be learned using Equation 2.

However, we find SHSMM overfits the meta-training datasets quickly, which may be attributed to the distribution shift and the introduction of global-update-only parameters $\boldsymbol{\mu}$. Those factors would harm the generalization capability of SHSMM. Therefore, we further present a local update learning rate scheduling method, named Incremental Dropout, to postpone the overfitting issue.

The basic idea is that when SHSMM is meta-trained at the early stage, we have chance to locally update the base model with a small gradient step (i.e., "dropout" the local update softly), which can force SHSMM to learn initial parameters with better generalization ability; when SHSMM is meta-trained at the late stage, the local update learning rate should be smaller (increase the extent of soft "dropout"), since we expect that an ideal task-specific model should be obtained by slight task adaption at that stage. To achieve both goals, during the meta-training stage, we apply a factor to the task-specific local update learning rate α_i as follows:

$$\hat{\alpha}_i = [(1-\eta)^n]^t \alpha_i,\ t \sim \varepsilon(\lambda) \tag{12}$$

where $\eta \in (0, 1]$ is the soft dropout rate, indicating the extent to make α_i smaller, and n is the current epoch during the meta-training. t is a stochastic variable, that follows the exponential distribution, and λ is the hyperparameter of the distribution. Intuitively, with the increase of n, the expectation of $[(1-\eta)^n]^t$ becomes smaller. However, due to the stochastic characteristic introduced by t, α_i also has a chance to be small when n is small. The pseudocode of the meta-training process with IDP is given in Appendix A.1.

Note that, $\hat{\alpha}_i$ is only applied at the local update process during the meta-training stage to avoid overfitting. During meta-validation/testing, we still use the task-specific local update learning rate α_i produced by LRS to perform the local update.

3.6 Efficiency Analysis

Training Time. In the meta-training phase, local update is conducted for each task respectively, and global update is conducted for all tasks. Assume T_{IPM}, T_{LRS} and T_{DMM} are the time for feed-forward in IPM, LRS and Dual-view Map Matcher, T_g and T_l are time for backward propagation during global and local update, N_{spt} and N_{qry} are sizes of support and query set, then the time cost is $N_{epoch}|\mathcal{T}|(T_{IPM} + T_{LRS} + K(N_{spt}T_{DMM} + T_l) + N_{qry}T_{DMM} + T_g)$.
Inference Time. In the meta-testing phase, only local update should be conducted. Thus, we can precompute the parameter gate and the learning rate scaler, so that T_{IPM} and T_{LRS} can be omitted, and the inference time is $K(N_{spt}T_{DMM} + T_l) + T_{DMM}$ for each sample. To further accelerate the inference speed, we can maintain the local updated model in memory, in which case the time cost is reduced to T_{DMM}.

4 Experiments

4.1 Experimental Setup

Datasets. Although a single positioning point is widely accessible, its label is difficult to obtain. The common practice in the literature [3, 33] is to first perform map matching for trajectories (the accuracy is rather high when the trajectories are at high sampling rate as reported in [17]), and then sample intermediate points with matched road segments to construct SMM datasets (the last point in the trajectory is also taken to form SMMD datasets) to represent different SMM(D) scenarios. We follow this paradigm and use two real-world trajectory datasets from a ride-hailing company, i.e., DiDi[2], whose sampling rate is 15s on average. Those trajectories are between Oct. 1st to Oct. 7th, 2018 from an area of 10 km×9 km in Xi'an and Chengdu, China, respectively. The road structure of Xi'an is grid-like, while that of Chengdu is star-like, which are suitable to evaluate the applicability of the proposed method. We perform map matching on digital maps from OpenStreetMap (OSM)[3], and randomly select 1% intermediate points as well as their labels from each trajectory to ensure diversity. After that, we take 80% positioning points for training, 10% points for validating, and the remaining 10% points for testing. The detailed statistics of resulting SMM(D) datasets are shown in Table 1. The semantic knowledge of study regions, i.e., digital map images, are also crawled from OSM.

Table 1: Data Descriptions.

Datasets	Xi'an	Chengdu
# of Roads	6,160	6,500
Avg. # of Candi. Roads	5.84	6.06
Max. # of Candi. Roads	33	34
Train/Val/Test	504K/71K/71K	763K/105K/105K

Baselines. We compare SHSMM with three types of baselines: (1) heuristics-based methods, which perform SMM based on the nearest neighbor, i.e., MinDist; (2) traditional map matching methods, which include PSMM [33], DeepMM [7], RNTrajRec [3] and DTInf+ [20]; (3) meta-learning-based methods, which include MAML [9] and other variants that deal with task heterogeneity, i.e., HSML [37] and Adaptive-MAML [15] and task sparseness, i.e., MetaMix [35] and MLTI [39]. In meta-learning-based methods, the base model is replaced by ours. For fair comparison, we have tried to concatenate the spatial prior knowledge into baselines, and report the performance if it is higher than the original version. Detailed descriptions of baselines are introduced in Appendix A.2.
Meta-learning Setup. We divide the whole region into $2^{H-1} \times 2^{H-1}$ uniform grids, each of which is considered as a single task. H is set to 10 by default, resulting in regions with size $20m\times 18m$. During the meta-training phase, training data are divided into support and query set. During the meta-validation/testing phase, the whole training data is used as the support set, and the validation set and test set are used as the query set, respectively. The detailed statistics of meta-tasks are shown in Table 2.

Table 2: Data Statistics in Meta-Tasks.

Datasets	Xi'an	Chengdu
Avg. # of Roads	6.96	7.52
Avg. Support Set Size	23.45	29.20
Avg. Query Set Size	3.30	4.00
# of Tasks	21,514	26,138

Training Details & Hyperparameters. Our method, as well as the baselines, are implemented by PyTorch with one GeForce RTX 4090 GPU. For the spatial hierarchy, we set $H_r = 4$. For the learning rate scaling module, we set $\sigma = 20$. For the local update, local update learning rate $\alpha = 0.01$, and the step number of local update $K = 5$. During the meta-training, each training batch consists of 32 tasks, and we use the Adam optimizer, setting the global update learning

[2]https://gaia.didichuxing.com/

[3]http://www.openstreetmap.org/

Table 3: Overall Performance.

City	Xi'an		Chengdu	
Method	**SMM**	**SMMD**	**SMM**	**SMMD**
MinDist	63.28%	63.28%	63.81%	63.81%
PSMM	67.32%	67.65%	67.34%	68.08%
DTInf+	72.26%	80.81%	70.21%	78.57%
DeepMM	73.49%	79.97%	70.81%	76.98%
RNTrajRec	74.55%	80.79%	72.83%	81.99%
MetaMix	73.74%	81.46%	70.97%	78.14%
Adaptive-MAML	73.92%	82.40%	71.44%	79.05%
MLTI	73.94%	83.07%	71.67%	78.84%
MAML	74.95%	83.57%	72.08%	79.79%
HSML	75.66%	84.10%	73.83%	81.25%
SHSMM (Ours)	**76.85%**	**84.69%**	**74.60%**	**82.09%**

rate $\beta = 0.001$, soft dropout rate $\eta = 0.03$ and $\lambda = 10$. After the loss on the validation set no longer decreases for 5 epochs, then early stop will occur. The detailed structure of Dual-view Map Matcher can be found in Appendix A.3.

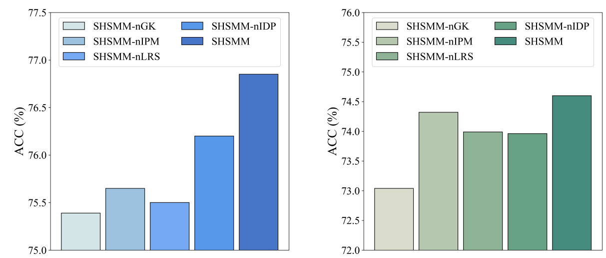
Evaluation Metrics. We follow existing work [7, 22, 33] to employ the matching accuracy (ACC) to evaluate the model performance.

4.2 Overall Performance

The overall performance of our SHSMM and other baselines are presented in Table 3. As we can see, SHSMM achieves the best performance on the datasets collected from two cities for both SMM and SMMD problems. This indicates that SHSMM is effective in dealing with the SMM(D) problem which inherits the characteristic of spatial correlations and spatial heterogeneity.

Specifically, the meta-learning-based models generally outperform traditional map matching methods (PSMM, DTInf+, DeepMM, and RNTrajRec), which indicates the appropriateness of adopting a meta-learning method to solve the problem. For the meta-learning methods that adopt a hierarchical structure to organize tasks and initial parameters (HSML and Adaptive-MAML), HSML, which builds the structure based on task representation, performs better than MAML, while Adaptive-MAML, which builds the structure based on the task complexity, performs worse than MAML. These results lead to the conclusion that a suitable hierarchy is of critical importance for problems facing severe task heterogeneity and complex correlations, and a wrong hierarchy may even lead to a worse performance. Moreover, since we build grid-based non-overlapped tasks to solve the SMM(D) problem, this indeed leads to some kind of task sparseness. Yet the methods aim to deal with task sparseness and improve generalization ability through task augmentation (MetaMix, MLTI), fail to improve the performance compared to MAML. A possible explanation is that the augmentation simply based on the samples (i.e. inputs and labels) cannot effectively reconstruct spatial correlations and spatial continuity.

The performance of the meta-learning-based methods demonstrates the importance of building a suitable hierarchy and reconstructing continuity based on spatial characteristics and the tendency to obey TFL for a spatial problem. This brings us to partially understand why SHSMM can outperform the others. The reason is that SHSMM incorporates both local and global knowledge, and builds spatial-based hierarchy. The former design reunites the road network that has been broken down by the non-overlapped meta-tasks, while the latter one generates region-specific initial parameters and task-specific local update learning rates.



(a) Xi'an (b) Chengdu

Figure 3: Ablation Study.

From the efficiency aspect, though SHSMM takes more time than simple heuristics or machine learning baselines to make task-specific adaption, it is still quite practical for real-world usage, which only takes 0.0285s/sample.

4.3 Ablation Study

We compare SHSMM with the following variants:

- **SHSMM-nGK**, which removes the global edge ID embedding from the base model of SHSMM.
- **SHSMM-nIPM**, which removes the IPM module from SHSMM, so that all tasks share the same initial parameters $\boldsymbol{\theta}_0$.
- **SHSMM-nLRS**, which removes the LRS module from SHSMM, so that all the tasks share the same local update learning rate α.
- **SHSMM-nIDP**, which replaces the Incremental Dropout during meta-training phase with the commonly used dropout technique.

As shown in Figure 3, SHSMM outperforms all of the 4 variants on the datasets of two cities for the SMM problem, which proves the effectiveness of our design in tackling severe task heterogeneity problems existing in real-world datasets. Specifically, the removal of the global edge ID embedding (SHSMM-nGK) limits the knowledge sharing between tasks, indicating the importance of introducing both local view and global view when the integrity of the dataset is disrupted due to the task division for adopting meta-learning. The removal of the initial parameter modulator (SHSMM-nIPM) and local update learning rate scaler (SHSMM-nLRS) makes it impossible for the tasks to adopt a more customized starting point (i.e., the initial parameter) and a more favorable adapting route (i.e., the inner loop optimization process affected by local update learning rate), which is indeed important when dealing with heterogeneous tasks. Moreover, the replacement of dropout mechanism (SHSMM-nIDP) degrades the performance, suggesting that it is a possible way to improve generalization capability. Since adopting global-update-only parameters $\boldsymbol{\mu}$ may arrive at the stage of overfitting much faster than $\boldsymbol{\theta}_0$, and will thus limit the generalization ability, stochastically and incrementally dropping out more local update learning rate as epoch grows can balance the training speed between $\boldsymbol{\mu}$ and $\boldsymbol{\theta}_0$. Note that the differences in module effects between two cities may possibly be caused by structure and distribution differences.

4.4 Hyperparameter Analysis

Since SHSMM involves some hyperparameters, we further study the impact of several important hyperparameters.

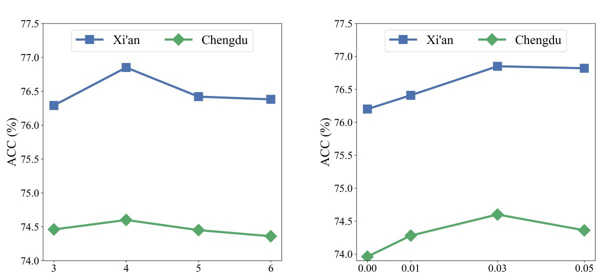
Study on Level Number H. The whole region will be divided into 2^{H-1} rows and columns, and this is the granularity where we construct tasks and extract semantic knowledge. We investigate the effect of the granularity of tasks, which will change the task component, support/query set size and complexity, task-wise semantic knowledge, etc. The result is as shown in Table 4. By increasing H, the grid size decreases and may possibly lead to a smaller yet more homogeneous support set, which in one way makes it possible for the support set to bring more customized guidance, while in the other way breaks down the integrity of the region into more non-overlapped tasks. Based on this trade-off, we can see that for SMM problem, $H = 10$ brings the best performance.

Table 4: Performance under Different Level Number H.

H	9	10	11
Xi'an	75.85%	**76.85%**	76.67%
Chengdu	73.83%	**74.60%**	74.32%

Study on Level Number of Region Hierarchy H_r. We study the influence of H_r used for initial parameter modulation, as shown in Figure 4 (a). By increasing H_r, fewer tasks will share the same initial parameters, which will increase the customization ability in sacrificing the generalization ability. On the contrary, by decreasing H_r, more tasks would share a single starting point, which reduces the learning difficulties for parameters in SHSMM, while hurts the customization ability leading to degenerated performance. Based on this trade-off, we can see that for SMM problem, $H_r = 4$ is the best to effectively balance between customization and generalization ability of the initial parameters $\boldsymbol{\theta}_0$.

Study on Soft Dropout Rate η. We further present the result of using different soft dropout rates η in the meta-training process, as shown in Figure 4 (b). By increasing η, we drop more local update learning rates in the meta-training phase. Though we can increase generalization ability by increasing η, the result suggests that an effective η should not be too small. Based on this trade-off, we can see that for SMM problem, $\eta = 0.03$ is the best choice that allow the model to converge to a better place.



(a) Region hierarchy level H_r. (b) Soft dropout rate η.

Figure 4: Hyperparameter Study.

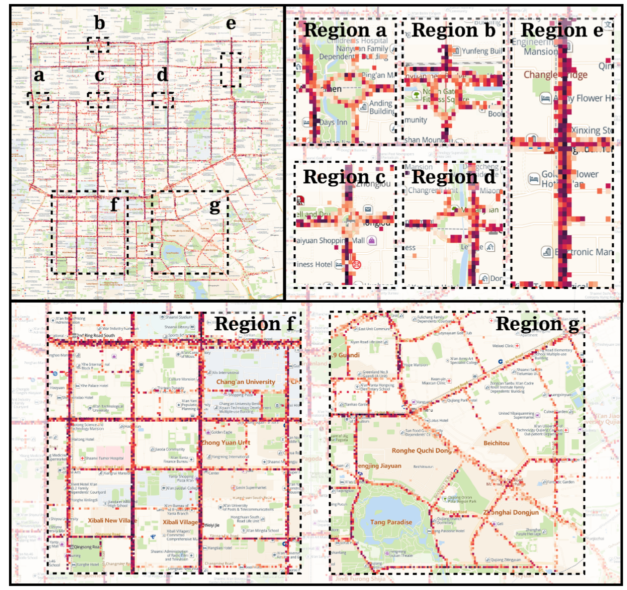


Figure 5: Visualizations of Local Update Learning Rate in Xi'an.

4.5 Case Studies

Visualization of Local Update Learning Rates. We further visualize some cases of the task-specific local update learning rates to better understand the purpose of Learning Rate Scaling in SHSMM. The study region of Xi'an is shown in Figure 5, where each colored pixel corresponds to a task region. The pixel with darker color means its local update learning rate is smaller. From this figure, we have the following two observations.

Firstly, complicated road conditions lead to larger local update learning rates. As shown in the figure, Regions a-d are in complex road conditions (e.g., many overpasses), in which α_i are similar with relatively high values, while Region e contains straight roads, in which most of the tasks have smaller α_i. These phenomena suggest that when we have a task in complicated road condition, we may need a larger α_i. This is a reasonable strategy, which can make our learned initial parameters more generalizable.

Secondly, the hierarchical semantic features successfully capture the local context. Due to the spatial heterogeneity, distant tasks may need different learning strategy even they share similar semantics. Thus, the corresponding hierarchical semantic knowledge would be useful. Regions f and g show two regions with similar road conditions in a micro-view, however, we can find tasks in Region f generally have smaller α_i than tasks in Region g, demonstrating the spatial heterogeneity captured by the semantic hierarchy.

Table 5: Performance under Extreme Cases for SMM on Xi'an. Symbol "/" means no constraints.

# of Roads	/	/	/	/	>10	>15	>20	>10	>15	>15
Support Set Size	≤10	≤5	≤1	0	/	/	/	≤1	≤5	≤1
MinDist	59.65%	58.31%	55.34%	54.42%	57.64%	52.42%	50.05%	39.91%	49.24%	40.00%
PSMM	62.25%	58.35%	53.26%	51.99%	57.64%	50.14%	41.71%	30.49%	32.58%	25.71%
MAML	69.60%	65.16%	57.08%	52.17%	68.30%	61.95%	60.30%	<u>43.95%</u>	<u>53.03%</u>	<u>51.43%</u>
HSML	<u>70.05%</u>	<u>65.98%</u>	<u>58.40%</u>	<u>54.24%</u>	<u>69.80%</u>	<u>64.67%</u>	<u>63.04%</u>	43.05%	50.76%	42.86%
SHSMM (Ours)	**71.09%**	**67.56%**	**60.72%**	**56.86%**	**71.66%**	**66.77%**	**65.47%**	**47.09%**	**57.58%**	**57.14%**

Performance under Extreme Cases. To further illustrate the advantage of organizing tasks based on spatial prior knowledge, we select several representative methods, and apply the trained model to extreme cases, such as tasks with few samples, many roads, or both. Taking SMM problem on Xi'an dataset as an example, the experiment result is shown in Table 5. In all three extreme cases, SHSMM shows better performance against other baselines. Notably, when the support set size and the number of roads both reach extreme states, the effectiveness of HSML [37] significantly declines, while the advantage of SHSMM over other baseline models became more pronounced.

A potential explanation is that HSML [37] organizes tasks purely based on task representations derived from meta-training set, which neglects spatial structure. It achieves overall improvements at the cost of sacrificing performance on complex and hard tasks. Conversely, incorporating spatial prior knowledge to hierarchically establish spatial structures like SHSMM, enhances the generalization capability of meta-learning models, whereby the performance under extreme cases will be simultaneously enhanced rather than being sacrificed. This further underscores the necessity of spatial structure modeling when addressing spatial problems.

5 Related Work

5.1 Map Matching

Map matching aims to associate positioning points of objects with geospatial entities, which is a fundamental step for many urban applications, e.g., traffic monitoring, travel time estimation, and ride-hailing. While associating positioning points with polygonal regions is straightforward, matching with road segments or Points of Interest (PoIs) is non-trivial due to positioning errors. A widely studied problem in this field is trajectories map matching (TMM), which can be handled by Hidden Markov Model [17], recurrent neural network [7, 19, 31], or Transformer [3, 24]. Unlike TMM, SMM is much more difficult because of the lack of contextual information. The pioneering work studying SMM is PSMM [33], which generates the matching result by a probabilistic model. Matching positioning points to PoIs, a variant of SMM, faces similar difficulties, and existing works also leverage a learning-based approach to tackle it [20, 22]. For all aforementioned methods, only PSMM [33] builds several individual models considering spatial heterogeneity, while others build a universal model considering the sufficiency of training data. In this paper, we propose a spatial hierarchical meta-learning-based approach to tackle SMM combining their advantages.

5.2 Meta-Learning

Meta-learning, also known as learning to learn, aims to tell the learner how to learn from the training data. There are three categories, i.e., optimization-based, model-based and metric-based meta-learning, among which optimization-based meta-learning, e.g., MAML [9], is commonly used due to its simplicity and model-agnostic property. Traditional meta-learning methods learn task correlations implicitly, which have limited their generalization capability facing heterogeneous tasks. Further work established different structures according to task representations [37, 38], model performance [15] and domain attributes [10] to organize tasks. These methods have shown the importance of structure in dealing with heterogeneous tasks, yet the structure adopted by these methods is not sufficiently suitable for spatial tasks. There are some studies leveraging meta-learning to tackle spatial tasks, e.g., traffic prediction [18, 36], travel time estimation [5, 6], POI completion [4], and service time prediction [21, 29]. Among them, [18, 21, 29] conduct primary attempts to fuse semantic information into meta-learning. Different from these work, we simultaneously adopt geographical knowledge and semantic knowledge of spatial tasks into meta-learning considering the hierarchical spatial structure, and design a dual-view base model dedicated for SMM.

6 Conclusion

In this paper, we propose a spatial hierarchical meta-learning method for single-point map matching. We first divide the study region into uniform grid regions, then treat each region as an SMM task. For each task, SHSMM leverages Dual-view Map Matcher to perform the matching which considers both local and global knowledge about road segments. To learn the parameters of Dual-view Map Matcher for each task, we perform customized local updates via Initial Parameter Modulation and Learning Rate Scaling based on the hierarchical knowledge of spatial tasks. Incremental Dropout is further proposed to avoid overfitting during the meta-training phase. Extensive experiments as well as case studies on two real-world datasets show the effectiveness of SHSMM. We envision that the proposed framework also has the potential to address other spatial tasks, such as house price prediction, which remains a subject for future work.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2023YFC3321505), and the National Natural Science Foundation of China (No. 62306033, 42371480).

References

[1] Bang An, Xun Zhou, Amin Vahedian, Nick Street, Jinping Guan, and Jun Luo. 2024. LISA: Learning-Integrated Space Partitioning Framework for Traffic Accident Forecasting on Heterogeneous Spatiotemporal Data. In *2024 IEEE International Conference on Data Mining (ICDM)*. 11–20.

[2] Thanh-Hieu Bui. 2023. Automatic construction of POI address lists at city streets from geo-tagged photos and web data: a case study of San Jose City. *Multimedia Tools and Applications* 82, 22 (2023), 34749–34770.

[3] Yuqi Chen, Hanyuan Zhang, Weiwei Sun, and Baihua Zheng. 2023. Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 829–842.

[4] Miao Fan, Yibo Sun, Jizhou Huang, Haifeng Wang, and Ying Li. 2021. Meta-Learned Spatial-Temporal POI Auto-Completion for the Search Engine at Baidu Maps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2822–2830.

[5] Yu Fan, Jiajie Xu, Rui Zhou, Jianxin Li, Kai Zheng, Lu Chen, and Chengfei Liu. 2022. MetaER-TTE: An Adaptive Meta-learning Model for En Route Travel Time Estimation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2023–2029.

[6] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. 2021. SSML: Self-Supervised Meta-Learner for En Route Travel Time Estimation at Baidu Maps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2840–2848.

[7] Jie Feng, Yong Li, Kai Zhao, Zhao Xu, Tong Xia, Jinglin Zhang, and Depeng Jin. 2022. DeepMM: Deep Learning Based Map Matching With Data Augmentation. *IEEE Transactions on Mobile Computing* 21, 7 (2022), 2372–2384.

[8] Raphael A Finkel and Jon Louis Bentley. 1974. Quad trees a data structure for retrieval on composite keys. *Acta informatica* 4 (1974), 1–9.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[10] Yunchuan Guan, Yu Liu, Ke Zhou, and Junyuan Huang. 2023. Hierarchical Meta-Learning with Hyper-Tasks for Few-Shot Learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. New York, NY, USA, 587–596.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[12] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 9 (2021), 5149–5169.

[13] Amir Karami, Rachana Redd Kadari, Lekha Panati, Siva Prasad Nooli, Harshini Bheemreddy, and Parisa Bozorgi. 2021. Analysis of geotagging behavior: Do geotagged users represent the Twitter population? *ISPRS International Journal of Geo-Information* 10, 6 (2021), 373.

[14] Ilkcan Keles, Matthias Schubert, Peer Kröger, Simonas Šaltenis, and Christian S Jensen. 2017. Extracting visited points of interest from vehicle trajectories. In *Proceedings of the Fourth International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data*. 1–6.

[15] Zhexiong Liu, Licheng Liu, Yiqun Xie, Zhenong Jin, and Xiaowei Jia. 2023. Task-Adaptive Meta-Learning Framework for Advancing Spatial Generalizability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 14365–14373.

[16] Zhidan Liu, Hongquan Zhang, Guofeng Ouyang, Junyang Chen, and Kaishun Wu. 2024. Data-Driven Pick-Up Location Recommendation for Ride-Hailing Services. *IEEE Transactions on Mobile Computing* 23, 2 (2024), 1001–1015.

[17] Paul Newson and John Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. 336–343.

[18] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1720–1730.

[19] Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. 2021. Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1410–1419.

[20] Sijie Ruan, Xi Fu, Cheng Long, Zi Xiong, Jie Bao, Ruiyuan Li, Yiheng Chen, Shengnan Wu, and Yu Zheng. 2023. Filling Delivery Time Automatically Based on Couriers' Trajectories. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2023), 1528–1540.

[21] Sijie Ruan, Cheng Long, Zhipeng Ma, Jie Bao, Tianfu He, Ruiyuan Li, Yiheng Chen, Shengnan Wu, and Yu Zheng. 2022. Service time prediction for delivery tasks via spatial meta-learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3829–3837.

[22] Sijie Ruan, Zi Xiong, Cheng Long, Yiheng Chen, Jie Bao, Tianfu He, Ruiyuan Li, Shengnan Wu, Zhongyuan Jiang, and Yu Zheng. 2020. Doing in one go: delivery time inference based on couriers' trajectories. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2813–2821.

[23] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *The Thirteenth International Conference on Learning Representations*.

[24] Tian Sun, Yuqi Chen, Baihua Zheng, and Weiwei Sun. 2025. Learning Spatio-Temporal Dynamics for Trajectory Recovery via Time-Aware Transformer. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)* (2025).

[25] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.

[26] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[27] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems* 28 (2015).

[28] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. 2019. Multimodal Model-Agnostic Meta-Learning via Task-Aware Modulation. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.

[29] Shuliang Wang, Qianyu Yang, Sijie Ruan, Cheng Long, Ye Yuan, Qi Li, Ziqiang Yuan, Jie Bao, and Yu Zheng. 2025. Spatial Meta Learning With Comprehensive Prior Knowledge Injection for Service Time Prediction. *IEEE Transactions on Knowledge and Data Engineering* 37, 2 (2025), 936–950.

[30] Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, Jiyi Li, and Dongjin Yu. 2023. Eedn: Enhanced encoder-decoder network with local and global context learning for poi recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 383–392.

[31] Tonglong Wei, Youfang Lin, Yan Lin, Shengnan Guo, Lan Zhang, and Huaiyu Wan. 2024. Micro-Macro Spatial-Temporal Graph-Based Encoder-Decoder for Map-Constrained Trajectory Recovery. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[32] Haomin Wen, Youfang Lin, Huaiyu Wan, Shengnan Guo, Fan Wu, Lixia Wu, Chao Song, and Yinghui Xu. 2022. Deeproute+: Modeling couriers' spatial-temporal behaviors and decision preferences for package pick-up route prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 2 (2022), 1–23.

[33] Hao Wu, Weiwei Sun, and Baihua Zheng. 2016. Is only one gps position sufficient to locate you to the road network accurately?. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 740–751.

[34] Congxi Xiao, Jingbo Zhou, Jizhou Huang, Tong Xu, and Hui Xiong. 2023. Spatial heterophily aware graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2752–2763.

[35] Huaxiu Yao, Longkai Huang, Linjun Zhang, Ying Wei, Li Tian, James Zou, Junzhou Huang, and Zhenhui Li. 2021. Improving Generalization in Meta-learning via Task Augmentation. In *Proceeding of the Thirty-eighth International Conference on Machine Learning*.

[36] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*. 2181–2191.

[37] Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. 2019. Hierarchically Structured Meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. 7045–7054.

[38] Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. 2020. Automated Relational Meta-learning. In *International Conference on Learning Representations*.

[39] Huaxiu Yao, Linjun Zhang, and Chelsea Finn. 2022. Meta-Learning with Fewer Tasks through Task Interpolation. In *Proceeding of the 10th International Conference on Learning Representations*.

[40] Detian Zhang, Lun Jin, Chunjiang Zhu, and Qing Li. 2023. Efficient Optimal Pick-up and Drop-off Point Recommendation for Ride-hailing Services. In *2023 IEEE International Conference on Web Services (ICWS)*. IEEE, 124–130.

A Appendix

A.1 Pseudocode for meta-training of SHSMM

In this section, we show the pseudocode of the meta-training process in SHSMM. In Line 5, we use IPM to generate region specific initial parameter $\boldsymbol{\theta}_{r_i}^0$; while in Line 6, we use LRS to generate task-specific update learning rate α_i. In meta-training phase, α_i will be randomly dropout as shown in Line 7. Then we will get task-specific parameter $\boldsymbol{\theta}_i^K$ through K step local update as shown in Line 9-11.

Algorithm 1 Meta-training of SHSMM.

Input: Task set $\mathcal{T}$; geographical knowledge $\mathcal{K}^{Geo}$; semantic knowledge $\mathcal{K}^{Sem}$; the number of step for local update K; the initial local update learning rate α; global update learning rate β.
Output: The optimized parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{\mu}$ of SHSMM.
Initialization: $\boldsymbol{\theta}_0$ (local and global update) and $\boldsymbol{\mu}$ (global update only).

1: Epoch $n \leftarrow 0$;
2: **repeat**
3: **for** task $\mathcal{T}_i \in \mathcal{T}$ **do**
4: $\mathcal{D}_i^s$ and $\mathcal{D}_i^q$ are the corresponding support and query set;
5: $\boldsymbol{\theta}_{r_i}^0 \leftarrow IPM(\mathcal{K}_i^{Geo}, \boldsymbol{\theta}_0)$;
6: $\alpha_i \leftarrow LRS(\mathcal{K}_i^{Sem}, \alpha)$;
7: $\hat{\alpha}_i = [(1-\eta)^n]^t \alpha_i,\ t \sim \varepsilon(\lambda)$;
8: Let $k \leftarrow 0, \boldsymbol{\theta}_i^0 \leftarrow \boldsymbol{\theta}_{r_i}^0$;
9: **for** k < K **do**
10: $\boldsymbol{\theta}_i^{k+1} \leftarrow \boldsymbol{\theta}_i^k - \hat{\alpha}_i \nabla \mathcal{L}(\phi_{(\boldsymbol{\theta}_i^k, \boldsymbol{\mu})}, \mathcal{D}_i^s)$;
11: $k \leftarrow k+1$;
12: $(\boldsymbol{\theta}_0, \boldsymbol{\mu}) \leftarrow (\boldsymbol{\theta}_0, \boldsymbol{\mu}) - \beta \sum_{T_i \in \mathcal{T}} \nabla \mathcal{L}(\phi_{(\boldsymbol{\theta}_i^K, \boldsymbol{\mu})}, \mathcal{D}_i^q)$;
13: $n \leftarrow n+1$;
14: **until** *stopping criteria is met*
15: **return** $\boldsymbol{\theta}_0, \boldsymbol{\mu}$;

A.2 Baseline Descriptions

- **MinDist**, which simply matches the positioning point to the nearest road segment.
- **PSMM** [33], which is specifically designed to tackle the SMM problem, and matches the positioning point to nearby road segments that produce the highest matching probability produced by probabilistic models.
- **DeepMM** [7], which is the first deep-learning-based trajectory map matching model. It first discretes the positioning point into a location ID, uses an embedding layer to transform it into a hidden representation, and finally decodes it into a road segment by a softmax function.
- **RNTrajRec** [3], which proposes to use sub-graph to encode the surrounding road network environment of a positioning point, and uses a customized softmax layer constrained on distance to infer the matched road segment.
- **DTInf⁺** [20], which leverages a Transformer encoder to encode different candidate road segments, and an attention-based decoder to infer the matched one.
- **MAML** [9], which is a classic meta-learning method. It extracts the shared knowledge among tasks into the initial parameters of the base model.
- **HSML** [37], which hierarchically structures tasks based on task representation to tackle heterogeneous tasks that balance the generalization and customization ability.
- **MetaMix** [35], which linearly combines features and labels from both support and query sets.
- **MLTI** [39], which is a task augmentation method that interpolates features and labels between tasks to achieve data-adaptive meta-regularization.
- **Adaptive-MAML** [15], which hierarchically structures tasks based on the task adaptive performance.

A.3 Structure of Dual-view Map Matcher

The distance threshold to find candidate road segments D is set to $50m$. To represent the road segments, local edge IDs eid^l and global edge IDs eid^g are embedded into 3-dim and 16-dim vectors respectively. The transformer encoder is of 64-dim and one head.