MetaER-TTE: An Adaptive Meta-learning Model for En Route Travel Time Estimation

Yu Fan[1], **Jiajie Xu**[1*], **Rui Zhou**[2], **Jianxin Li**[3], **Kai Zheng**[4], **Lu Chen**[2], **Chengfei Liu**[2]
[1]School of Computer Science and Technology, Soochow University
[2]Swinburne University of Technology
[3]Deakin University
[4]University of Electronic Science and Technology of China
20205227013@stu.suda.edu.cn, xujj@suda.edu.cn, {rzhou, luchen,cliu}@swin.edu.au,
jianxin.li@deakin.edu.au, zhengkai@uestc.edu.cn

Abstract

En route travel time estimation (ER-TTE) aims to predict the travel time on the remaining route. Since the traveled and remaining parts of a trip usually have some common characteristics like driving speed, it is desirable to explore these characteristics for improved performance via effective adaptation. This yet faces the severe problem of data sparsity due to the few sampled points in a traveled partial trajectory. Since trajectories with different contextual information tend to have different characteristics, the existing meta-learning methods for ER-TTE cannot fit each trajectory well because it uses the same model for all trajectories. To this end, we propose a novel adaptive meta-learning model called MetaER-TTE. Particularly, we utilize soft-clustering and derive cluster-aware initialized parameters to better transfer the shared knowledge across trajectories with similar contextual information. In addition, we adopt a distribution-aware approach for adaptive learning rate optimization, so as to avoid task-overfitting which will occur when guiding the initial parameters with a fixed learning rate for tasks under imbalanced distribution. Finally, we conduct comprehensive experiments to demonstrate the superiority of MetaER-TTE.

1 Introduction

Travel time estimation (TTE) is a fundamental problem in many applications such as route planning [Xu *et al.*, 2019; Xu *et al.*, 2015], navigation [Kisialiou *et al.*, 2018] and vehicle dispatching [Yuan *et al.*, 2013]. Most of the existing TTE methods [Zhang *et al.*, 2018; Xu *et al.*, 2020; Fang *et al.*, 2020] mainly focus on pre-route travel time estimation (PR-TTE), which estimate the travel time of a given entire route.

Different from PR-TTE problem, en route travel time estimation (ER-TTE) is proposed to predict the travel time for the remaining route while driving. As we know, the traveled and remaining parts of a trajectory tend to have some common characteristics owing to the shared contextual information (e.g., departure time, weekdays and weather condition) and driver's emotion. For example, if a user drives fast in the traveled route, he is likely to drive at a high speed in the remaining route, probably because he has to catch a plane. Although PR-TTE methods can also support ER-TTE, they do not consider traveled routes, resulting in sub-optimal results. Therefore, we need to not only capture the useful characteristics from the traveled route, but also utilize these characteristics in the estimation accordingly. However, model adaptation inevitably faces the notorious cold-start problem due to the few sampled points in traveled routes.

Recently, meta-learning is known as one of the most successful approaches for cold-start problem [Finn *et al.*, 2017; Hospedales *et al.*, 2020], with a basic idea to learn the general knowledge through several tasks that can be rapidly adapted to new tasks. It has been leveraged in various domains such as computer vision [Ye *et al.*, 2020] and natural language processing [Madotto *et al.*, 2019], and proven to be successful due to its good generalization ability. Therefore, it provides great opportunities for ER-TTE as well. By considering each trajectory as a learning task, we can learn a more generalized model for ER-TTE with meta-learning first, then the generalized model can be rapidly adapted to estimate the travel time of remaining routes via fine tuning on traveled routes.

However, directly applying classical meta-learning framework [Mishra *et al.*, 2017; Finn *et al.*, 2017] in ER-TTE like state-of-the-art method SSML [Fang *et al.*, 2021] would incur inaccuracy due to two limitations. First, the same global parameters are utilized to guide the parameter initialization for all tasks. Unfortunately, using a globally shared parameter setting is unlikely to achieve better performance in ER-TTE, because trajectories with different contextual information tend to have different characteristics including the travel speed, which will severely affect the travel time. For example, the travel duration of a trajectory is usually much longer in rush hours than in off-peak hours. Therefore, it would be better to divide trajectories into several categories according to their contextual information, and capture the shared characteristics in each category respectively, i.e., the similar contextual information can be shared to engage the travel time

*Contact Author

estimation of trajectories within each category.

Second, these meta-learning methods [Finn *et al.*, 2017; Santoro *et al.*, 2016] adapt the initialized parameters to each task with a fixed learning rate, based on the assumption that tasks are distributed uniformly. However, in ER-TTE, task distribution is not always balanced, the uneven distribution of contextual information would result in task-overfitting [Yu *et al.*, 2021] during the adaptation. For example, the traffic flow is lower in midnight than in the daytime, trajectories collected in midnight are much fewer than those collected in the daytime, and they have different characteristics. Guiding the initialized parameters with a fixed learning rate tends to overfit the trajectories in the daytime, because these trajectories account for the majority and fitting them can achieve good average performance. It is nevertheless easy for these methods to ignore the trajectories collected in midnight, which may not be acceptable for real-world applications. Therefore, we need to adaptively guide the parameter initialization for different trajectories with different learning rates.

To address the above limitations, we propose a novel framework, namely MetaER-TTE, which not only alleviates the cold-start problem, but also supports personalized adaptation for each trajectory. Specifically, to better share the generalized characteristics among trajectories with similar contextual information, we adopt a cluster-enhanced parameter initialization method based on soft-clustering and a cluster-aware parameter memory. By deriving cluster-aware network parameters in the initialization step, we can guarantee that the performance will not be affected by the diversity of trajectories with different contextual information. Moreover, a learning rate generator is designed to set different learning rates for different trajectories, so as to avoid task-overfitting which may occur when guiding the parameter initialization with a fixed learning rate to tasks with uneven distribution. The contributions of this paper can be summarized as follows:

- We propose an adaptive meta-learning method for ER-TTE, which supports personalized adaptation to each trajectory for more accurate estimation.
- We adopt a soft-clustering method to derive the cluster-aware initialized parameters, in order to better transfer the shared knowledge across trajectories with similar contextual information.
- A learning-rate generator is further designed to adaptively guide the global initial parameters for each trajectory with a reasonable distribution-aware learning rate to prevent task-overfitting.
- We conduct extensive experiments on two real-world datasets to demonstrate the effectiveness of our method.

2 Related Work

2.1 Travel Time Estimation

TTE is one of the key topics in transportation systems. These years, large datasets and computational power enable the success of deep learning. Some recent studies attempt to solve the TTE problem by deep learning.

DeepTTE [Wang *et al.*, 2018] utilizes a geo-convolution to split the whole path with intermediate GPS points into several sub-paths, then predicts the travel time of each sub-path and the whole path by a multi-task loss function. ConST-GAT [Fang *et al.*, 2020] leverages a spatial-temporal graph attention network to fully exploit the joint relations of spatial and temporal information to improve the performance of TTE. CompactETA [Fu *et al.*, 2020] considers road network constraints, and applies graph attention network to learn the spatio-temporal dependencies as well as positional encoding to encode the sequential information of the path to infer the travel time. TADNM[Xu *et al.*, 2020] divides the path into several segments according to their transportation modes, and capture the spatial-temporal correlations for each segment to provide accurate TTE for mixed-mode paths.

Although the above methods can well support PR-TTE, they can only achieve suboptimal results in ER-TTR due to the failure of considering traveled part of trajectories.

2.2 Meta-learning

Meta-learning is a learning paradigm which aims to learn the general knowledge across a variety of different tasks to rapidly adapt to new tasks with little training data. Recent meta-learning algorithms can be divided into three categories: model-based [Munkhdalai *et al.*, 2018; Santoro *et al.*, 2016], metric-based [Snell *et al.*, 2017; Sung *et al.*, 2018] and optimization-based [Andrychowicz *et al.*, 2016; Finn *et al.*, 2017] meta-learning.

State-of-the-art work SSML [Fang *et al.*, 2021] is a model-based meta-learning methods for ER-TTE, which aims to learn the meta-knowledge from the traveled partial trajectories to estimate the travel time of the remaining routes. However, it cannot adapt to each trajectory well due to the lack of adaptation. MAML [Finn *et al.*, 2017] is one of the most successful optimization-based meta-learning for learning a good initialization. On the basis of MAML, some approaches have attempted to fit each task adaptively. PAML [Yu *et al.*, 2021] is also for recommendation that finds similar users as a reference to provide better personalized learning rates. HSML [Yao *et al.*, 2019] divides different tasks into several categories and promote knowledge customization to different clusters to enhance the effectiveness.

Inspired by the above works, we propose an adaptive meta-learning method for ER-TTR based on MAML, which provides personalized initial parameters and learning rates for trajectories with different contextual information.

3 Preliminary

We define a trajectory t as a sequence of road segments, i.e., $t = \{r_1, r_2, ..., r_n\}$, where r_i is the i-th road segment in this trajectory. Furthermore, we record contextual information such as the departure time, the day of the week and the weather condition for each trajectory. Then we divide the entire trajectory into two parts. The former is the traveled route which has already been collected during the travel, represented as $t_{tr} = \{r_1, r_2, ..., r_m\}$, $m < n$; the later is the remaining route denoted as $t_{re} = \{r_{m+1}, r_{m+2}, ..., r_n\}$. The goal of ER-TTE task is to estimate the travel time of remaining routes by making use of traveled routes.

For the meta-learning setting, each trajectory is viewed as a learning task. We divide the trajectories into a training set $\mathcal{T}^{train}$ and a testing set $\mathcal{T}^{test}$. For each trajectory t, in order to make full use of temporal labels in the traveled route, we generate sub-trajectories $\{r_1, ..., r_{m*20\%}\}$, $\{r_1, ..., r_{m*40\%}\}, ...,$ and $\{r_1, ..., r_m\}$ forming a support set $\mathcal{D}^s$, each sub-trajectory in support set has a travel time, and take the remaining route as a query set $\mathcal{D}^q$.

4 MetaER-TTE

In this section, we introduce our proposed method: an adaptive meta-learning model for ER-TTE, namely MetaER-TTE shown in Figure 1. First, we introduce the base model named ConSTGAT [Fang *et al.*, 2020]. It is a state-of-the-art TTE method that integrates relations of road segments and traffic prediction to estimate the travel time. Second, we present the details of the MetaER-TTE model, which support effective adaptation to each trajectory to achieve better performance in ER-TTE.

4.1 Base Model

We adopt a PR-TTE method ConSTGAT [Fang *et al.*, 2020] as our base model, since it is an existing work, we briefly introduce it in this section. For each road segment r_i, it learns the representation of contextual information X_i^{CI} (e.g., departure time, weekdays and weather condition), predicts the traffic condition X_i^{TC} and captures the spatial correlations X_i^{SC} (i.e., the spatial relationship of r_i and its adjacent road segments). Then it combines these information to estimate the travel time of r_i:

$$\widehat{y_i} = FC_{\theta^{est}}((X_i^{CI} \oplus X_i^{TC} \oplus X_i^{SC})) \tag{1}$$

where $\theta^{est} \in \mathbf{R}^{d_{est}}$ denotes the parameters of the estimation layer, and $\oplus$ means the tensors concatenation. The predicted travel times of the road segments are summed up to obtain the predicted travel time of the entire route. In particular, we denote the parameters of network layers that obtain X_i^{CI}, X_i^{TC} and X_i^{SC} as θ^*, so the parameters of the base model can be represented as $\theta = \{\theta^*, \theta^{est}\}$.

Finally, ConSTGAT calculates the loss for road segment L_{r_i} and the entire route L_{t_j} with Huber loss and absolute percentage error (APE) respectively, and combines them to obtain the joint loss:

$$L_{joint} = \frac{1}{h} \sum_{j=1}^{h} (\frac{1}{n^{(j)}} \sum_{i=1}^{n^{(j)}} L_{r_i} + L_{t_j}) \tag{2}$$

where h is the number of entire routes and $n^{(j)}$ is the number of road segments in t_j.

4.2 Meta Optimization

Next, we elaborate an adaptive meta-learning framework with parameters ϕ^* for ER-TTE named MetaER-TTE, which supports the effective adaptation to each trajectory. The framework is composed of three components: task-clustering, a cluster-aware parameter memory and a learning rate generator. We cluster trajectories into several categories according to their contextual information, and then derive the cluster-aware initialized parameters for personalized estimation with a cluster-aware parameter memory. Moreover, the learning rate generator provides different learning rates for different trajectories to adaptively guide the generalized knowledge to each trajectory.

Task-clustering

In order to better transfer the shared knowledge among trajectories with similar contextual information, we utilize a soft-clustering method to divide the training trajectories into several categories according to their contextual information. Here we adopt soft-clustering method instead of hard-clustering or classification methods because trajectories with different contextual information may share the knowledge due to the rapidly changing traffic condition. For example, the travel speed in off-peak hours may be as slow as in rush hours affected by a sudden traffic accident. Moreover, soft-clustering can guarantee differentiability, and ensures that relevant knowledge can be attentively learned from trajectories in different categories.

First, we conduct a cluster assignment to each cluster for each trajectory. Specifically, we project the contextual information $X^{CI} \in \mathbf{R}^{d_{ci}}$ of trajectory t_j to get the query vector $q_j \in \mathbf{R}^{d_q}$, represented as:

$$q_j = W_q(X^{CI}) + b_q \tag{3}$$

where $W_q \in \mathbf{R}^{d_{ci} \times d_q}$ and $b_q \in \mathbf{R}^{d_q}$ are learned parameters. Then we use the query vector to calculate the similarity score s_j^k between it and each learned cluster center $\{g_k\}_{k=1}^K$ as:

$$s_j^k = \frac{exp(\langle q_j, g_k \rangle)}{\sum_{k=1}^{K} exp(\langle q_j, g_k \rangle)} \tag{4}$$

where K denotes the number of clusters, the determination of cluster numbers will be discussed in experiments. The learned similarity score will be used to derive cluster-aware initialized parameters for personalized travel time estimation, and then obtain cluster enhanced representation for each trajectory to generate a more reasonable learning rate. Before training, we randomly initialize each cluster center, and update the cluster centers during the training process.

Cluster-aware Parameter Memory

Moreover, considering that trajectories with different contextual information (e.g., departure time, weekdays, weather condition) have different characteristics, it is irrational to utilize the same global parameters to estimate the travel time for all the remaining partial trajectories, because there exists impact from the diversity of different trajectories. Therefore, a cluster-aware parameter memory $M_P \in \mathbf{R}^{K \times d_q \times d_{est}}$ is designed to store the parameters of the estimation layer for different clusters. It aims to provide personalized initialized parameters for each trajectory according to the similarity score s_j^k calculated by Eq. (4).

Following the idea of Neural Turing Machine [Graves *et al.*, 2014], the memory cube M_P has a read head to retrieve the memory and a write head to update the memory. Specifically, we retrieve the parameter matrix $M_{j,P} \in \mathbf{R}^{d_q \times d_{est}}$ for

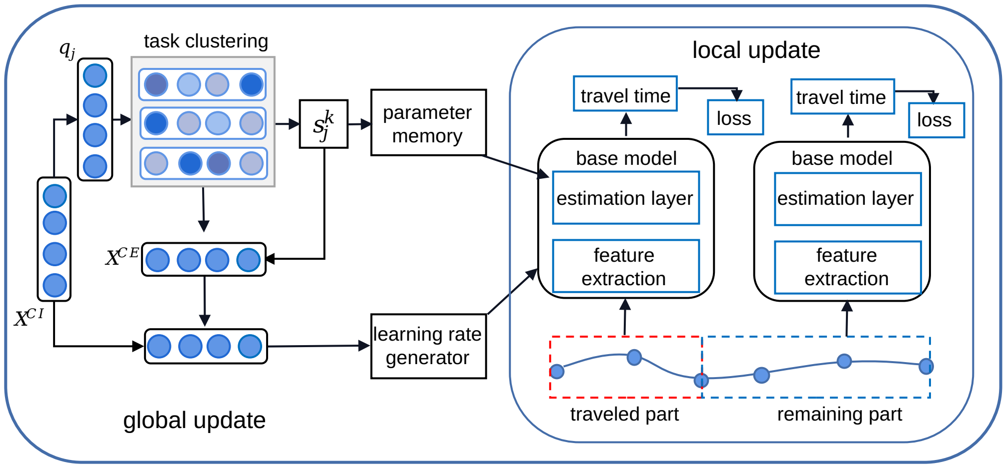

Figure 1: the architecture of MetaER-TTE. Each trajectory is viewed as a learning task, the goal is to estimate the travel time of remaining routes by utilizing the traveled route. The task-clustering and the cluster-aware parameter memory are designed to derive cluster-aware initialized parameters for each cluster. The learning rate generator combines the representation of contextual information and cluster-enhanced representation to provide personalized learning rate for each trajectory to avoid task-overfitting.

trajectory t_j:

$$M_{j,P} = s_j^k \cdot M_P \tag{5}$$

$M_{j,P}$ is served as the personalized initial parameters of the estimation layer to provide more accurate estimation, and will be updated locally during the training process as:

$$M_P = \alpha \cdot (s_j^k \otimes M_{j,P}) + (1-\alpha)M_P \tag{6}$$

where $\otimes$ denotes the tensor product, and α is a hyper-parameter to control how much new parameter information is added to the memory. In particular, the similarity score s_j^k is used here to ensure that the new information will be attentively added to the memory.

Learning Rate Generator

The learning rate in local update determines whether the initialized parameters can be moved to the optimal value for each task, simply using the same learning rate like previous meta-learning methods [Finn *et al.*, 2017] will result in task-overfitting in ER-TTE due to the uneven distribution of trajectory contextual information.

In ER-TTE, the trajectory contextual information is not always balanced. For example, the traffic flow is lower in midnight than in daytime, thus trajectories collected in the daytime are more than those collected in midnight. Adapting with the fixed learning rate will overfit trajectories in daytime, and fail to achieve optimal parameters for other trajectories in midnight, because trajectories in daytime account for the majority and fitting them can achieve good average performance. To this end, we propose a learning rate generator to provide distribution-aware learning rate for each trajectory.

Since the learning rates are related to the contextual information distribution, trajectories with similar contextual information are likely to share similar learning rates. Directly using only contextual information of trajectory itself is not sufficient to generate a reasonable learning rate, it is better to take generalization among similar contextual information into consideration as well. Therefore, we make full use of clusters to obtain a cluster enhanced representation which contains the shared contextual information in the same cluster. The cluster enhanced representation X^{CE} of trajectory t_j is:

$$X^{CE} = \sum_{k=1}^{K} s_j^k \cdot g_k \tag{7}$$

where $\cdot$ is multiplication. Next we combine the contextual information x_j^{CI} and the cluster enhanced representation X^{CE} as a reference to obtain an adaptive learning rate. Now, we can get a learning rate function for a trajectory:

$$lr_j = FC_\tau(X^{CI} \oplus X^{CE}) \tag{8}$$

where FC_τ is a fully-connected layer with the parameter τ which will be trained in global update. The distribution-aware learning rate is used to guide the initialized parameters so as to find task-adaptive parameters.

Local Update

In traditional model training, the parameters of a neural network are initialized and converge to a good local optimum based on a large number of training data. Similarly, the optimization goal in local training is to update the local parameters for each trajectory by minimizing the loss function based on the support set. Thus the local parameters for task t_j will be updated as follows:

$$\theta_j^* \leftarrow \theta_j^* - lr_j \cdot \nabla_{\theta_j} L_{joint}^{\mathcal{D}^s} \tag{9}$$

$$\theta_j^{est} \leftarrow \theta_j^{est} - lr_j \cdot \nabla_{\theta_j} L_{joint}^{\mathcal{D}^s} \tag{10}$$

where θ_j^* and θ_j^{est} are initialized from the global parameters ϕ^* and $M_{j,P}$ respectively, L_{joint} is the joint loss calculated in Eq. (2), lr_j is the personalized learning rate for t_j, calculated in Eq. (8).

Global Update

In meta optimization process, we aim to minimize the loss function on the query set. All the parameters needed to be updated globally are represented as Θ, including the shared initial parameters ϕ^*, the parameters for clustering and those for knowledge adaptation such as τ. Similar to MAML [Finn *et al.*, 2017], we take one-step gradient decent to update the global parameters Θ as follows:

$$\Theta \leftarrow \Theta - \gamma \sum_{t \in \mathcal{T}^{train}} \nabla_\Theta L_{joint}^{\mathcal{D}^q} \tag{11}$$

where γ is the fixed learning rate for updating the initialization of global parameters. And the L_{joint} is the joint loss on the query set. Both local update and global update are computed via back-propagation. Meanwhile, the cluster-aware parameter memory M_P is updated by Eq. (6).

5 Experiment

5.1 Datasets

We perform our experiments on two real trajectory datasets, Beijing and Porto. Trajectories in Beijing dataset were collected from May 1st to May 18th, 2016. There are 1041584 trajectories covering the road network. Porto dataset is publicly available with 737063 trajectories generated from Jul 1st, 2013 to Jul 1st, 2014. We map the GPS trajectories to the road network to get the corresponding sequence of road segments for two datasets respectively. Then we remove the noisy records which have extremely small travel time (i.e., $< 120s$) or those with very few road segments (i.e., < 10). For Beijing dataset, 80% of the trajectories are used to train the model and the remaining 20% are used to test. For Porto dataset, we select the last two months for testing the model and the remaining ten months for training. We choose hyperparameters by conducting meta-train using different hyperparameters and select the parameters with the best performance, thus we do not set the validation set.

5.2 Evaluation Metrics and Configuration

Three metrics are used to evaluate the performance of our methods, including mean absolute percentage error (MAPE), mean average error (MAE), and root mean square error (RMSE), which are widely used in regression problems. The time slot is set to 30 minutes to avoid the absence of historical traffic conditions due to the sparse data. Other settings of our base model are the same as ConstGAT [Fang *et al.*, 2020]. For each trajectory, we split it into 30% as the traveled route and 70% as the remaining route, the average numbers of the road segments in the traveled and remaining parts of trajectories are 15, 34 in Beijing dataset, and 14, 31 in Porto dataset. The number of clusters is set as 3 for two datasets according to the comparisons of different cluster numbers in Figure 2. The initial learning rate of global update is set as 0.0001.

5.3 Baselines

First, in order to verify that we choose the strongest PR-TTE method as our base model, we estimate the travel time of the remaining route with existing PR-TTE methods. Second, to confirm the importance of the traveled route, as well as the effectiveness of our model for solving the cold-start problem in ER-TTE, we compare our model with methods that deal with cold-start problems, e.g., TransferTTE, MAML and SSML, in these methods, we estimate the travel time of remaining routes via fine-tuning on traveled routes. For fair comparison, the same base model ConSTGAT is used in TransferTTE, MAML, SSML and our model MetaER-TTE,

- **DeepTTE** [Wang *et al.*, 2018] transforms the raw GPS trajectory to a series of local-paths, and captures spatial-temporal dependencies based on these local-paths to estimate the travel time.
- **CompactETA** [Fu *et al.*, 2020] considers road network constraints and spatial-temporal dependencies to infer the travel time.
- **ConstGAT** [Fang *et al.*, 2020] is selected as our base model, it integrates the relations of road segments and traffic prediction to estimate the travel time.
- **TransferTTE** is based on transfer learning, which trains the base model ConstGAT on the training set and fine-tunes the model on traveled routes for testing.
- **MAML** [Finn *et al.*, 2017] is a classic meta-learning method, which aims to learn a general initialization from multiple tasks and adapt it to new tasks.
- **SSML** [Fang *et al.*, 2021] is a state-of-the-art meta-learning model for ER-TTE, it aims to learn the meta-knowledge to fast adapt to a user's driving preference.

5.4 Performance Comparison

We ran three times of each method and take the average values as the final results. Table 1 illustrates the experimental results of all methods on two datasets. The improvements are calculated as the difference of MetaER-TTE and the best method (underlined) over the best method, shown in percentage. For MAPE, MAE, RMSE, smaller values indicate better performances, thus the values of improvements are negative.

Base Model Comparison. The framework of meta-learning is based on a base model. We need to select a base model which is relatively good, because it can increase the bottom-line performance. As is shown in Table 1, the ConSTGAT model which fully exploits the joint relations of spatial and temporal information, has the best performance. Therefore, we choose ConSTGAT as the base model.

Meta-learning Strategy Comparison. Next, we evaluate the effectiveness of meta-learning strategies for ER-TTE. As is shown in Table 1, we can observe that the performance of TransferTTE is better than all the PR-TTE methods, indicating that learning from the traveled route can improve the accuracy of the estimation for the remaining route. Then the meta-learning based models (MAML, SSML, MetaER-TTE) outperform TransferTTE on both datasets, revealing that meta-learning can further improve the performance by alleviating the cold-start problem in ER-TTE. SSML performs worse than our method, because it cannot adapt to each trajectory well. The experimental results demonstrate the superiority of our model, which can be explained as that MetaER-

Dataset	Beijing			Porto		
Metrics	MAPE (%)	MAE (s)	RMSE (s)	MAPE (%)	MAE (s)	RMSE (s)
deepTTE	32.77	160.80	237.67	19.65	119.52	183.59
CompactETA	30.65	154.02	217.58	19.37	115.82	174.25
ConstGAT	27.02	136.35	194.95	18.68	109.79	168.39
TransferTTE	26.77	135.37	194.15	18.61	109.56	168.17
MAML	26.70	133.86	193.35	17.80	102.02	158.76
SSML	<u>25.38</u>	<u>127.21</u>	<u>188.11</u>	<u>17.68</u>	<u>98.63</u>	<u>150.96</u>
MetaER-TTE-H	25.01	123.41	179.98	17.74	99.65	154.84
MetaER-TTE-P	25.52	127.85	180.69	17.44	96.30	149.22
MetaER-TTE-L	26.56	132.67	191.79	17.58	98.35	150.82
MetaER-TTE	**24.21**	**122.27**	**176.09**	**17.37**	**93.75**	**145.39**
Improvement	-4.60%	-3.88%	-6.38%	-1.75%	-4.94%	-3.68%

Table 1: Performance comparison of MetaER-TTE and its competitors

Datasets	Beijing			Porto		
Metrics	MAPE (%)	MAE (s)	RMSE (s)	MAPE (%)	MAE (s)	RMSE (s)
K=1	26.70	133.86	193.35	17.80	102.02	158.76
K=2	24.65	125.01	179.69	17.71	98.61	152.14
K=3	**24.21**	**123.27**	**178.09**	**17.56**	**96.94**	**150.74**
K=4	24.69	124.11	178.78	17.70	98.15	151.27
K=5	24.67	124.47	179.07	17.65	98.04	151.25
K=6	24.71	124.94	179.90	17.70	98.10	151.34
Improvement	-9.33%	-7.91%	-7.89%	-1.35%	-4.98%	-5.05%

Table 2: Performance of different cluster numbers

TTE can not only alleviate the cold-start problem in ER-TTE, but also support personalized adaptation for each trajectory.

Comparison with Variants. In this part, we aim to evaluate the usefulness of each designed component in MetaER-TTE, we compare the MetaER-TTE with several variants:

- **MetaER-TTE-H** uses hard-cluster instead of soft-cluster method. The numbers of clusters are set to 3 as the same as our model.
- **MetaER-TTE-P** removes the cluster-aware parameter memory and transfer the shared characteristics across all trajectories, even they may have completely different contextual information.
- **MetaER-TTE-L** removes the learning rate generator in this variant and adapt the initialization to each trajectory with a fixed learning rate that is set to 0.00001.

First, the soft-clustering method is superior than hard-clustering, indicating that trajectories with different contextual information may share the knowledge due to the dynamic traffic condition. Second, from the result of MetaER-TTE-P and MetaER-TTE-L, we can conclude that removing either cluster-aware parameter memory or learning rate generator would affect the estimation.

Influence of the Cluster Numbers. The improvements in Table 2 are computed as the difference of the best performances (K=3) and performances with 1 cluster over the performances with 1 cluster on that metric, shown in percentage. As is shown in Table 2, the best performances are achieved at K=3. That is because departure time has dominated the clustering. When K=3, we have approximate morning peak, evening peak and off-peak clusters. Therefore, we set the cluster number as 3.

6 Conclusion

In this paper, we propose a novel adaptive meta-learning method for ER-TTE called MetaER-TTE. Specifically, we adopt a soft-clustering method and a cluster-aware parameter memory to derive cluster-aware network parameters in the initialization step, so as to better transfer the shared characteristics across trajectories with similar contextual information. Moreover, in order to prevent task-overfittingwe design a learning rate generator to guide the initialized parameters for each trajectory with a reasonable learning rate. Finally, we conduct extensive experiments on two real-world datasets to verify the effectiveness of our proposed model.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 61872258, 62072125, 62073255, 61972069, 61836007 and 61832017), ARC Discovery Projects (Grant No. DP170104747, DP200103700), the ARC Linkage Project (Grant No. LP18010075), the major project of natural science research in universities of Jiangsu Province under grant number 20KJA520005, the priority academic program development of Jiangsu higher education institutions, young scholar program of Cyrus Tang Foundation, and Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant No. SJCX22_1498.

References

[Andrychowicz *et al.*, 2016] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, pages 3981–3989, 2016.

[Fang *et al.*, 2020] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps. In *KDD*, pages 2697–2705, 2020.

[Fang *et al.*, 2021] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. SSML: self-supervised meta-learner for en route travel time estimation at baidu maps. In *KDD*, pages 2840–2848, 2021.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, volume 70, pages 1126–1135, 2017.

[Fu *et al.*, 2020] Kun Fu, Fanlin Meng, Jieping Ye, and Zheng Wang. Compacteta: A fast inference system for travel time prediction. In *KDD*, pages 3337–3345, 2020.

[Graves *et al.*, 2014] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, 2014.

[Hospedales *et al.*, 2020] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[Kisialiou *et al.*, 2018] Yauheni Kisialiou, Irina Gribkovskaia, and Gilbert Laporte. The periodic supply vessel planning problem with flexible departure times and coupled vessels. *COR*, 94:52–64, 2018.

[Madotto *et al.*, 2019] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. In *ACL*, pages 5454–5459, 2019.

[Mishra *et al.*, 2017] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[Munkhdalai *et al.*, 2018] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *ICML*, pages 3664–3673, 2018.

[Santoro *et al.*, 2016] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, volume 48, pages 1842–1850, 2016.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018.

[Wang *et al.*, 2018] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. When will you arrive? estimating travel time based on deep neural networks. In *AAAI*, pages 2500–2507, 2018.

[Xu *et al.*, 2015] Jiajie Xu, Yunjun Gao, Chengfei Liu, Lei Zhao, and Zhiming Ding. Efficient route search on hierarchical dynamic road networks. *DPD*, 33(2):227–252, 2015.

[Xu *et al.*, 2019] Jiajie Xu, Jing Chen, Rui Zhou, Junhua Fang, and Chengfei Liu. On workflow aware location-based service composition for personal trip planning. *FGCS*, 98:274–285, 2019.

[Xu *et al.*, 2020] Saijun Xu, Jiajie Xu, Rui Zhou, Chengfei Liu, Zhixu Li, and An Liu. Tadnm: A transportation-mode aware deep neural model for travel time estimation. In *DSFAA*, pages 468–484, 2020.

[Yao *et al.*, 2019] Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 7045–7054, 2019.

[Ye *et al.*, 2020] Han-Jia Ye, Xiang-Rong Sheng, and De-Chuan Zhan. Few-shot learning with adaptively initialized task optimizer: a practical meta-learning approach. *ML*, 109(3):643–664, 2020.

[Yu *et al.*, 2021] Runsheng Yu, Yu Gong, Xu He, Yu Zhu, Qingwen Liu, Wenwu Ou, and Bo An. Personalized adaptive meta learning for cold-start user preference prediction. In *AAAI*, pages 10772–10780, 2021.

[Yuan *et al.*, 2013] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, and Xing Xie. T-finder: A recommender system for finding passengers and vacant taxis. *TKDE*, 25(10):2390–2403, 2013.

[Zhang *et al.*, 2018] Hanyuan Zhang, Hao Wu, Weiwei Sun, and Baihua Zheng. Deeptravel: a neural network based travel time estimation model with auxiliary supervision. In *IJCAI*, pages 3655–3661, 2018.