Learning from Multiple Cities: A Meta-Learning Approach for Spatial-Temporal Prediction

Huaxiu Yao
Pennsylvania State University
huaxiuyao@psu.edu

Yiding Liu
Nanyang Technological University
liuy0130@e.ntu.edu.sg

Ying Wei
Tencent AI Lab
judyweiying@gmail.com

Xianfeng Tang
Pennsylvania State University
tangxianfeng@outlook.com

Zhenhui Li
Pennsylvania State University
jessieli@ist.psu.edu

ABSTRACT

Spatial-temporal prediction is a fundamental problem for constructing smart city, which is useful for tasks such as traffic control, taxi dispatching, and environment policy making. Due to data collection mechanism, it is common to see data collection with unbalanced spatial distributions. For example, some cities may release taxi data for multiple years while others only release a few days of data; some regions may have constant water quality data monitored by sensors whereas some regions only have a small collection of water samples. In this paper, we tackle the problem of spatial-temporal prediction for the cities with only a short period of data collection. We aim to utilize the long-period data from other cities via transfer learning. Different from previous studies that transfer knowledge from one single source city to a target city, we are the first to leverage information from multiple cities to increase the stability of transfer. Specifically, our proposed model is designed as a spatial-temporal network with a meta-learning paradigm. The meta-learning paradigm learns a well-generalized initialization of the spatial-temporal network, which can be effectively adapted to target cities. In addition, a pattern-based spatial-temporal memory is designed to distill long-term temporal information (i.e., periodicity). We conduct extensive experiments on two tasks: traffic (taxi and bike) prediction and water quality prediction. The experiments demonstrate the effectiveness of our proposed model over several competitive baseline models.

CCS CONCEPTS

• **Information systems → Data Mining**; • **Computing methodologies → Transfer learning**.

KEYWORDS

Spatial-temporal prediction, periodicity, meta-learning

ACM Reference Format:
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 *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313577

1 INTRODUCTION

Recently, the construction of smart cities significantly changes urban management and services [12, 13, 25, 28, 42, 43]. One of the most fundamental techniques in building smart city is accurate spatial-temporal prediction. For example, a traffic prediction system can help the city pre-allocate transportation resources and control traffic signal intelligently. An accurate environment prediction system can help the government develop environment policy and then improve the public's health.

Traditionally, basic time series models (e.g., ARIMA, Kalman Filtering and their variants) [10, 15, 19], regression models with spatial-temporal regularizations [6, 44] and external context features [30, 35] are used for spatial-temporal prediction. Recently, advanced machine learning methods (e.g., deep neural network based models) significantly improve the performance of spatial-temporal prediction [34, 35, 41] by characterizing non-linear spatial-temporal correlations more accurately. Unfortunately, the superior performance of these models is conditioned on large-scale training data which are probably inaccessible in real-world applications. For example, there may exist only a few days of GPS traces for traffic prediction in some cities.

Transfer learning has been studied as an effective solution to address the data insufficiency problem, by leveraging knowledge from those cities with abundant data (e.g., GPS traces covering a few years). In [29], the authors proposed to transfer semantically-related dictionaries learned from a data-rich city, i.e., a source city, to predict the air quality category in a data-insufficient city, i.e., a target city. The method proposed in [27] aligns similar regions across source and target cities for finer-grained transfer. However, these methods, transferring the knowledge from only a single source city, would cause unstable results and the risk of negative transfer. If the underlying data distributions are significantly different between cities, the knowledge transfer will make no contribution or even hurt the performance.

To reduce the risk, in this paper, we focus on transferring knowledge from *multiple source cities* for the spatial-temporal prediction in a target city. Compared with single city, the knowledge extracted from multiple cities covers more comprehensive spatial-temporal correlations of a city, e.g., temporal dependency, spatial closeness, and region functionality, and thus increases the stability of transfer. However, this problem faces two key challenges.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313577

- **How to adapt the knowledge to meet various scenarios of spatial-temporal correlations in a target city?** The spatial-temporal correlations in the limited data of a target city may vary considerably from city to city and even from time to time. For example, the knowledge to be transferred to New York is expected to differ from that to Boston. In addition, the knowledge to be transferred within the same city also differs from weekdays to weekends. Thus a sufficiently flexible algorithm capable of adapting the knowledge learned from multiple source cities to various scenarios is required.
- **How to capture and borrow long-period spatial-temporal patterns from source cities?** It is difficult to capture long-term spatial-temporal patterns (e.g., periodicity) accurately in a target city with limited data due to the effects of special events (e.g. holiday) or missing values. These patterns as indicators of region functionality, however, are crucial for spatial-temporal prediction [33, 45]. Take the traffic demand prediction as an instance. The traffic demand in residential areas is periodically high in the morning when people transit to work. Thus, it is promising but challenging to transfer such long-term patterns from source cities to a target city.

To address the challenges, we propose a novel framework for spatial-temporal prediction, namely **MetaST**. It is the first to incorporate the meta-learning paradigm into spatial-temporal networks (ST-net). The ST-net consists of a local CNN and an LSTM which jointly capture spatial-temporal features and correlations. With the meta-learning paradigm, we solve the first challenge by learning a well-generalized initialization of the ST-net from a large number of prediction tasks sampled from multiple source cities, which covers comprehensive spatial-temporal scenarios. Subsequently, the initialization can easily be adapted to a target city via fine-tuning, even when only few training samples are accessible. Second, we learn a global pattern-based spatial-temporal memory from all source cities, and transfer it to a target city to support long-term patterns. The memory, describing and storing long-term spatial-temporal patterns, is jointly trained with the ST-net in an end-to-end manner.

We evaluate the proposed framework on several datasets including taxi volume, bike volume, and water quality. The results show that our proposed MetaST consistently outperforms several baseline models. We summarize our contributions as follows.

- To the best of our knowledge, we are the first to study the problem of transferring knowledge from multiple cities for the spatial-temporal prediction in a target city.
- We propose a novel MetaST framework to solve the problem by combining a spatial-temporal network with the meta-learning paradigm. Moreover, we learn from all source cities a global memory encrypting long-term spatial-temporal patterns, and transfer it to further improve the spatial-temporal prediction in a target city.
- We empirically demonstrate the effectiveness of our proposed MetaST framework on three real-world spatial-temporal datasets.

The rest of this paper is organized as follows. We first review and discuss the related work in Section 2. Then, we define some notations and formulate the problem in Section 3. After that, we introduce details of the framework of MetaST in Section 4. We apply our model on three real-world datasets from two different domains and conduct extensive experiments in Section 5. Finally, we conclude our paper in Section 6.

2 RELATED WORK

In this section, we briefly review the works in two categories: some representative works for spatial-temporal prediction and knowledge transferring.

2.1 Spatial-Temporal Prediction

The earliest spatial-temporal prediction models are based on basic time series models (e.g., ARIMA, Kalman Filtering) [10, 15, 19]. Recent studies further utilize external context data (e.g., weather condition, venue types, holiday, event information) [26, 30, 35] to enhance the prediction accuracy. Also, spatial correlation is taken into consideration by designing regularization of spatial smoothness [21, 31, 44].

Recently, various deep learning methods have been used to capture complex non-linear spatial-temporal correlations and predict spatial-temporal series, such as stacked fully connected network [24, 35], convolutional neural network (CNN) [23, 41] and recurrent neural network (RNN) [39]. Several hybrid models have been proposed to model both spatial and temporal information [7, 33, 34]. These methods combine CNN and RNN, and achieve the state-of-the-art performance on spatial-temporal prediction. In addition, based on the road network structure, another type of hybird models combines graph aggregation mechanism and RNN for spatial-temporal prediction [9, 38, 40]

Different from previous studies of deep spatial-temporal prediction which all rely on a large set of training samples, we aim to transfer learned knowledge from source cities to improve the performance of spatial-temporal prediction in a target city with limited data samples.

2.2 Knowledge Transfer and Reuse

Transfer learning and its related fields utilize previously learned knowledge to facilitate learning in new tasks when labeled data is scarce [16, 17]. Previous transfer learning methods transfer different information from a source domain to a target domain, such as parameters [32], instances [1], manifold structures [3, 4], deep hidden feature representations [22, 37]. Recently, meta-learning (a.k.a., learning to learn) transfers shared knowledge from multiple training tasks to a new task for quick adaptation. These techniques include learning a widely generalizable initialization [2, 11], optimization trace [18], metric space [20], transfer learning skills [36].

However, only a few attempts have been made on transferring knowledge on the space. [29] proposes a multi-modal transfer learning framework for predicting air quality category, which combines multi-modal data by learning a semantically coupled dictionary for multiple modalities in a source city. *However, this method works on multimodal features instead of spatial-temporal sequences we focus on. Therefore, it cannot be applied to solve the problem.* For predicting traffic flow, [27] leverages the similarity of check-in records/spatial-temporal sequences between a source city and a target city to construct the similarity regularization for knowledge transfer. *Different from this method that intelligently learns the correlation which could be linear or non-linear. Compared with both* *methods above, in addition, our model transfers the shared knowledge from multiple cities to a new city, which increase the stability of transfer and prediction.*

3 DEFINITIONS AND PROBLEM FORMULATION

In this section, we define some notations used in this paper and formally define the setup of our problem.

Definition 1 (Region) Following previous works [34, 41], we spatially divide a city c into an $I_c \times J_c$ grid map which contains I_c rows and J_c columns. We treat each grid as a region r_c, and define the set of all regions as $\mathcal{R}_c$.

Definition 2 (Spatial-Temporal Series) In city c, we denote the current/latest timestamp as k_c, and the time range as a set $\mathcal{K}_c = \{k_c - |\mathcal{K}_c| + 1, ..., k_c\}$ consisting of $|\mathcal{K}_c|$ evenly split non-overlapping time intervals. Then, the spatial-temporal series in city c is represented as

$$\mathcal{Y}_c = \{y_{r_c,k_c} | r_c \in \mathcal{R}_c, k_c \in \mathcal{K}_c\}, \tag{1}$$

where y_{r_c,k_c} is the spatial-temporal information to be predicted (e.g., traffic demand, air quality, climate value).

Problem Definition Suppose that we have a set of source cities $C_S = \{c_1, ..., c_S\}$ and a target city c_t with insufficient data (i.e., $\forall s \in \{1, \cdots, S\}$, $|\mathcal{K}_{c_s}| \gg |\mathcal{K}_{c_t}|$), our goal is to predict the spatial-temporal information of the next timestamp $k_{c_t} + 1$ in the testing dataset of the target city c_t, i.e.,

$$y^*_{r_{c_t},k_{c_t}+1} = \arg\max\nolimits_{y_{r_{c_t},k_{c_t}+1}} p(y_{r_{c_t},k_{c_t}+1} | \mathcal{Y}_{c_t}, f_{\theta_0}), \tag{2}$$

where f represents the ST-net which serves as the base model to predict the spatial-temporal series. Detailed discussion about ST-net is in given Section 4.1. More importantly, in the meta-learning paradigm, θ_0 denotes the initialization for all parameters of the ST-net, which is adapted from $\mathcal{Y}_{c_1}, \cdots, \mathcal{Y}_{c_S}$.

4 METHODOLOGY

In this section, we elaborate our proposed method **MetaST**. In particular, we first introduce the ST-net as the base model f for spatial-temporal prediction, and then present our proposed spatial-temporal meta-learning framework for knowledge transfer.

4.1 Spatial-Temporal Network

Recently, hybrid models combining convolution neural networks (CNN) [8] and LSTM [5] have achieved the state-of-the-art performances on spatial-temporal prediction. Thus, following [34], we adopt a CNN to capture the spatial dependency between regions, and an LSTM to model the temporal evolvement of each region.

In city c, for each region r_c at time k_c, we regard the spatial-temporal value of this region and its surrounding neighbors as an $N \times N$ image with v channels $\mathbf{Y}_{r_c,k_c} \in \mathbb{R}^{N \times N \times v}$, where region r_c is in the center of this image. For instance, when N=3, we are considering a center cell as well as 8 adjacent grid cells of the cell, which is a 3*3 size neighborhood. The number of channels v depends on a specific task. For example, in taxi volume prediction, we jointly predict taxi pick-up volume and drop-off volume, so that the number of channels equals two, i.e., $v = 2$. Taking $\mathbf{Y}_{r_c,k_c}$ as input $\mathbf{Y}^0_{r_c,k_c}$, a local CNN computes the q-th layer progressively:

$$\mathbf{Y}^q_{r_c,k_c} = \mathrm{ReLU}(\mathbf{W}^q_r * \mathbf{Y}^{q-1}_{r_c,k_c} + \mathbf{b}^q_r), \tag{3}$$

where $*$ represents the convolution operation, $\mathbf{W}^q_r$ and $\mathbf{b}^q_r$ are learnable parameters. After Q convolutional layers, a fully connected layer following a flatten layer is used to infer the spatial representation of region r_c as $\mathbf{s}_{r_c,k_c}$ eventually.

Then, for predicting y_{r_c,k_c+1}, we model the temporal unfolding of region r_c by passing all the spatial representations along the time span $\{k_c - |\mathcal{K}_c| + 1, ..., k_c\}$ through an LSTM, which can be formulated as

$$\begin{aligned}
\mathbf{i}_{r_c,k_c} &= \sigma(\mathbf{U}_i[\mathbf{s}_{r_c,k_c}; \mathbf{e}_{r_c,k_c}] + \mathbf{W}_i \mathbf{h}_{r_c,k_c-1} + \mathbf{b}_i),\\
\mathbf{f}_{r_c,k_c} &= \sigma(\mathbf{U}_f[\mathbf{s}_{r_c,k_c}; \mathbf{e}_{r_c,k_c}] + \mathbf{W}_f \mathbf{h}_{r_c,k_c-1} + \mathbf{b}_f),\\
\mathbf{d}_{r_c,k_c} &= \sigma(\mathbf{U}_d[\mathbf{s}_{r_c,k_c}; \mathbf{e}_{r_c,k_c}] + \mathbf{W}_d \mathbf{h}_{r_c,k_c-1} + \mathbf{b}_d),\\
\hat{\mathbf{c}}_{r_c,k_c} &= \tanh(\mathbf{U}_c[\mathbf{s}_{r_c,k_c}; \mathbf{e}_{r_c,k_c}] + \mathbf{W}_c \mathbf{h}_{r_c,k_c-1} + \mathbf{b}_c),\\
\mathbf{c}_{r_c,k_c} &= \mathbf{f}_{r_c,k_c} \circ \mathbf{c}_{r_c,k_c-1} + \mathbf{i}_{r_c,k_c} \circ \hat{\mathbf{c}}_{r_c,k_c},\\
\mathbf{h}_{r_c,k_c} &= \tanh(\mathbf{c}_{r_c,k_c}) \circ \mathbf{d}_{r_c,k_c},
\end{aligned} \tag{4}$$

where $\circ$ denotes Hadamard product, $\mathbf{U}_j$, $\mathbf{W}_j$, and $\mathbf{b}_j$ ($j \in \{i, f, d, c\}$) are learnable parameters, $\mathbf{i}_{r_c,k_c}$, $\mathbf{f}_{r_c,k_c}$, and $\mathbf{d}_{r_c,k_c}$ are forget gate vector, input gate vector, and output gate vector of the i-th context feature, respectively. $\mathbf{h}_{r_c,k_c}$ denotes the spatial-temporal representation of region r_c, and $|\mathcal{K}_c|$ is the number of time steps we use to consider the temporal information. Note that $\mathbf{e}_{r_c,k_c}$ represents other external features (e.g., weather, holiday) that can be incorporated, if applicable. By doing these, $\mathbf{h}_{r_c,k_c}$ encodes both the spatial and temporal information of region r_c. As a result, the value of the next time step of spatial-temporal series, i.e., $\hat{y}_{r_c,k_c+1}$, can be predicted by

$$\hat{y}_{r_c,k_c+1} = \tanh(\mathbf{W}_n \mathbf{h}_{r_c,k_c} + \mathbf{b}_n), \tag{5}$$

where $\mathbf{W}_n$ and $\mathbf{b}_n$ are learnable parameters. The output is scaled to (-1,1) via a tanh(·) function, to be consistent with the normalized spatial-temporal values. We later denormalize the prediction to get the actual demand values. We formulate the loss function of ST-net for each city c as:

$$\mathcal{L}_c = \sum_{r_c} \sum_{k_c} (\hat{y}_{r_c,k_c+1} - y_{r_c,k_c+1})^2, \tag{6}$$

so as to enforce the estimated spatial-temporal value to be as close to the ground-truth y_{r_c,k_c+1} as possible. As introduced previously, we denote all the parameters of the ST-net as θ, and the ST-net parameterized by θ as f_θ. For better illustration, we visualize the structure of the spatial-temporal network (**ST-net**) in Figure 1.

4.2 Knowledge Transfer

Next, we propose a meta-learning framework that enables our ST-net model to borrow knowledge from multiple cities. The framework consists of two parts: adapting the initialization and learning the spatial-temporal memory. We present the whole framework in Figure 2.

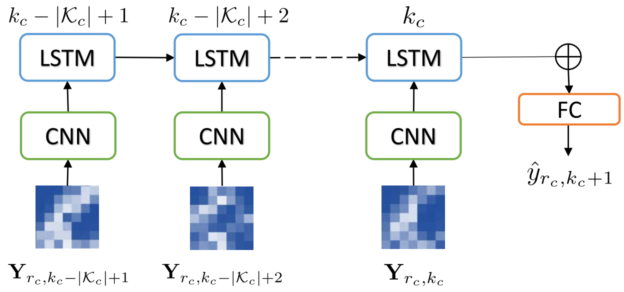

$k_c - |\mathcal{K}_c| + 1$
$k_c - |\mathcal{K}_c| + 2$
k_c
LSTM
LSTM
LSTM
CNN
CNN
CNN
FC
$\hat{y}_{r_c,k_c+1}$
$\mathbf{Y}_{r_c,k_c-|\mathcal{K}_c|+1}$
$\mathbf{Y}_{r_c,k_c-|\mathcal{K}_c|+2}$
$\mathbf{Y}_{r_c,k_c}$

Figure 1: The framework of the ST-net. The heatmaps are exemplar spatial-temporal values. CNN is used to capture spatial dependency and then LSTM is used to handle temporal correlation.

4.2.1 Adapting the Initialization. As we described before, we are supposed to increase the stability of prediction by transferring knowledge from multiple source cities. Since the spatial-temporal correlation of limited data in a target city may vary considerably from city to city and even from time to time. For example, in traffic prediction, the knowledge to be transferred to Boston with limited weekend data is expected to differ from that to Chicago with limited weekday data. Thus, the extracted knowledge from multiple cities is expected to include comprehensive spatial-temporal correlations such as spatial closeness and temporal dependency, so that we can adapt the knowledge to a target city with limited data under different scenarios.

In ST-net, the parameters θ are exactly the knowledge which encrypts spatial-temporal correlations. To effectively adapt the parameters to a target city, as suggested in model-agnostic meta-learning (MAML) [2], initialization of θ from multiple source cities, i.e., θ_0, so that the ST-net initialized by θ_0 achieves the minimum of the average of generalization losses over all source cities, i.e.,

$$\theta_0 = \min_{\theta_0} \sum_{c_s \in C_s} \mathcal{L}'_{c_s}(f_{\theta_0 - \alpha \nabla_\theta \mathcal{L}_{c_s}(f_\theta)}). \tag{7}$$

Here we would note that $\mathcal{L}_{c_s}(f_{\theta_{c_s}})$ denotes the training loss on the training set of a city c_s sampled from C_s, i.e., $\mathcal{D}_{c_s}$ (refer to S-train in Figure 2). We illustrate the iterative update of the parameters θ_{c_s} during the training process (shown as the yellow solid arrow in Figure 2), by showing one exemplar gradient descent, i.e.,

$$\theta_{c_s} = \theta_0 - \alpha \nabla_\theta \mathcal{L}_{c_s}(f_\theta). \tag{8}$$

In practice, we can use several steps of gradient descent to update from the initialization θ_0 to θ_{c_s}. For each city c_s, the training process is repeated on batches of series sampled from D_{c_s}.

Since Eq. (7) minimizes the generalization loss, $\mathcal{L}'_{c_s}(\cdot)$ evaluates the loss on the test set of the city c_s, i.e., $\mathcal{D}'_{c_s}$ (refer to S-test in Figure 2). By optimizing the problem in Eq. (7) using stochastic gradient descent (shown as the purple solid arrow in Figure 2), we obtain an initialization which can generalize well on different source cities. Therefore, it is widely expected that transferring the initialization θ_0 to a target city c_t would also bring superior generalization performance, which we will detail in the end of this section.

4.2.2 Spatial-Temporal Memory. In spatial-temporal prediction, long-term spatial-temporal patterns (e.g., periodic patterns) play an important role [33, 45]. These patterns reflect the spatial functionality of each region and can be regarded as the global property shared by different cities. An example of spatial-temporal patterns and their corresponding regions are shown in Figure 3. In this figure, one region around NYU-Tardon in New York City and another one around Georgetown University in Washington DC are selected. The averaged 5-days' taxi demand distributions of both regions are daily periodic and similar, whose value are higher in the afternoon when students and faculties go back to home. The similarity of distributions between different cities verifies our assumption, i.e., spatial functionality is globally shared. However, in a target city, these patterns are hard to be captured with limited data due to missing values or the effects of special events (e.g., holidays). Thus, we propose a global memory, named **ST-mem**, to store long-term spatial-temporal patterns and further transfer to improve prediction accuracy in target cities. The framework of ST-Mem is illustrated in Figure 4.

Based on spatial-temporal patterns, we first cluster all the regions in source cities to G categories. The categories G of regions represent different region functionalities. For region r_c, the clustering results are denoted as $\mathbf{o}_{r_c} \in \mathbb{R}^G$. If region r_c belongs to cluster g, $\mathbf{o}_{r_c}(g) = 1$, otherwise $\mathbf{o}_{r_c}(g) = 0$. Then, we construct a parameterized spatial-temporal memory $\mathcal{M} \in \mathbb{R}^{G \times d}$. Each row of the memory stores the pattern representation of a category, and the dimension of the pattern representation is d.

Next, we utilize the knowledge of patterns stored in memory $\mathcal{M}$, and distill this knowledge for prediction via attention mechanism [14]. Since we only have short-term data in a target city, we use the representation of short-term data to query ST-mem. In particular, we linearly project the spatial-temporal representation $\mathbf{h}_{r_c,k_c}$ of ST-net to get the query vector for the attention mechanism, which is formulated as:

$$\mathbf{v}_{r_c,k_c} = \mathbf{W}_l \mathbf{h}_{r_c,k_c} + \mathbf{b}_l. \tag{9}$$

Then, we use the query vector $\mathbf{v}_{r_c,k_c} \in \mathbb{R}^d$ to calculate the similarity score between it and the pattern representation for each category g. Formally, the similarity score is defined as

$$\mathbf{p}_{r_c,k_c}(g) = \frac{\exp(\langle \mathbf{v}_{r_c,k_c}, \mathcal{M}(g) \rangle)}{\sum_{g'=1}^{G} \exp(\langle \mathbf{v}_{r_c,k_c}, \mathcal{M}(g') \rangle)}, \tag{10}$$

where $\mathcal{M}(g)$ means the g-th row of memory (i.e., for the g-th pattern category). We calculate the representation of spatial-temporal pattern $\mathbf{z}_{r_c,k_c}$ as:

$$\mathbf{z}_{r_c,k_c} = \sum_{g=1}^{G} \mathbf{p}_{r_c,k_c}(g) * \mathcal{M}(g). \tag{11}$$

Then, we concatenate the representation $\mathbf{z}_{r_c,k_c}$ of spatial-temporal patterns with the representation $\mathbf{h}_{r_c,k_c}$ of ST-net. And the input of the final layer $\mathbf{h}_{r_c,k_c}$ in Eq. (5) is replaced by the enhanced representation i.e.,

$$\hat{y}_{r_c,k_c+1} = \tanh(\mathbf{W}'_n[\mathbf{h}_{r_c,k_c}; \mathbf{z}_{r_c,k_c}] + \mathbf{b}'_n). \tag{12}$$

where $\mathbf{W}'_n$ and $\mathbf{b}'_n$ are learnable parameters.

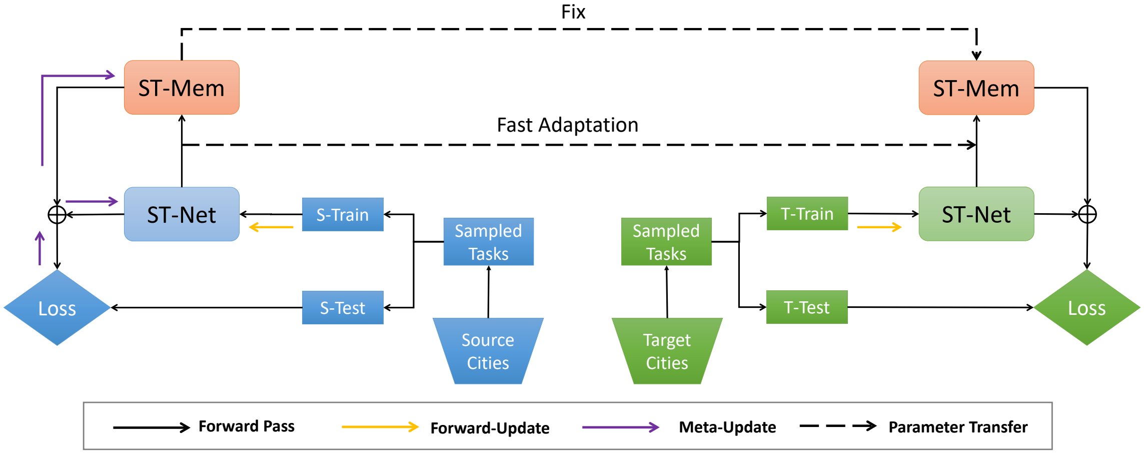


Figure 2: The framework of proposed MetaST. ST-net and ST-mem mean spatial-temporal network and spatial-temporal memory. S-train and S-test, T-train and T-test represent the training and testing set of source tasks (i.e., source cities) and target tasks (i.e., source cities), respectively. In the process of knowledge transfer, the parameters of ST-net will be updated by the training set in target city (i.e., T-train), while the parameters of ST-mem are fixed.

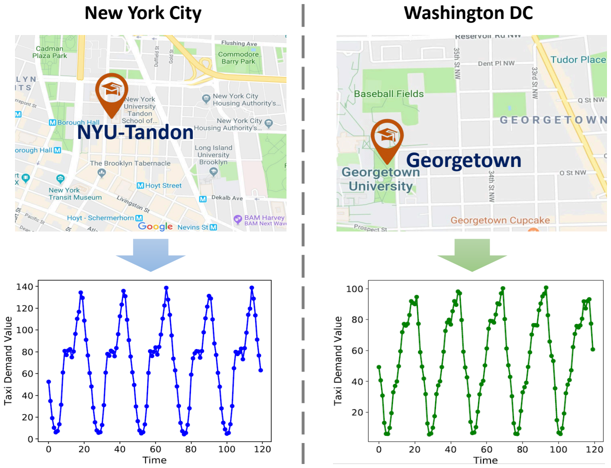


Figure 3: The illustration of spatial-temporal patterns of two regions and their corresponding region functionality.

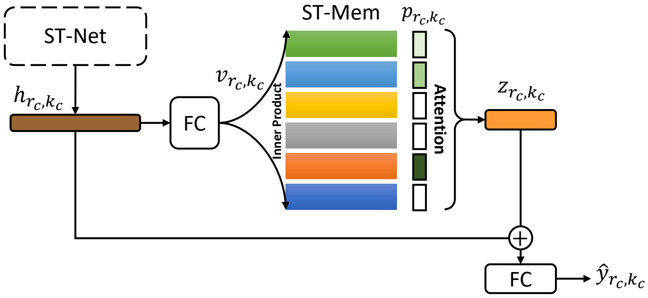


Figure 4: The stucture of ST-mem. The spatial-temporal representation $\mathbf{h}_{r_c,k_c}$ is projected as a query vector $\mathbf{v}_{r_c,k_c}$. Then, the attention mechanism is used to get the pattern representation $\mathbf{z}_{r_c,k_c}$. The short-term spatial-temporal representation and pattern representation are concatenated together for prediction.

In order to learn the memory $\mathcal{M}$, we construct the clustering loss of city c_s, which enforce the attention scores to be consistent with previous clustering results $\mathbf{o}_{r_c}$. The formulation of the clustering loss is as follows:

$$\mathcal{L}_{clu} = -\sum_{r_c}\sum_{k_c} \mathbf{o}_{r_c} \log(\mathbf{p}_{r_c,k_c}). \tag{13}$$

Additionally, the memory $\mathcal{M}$ is also updated when we train the MetaST framework, together with the initialization θ_0. Thus, we revise the loss function in Eq. (7) by adding the clustering loss. Then, our final objective function is:

$$\theta_0, \mathcal{M} = \min_{\theta_0, \mathcal{M}} \sum_{c_s \in C_s} \gamma \mathcal{L}_{clu}(f_{\theta_{c_s}}) + \mathcal{L}'_{c_s}(f_{\theta_{c_s}}), \tag{14}$$

where γ is a trade-off hypeparameter and is used to balance the effect of each part. $\theta_{c_s} = \theta_0 - \alpha\nabla_\theta \mathcal{L}_{c_s}(f_\theta)$ is defined in Eq. (8). By using testing set $\mathcal{D}'_{c_s}$ of each city c_s, the memory $\mathcal{M}$ and initialization θ_0 are updated by gradient descent. Note that, as we discussed before, the spatial-temporal patterns are common property between cities. We do not update memory $\mathcal{M}$ when training a specific task (i.e., Eq. (8)). In Figure 2, the purple solid arrow indicates the process of meta-update. In Eq. (14), the initialization θ_0 and memory $\mathcal{M}$ are mutually constrained. Since the memory $\mathcal{M}$ provides region-level relationship via spatial-temporal patterns, it can also help learn the initialization θ_0 of ST-net.

4.2.3 Transfer Knowledge to Target Domain. To improve the prediction in target cities, we transfer the ST-mem $\mathcal{M}$ and initialization θ_0 of ST-net. The black dotted line in Figure 2 shows the process of knowledge transfer. For each new city c_t sampled from C_t, the

```
Algorithm 1: Framework of MetaST
  Input: Set of target cities $C_t$; Set of source cities $C_s$;
         hyperparameter $\gamma$
  Output: Spatial-temporal predictions of each target city
  Randomly initialize $\theta_0$ and $\mathcal{M}$;
  Cluster all regions in $C_s$ and get $\mathbf{o}_{r_c}$ for each region $r_c$;
  /* learning $\theta_0$ and $\mathcal{M}$ on source cities */
  while not done do
      Sample a batch of cities from $C_s$;
      for each city $c_s$ do
          Sample a set $\mathcal{D}_{c_s}$ from $c_s$;
          Evaluate $\nabla\mathcal{L}_{c_s}(f_{\theta_{c_s}})$ using $\mathcal{D}_{c_s}$ by Eq. (6);
          Compute adapted parameters $\theta_{c_s}$ with gradient
            descent by Eq. (8);
          Sample a new set $\mathcal{D}^{'}_{c_s}$ from $c_s$;
          Evaluate $\mathcal{L}^{'}_{c_s}(f_{\theta_s})$, $\mathcal{L}_{clu}(f_{\theta_{c_s}})$ by $\mathcal{D}^{'}_{c_s}$;
      end
      Update $\theta_0$, $\mathcal{M}$ by gradient descent;
  end
  /* Evaluate our model on target cities */
  for each city $c_t$ in $C_t$ do
      Sample a sample set $\mathcal{D}_{c_t}$ from $c_t$;
      Compute adapted parameters $\theta_{c_t}$ with gradient descent by
        Eq. (15);
      Sample new series $\mathcal{D}^{'}_{c_t}$ and predict;
  end
```

memory is fixed and the parameters θ_{c_t} are trained with initialization θ_0 and training samples $\mathcal{D}_{c_t}$ (i.e., T-train in Figure 2), which is defined as:

$$\theta_{c_t} = \theta_0 - \alpha\nabla_{\theta}\mathcal{L}_{c_t}(f_{\theta}), \tag{15}$$

where $\mathcal{L}_{c_t}$ is the loss function of training set in target city t and the formulation is:

$$\mathcal{L}_{c_t} = \sum_{r_t}\sum_{k_t}(\hat{y}_{r_t,k_t+1} - y_{r_t,k_t+1})^2. \tag{16}$$

The predicted value $\hat{y}_{r_t,k_t+1}$ is calculated via Eq. (12). Hence, we distill knowledge of spatial-temporal patterns from source cities to target city via ST-mem $\mathcal{M}$. Finally, we evaluate the model $f_{\theta_{c_t}}$ on testing set $\mathcal{D}^{'}_{c_t}$ (i.e., T-test in Figure 2) of city c_t and get the prediction value of this city. The whole framework of MetaST is shown in Algorithm 1.

5 EXPERIMENT

In this section, we conduct extensive experiments for two domain applications to answer the following research questions:

- Whether MetaST can outperform baseline methods for inference tasks, i.e., traffic volume prediction in and water quality (pH value) prediction?
- How do various hyper-parameters, e.g., the dimensions of each cluster in ST-mem or trade-off factor, impact the model's performance?
- Whether ST-mem can detect distinguished spatial-temporal patterns?

Table 1: Data Statistics of Traffic Prediction

Data	City	Time Span (m/d/y)	Trips	Size
Taxi	NYC	1/1/15-7/1/15	6,748,857	10×20
	DC	5/1/15-1/1/16	8,151,077	16×16
	Porto	7/1/13 - 6/30/14	1,710,671	20×10
	CHI	9/1/13-11/1/14	124,820	15×18
	BOS	10/1/12 - 10/31/12	839,897	18×15
Bike	NYC	1/1/17-12/31/17	16,364,475	10×20
	DC	1/1/17-12/31/17	3,732,263	16×16
	CHI	1/1/17-2/1/17	106,165	15×18

5.1 Application-*I*: Traffic Prediction

5.1.1 Problem Overview. We first conduct experiments on two traffic prediction tasks, i.e., taxi volume prediction and bike volume prediction. Similar as the previous traffic prediction task in [33, 41], each individual trip departs from a region, and then arrives at the destination region. Thus, our task is to predict the pick-up (start) and drop-off (end) volume of taxi (and bike) at each time interval for each region. The time inveral of traffic prediction task is one hour. We use *Root Mean Square Error* (RMSE) to evaluate our model, which is defined as:

$$\text{RMSE} = \frac{1}{|N|}\sqrt{\sum_{r_t}\sum_{k_t}(\hat{y}_{r_t,k_t+1} - y_{r_t,k_t+1})^2_2}, \tag{17}$$

where $\hat{y}_{r_t,k_t+1}$ and y_{r_t,k_t+1} represent predicted value and actual value, respectively. $|N|$ means the number of all samples in testing set.

5.1.2 Data Description. For taxi volume prediction, we collect five representative mobility datasets from five different cities to evaluate the performance of our proposed model, i.e., New York City (NYC)[1], Washington DC (DC), Chicago (CHI), Porto[2], and Boston (BOS). We use NYC, DC, Porto as the source cities and CHI, BOS as the target cities. Note that the Boston data does not have drop-off records, and thus we only report the results on predicting pick-up volume.

For bike volume prediction, we collect three datasets from four cities, i.e., NYC[3], DC[4], and CHI[5]. NYC and DC are used as source cities, and CHI are used as target city. All trips in the above datasets contain time and geographic coordinates of pick-up and drop-off. For each city above, as discussed before, we spatially divide them to a grid map. The grid map size of NYC, DC, CHI, BOS, Porto is 10 × 20, 16 × 16, 15 × 18, 18 × 15, 20 × 10, respectively. Detailed statistics of these datasets are listed in Table 1.

In addition, for each source city, we select 80% data for training and validation, and the rest for testing. For each target city, we select the 1-day, 3-day and 7-day data for training, and the rest for testing.

[1] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

[2] https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data

[3] https://www.citibikenyc.com/system-data

[4] https://www.capitalbikeshare.com/system-data

[5] https://www.divvybikes.com/system-data

5.1.3 *Compared Baselines.* We compare our model with the following two categories of methods: non-transfer methods and transfer methods. Note that, for non-transfer baselines, we only use the training data of target cities (limited training data) to train the model and use the testing data to evaluate it. For transfer baselines, we transfer the learned knowledge from source cities to improve the prediction accuracy.

Non-transfer baselines:

- **Historical Average (HA)**: For each region, HA predicts spatial-temporal value based on the average value of the previous relative time. For example, if we want to predict the pick-up volume at 9:00am-10:00am, HA is the average value of all time intervals from 9:00am to 10:00am in training data.
- **ARIMA**: Auto-Regressive Integrated Moving Average (ARIMA) is a traditional time-series prediction model, which considers the temporal relationship of data.
- **ST-net**: This method simply use the spatial-temporal neural network defined in Section 4.1 to predict traffic volume. Both pick-up and drop-off volume are predicted together.

Transfer baselines:

- **Fine-tuning Methods**: We include two types of fine-tuning methods: (1) Single-source fine-tuning (**Single-FT**): train ST-net in one source city (e.g., NYC, DC or Porto in taxi data) and fine-tune the model for target cities; and (2) multi-source fine-tune (**Multi-FT**): mix all samples from all source cities and fine-tune the model in target cities.
- **RegionTrans [27]**: RegionTrans transfers knowledge from one city to another city for traffic flow prediction. Since we do not have auxiliary data, we compare the S-Match of RegionTrans. For each region in target city, RegionTrans uses short period data to calculate the linear similarity value with each region in source city. Then, the similarity is used as regularization for fine-tuning. For fair comparison, the base model of RegionTrans (i.e., the ST-net) is same as MetaST.
- **MAML [2]**: an state-of-the-art meta-learning method, which learns a better initialization from multiple tasks to supervise the target task. MAML uses the same base model (i.e., the ST-net) as MetaST.

5.1.4 *Experimental Settings.*

Hyperparameter Setting. For ST-net, we set the number of filters in CNN as 64, the size of map in CNN as 7×7, the number of steps in LSTM as 8, and the dimension of hidden state of LSTM as 128. In the training process of taxi volume prediction, we set the learning rate of inner loop and outer loop as 10^{-5} and 10^{-5} respectively. For bike volume prediction, we set the learning rate of inner loop and outer loop as 10^{-5} and 10^{-6} respectively. The parameter γ is set as 10^{-4}. The number of updates for each task is set as 5. All the models are trained by Adam. The training batch size for each meta-iteration is set as 128, and the maximum of iteration of meta-learning is set as 20000.

Spatial-temporal Clustering. In addition, since the pattern of traffic volume usually repeats every day. Thus, in this work, we use averaged 24-hour patterns of each region to represent its spatial-temporal pattern. We use K-means to cluster these patterns to 4 groups. Furthermore, in this work, we do not use other external features, which means that $[\hat{\mathbf{y}}_{r_c,k_c};\mathbf{e}_{r_c,k_c}]$ in Eq. (4) equals to $\hat{\mathbf{y}}_{r_c,k_c}$. We set the size of pattern representation in memory as 8.

5.1.5 *Results.* We implement our model and compare it with the baselines on taxi and bike-sharing datasets. We run 20 testing times and report the average values. The results are shown in Table 2 and Table 3, respectively. According to these tables, we draw the following conclusions.

- Comparing with ST-net and some single-FT models, in some cases (e.g., 1-day training data), traditional time-series prediction methods (i.e., HA and ARIMA) achieves competitive performance in this problem. The reason is that traffic data show a strong daily periodicity, so that if we only have limited traffic data, we can still use periodicity to predict traffic volume.
- Comparing with ST-net, all transfer learning models (i.e., fine-tune models (including Single-FT and Multi-FT models), MAML, RegionTrans, MetaST) significantly improves the performance (i.e., lower the RMSE values). The results indicate that (1) it is difficult to train a model from random initialization with limited data; (2) the knowledge transfer between cities is effective for prediction.
- In most cases, Multi-FT outperforms Single-FT. One possible reason is that Multi-FT increases the diversity of source domain. In other cases (e.g., Chicago pick-up prediction with 3-day training data), the best performance of Single-FT outperforms Multi-FT. The results implicitly indicates that if there exists a source city that is optimally transferable to the target city, simply mixing other cities may hurt the performance.
- RegionTrans models only slightly outperform their corresponding fine-tuning models (i.e., RegionTrans from NYC to Chicago v.s., Single-FT from NYC to Chicago). The results suggest that regional linear similarity regularization may not capture complex relationship between regions. In addition, since the data from different cities are collected from different time, regional similarity calculations may be inaccurate. Thus, it is not an effective and flexible way to transfer knowledge via regional similarity regularization.
- MAML and MetaST achieve better performance than fine-tuning methods and RegionTrans. This is because fine-tuning methods and RegionTrans cannot be adapted to different scenarios, and thereby decreasing the stability of transfer. However, MAML and MetaST not only learn the initialization based on multiple cities, but also achieve the best performance in every scenario sampled from these cities.
- Our proposed MetaST achieves the best performance in all experimental settings. Comparing with MAML, the averaged relative improvement is 5.0%. This is because our model can also capture and transfer long-term information, besides learning a better initialization. Moreover, the long-term pattern memory helps learn a further enhanced initialization. The stability of knowledge transfer increases to the highest degree.
- Finally, we compare the results under different training data size in target city (i.e., 1-day, 3-day, and 7-day data). For every learning models (except HA and ARIMA), the performance improves with more training data. Our proposed MetaST still outperforms all baselines.

Table 2: Comparing with baselines for taxi volum prediction

Taxi Data		Chicago						Boston		
		Pick-up			Drop-off			Pick-up		
		1-day	3-day	7-day	1-day	3-day	7-day	1-day	3-day	7-day
HA		2.83	2.36	2.18	2.67	2.28	2.13	11.07	9.13	7.71
ARIMA		3.19	2.76	2.71	2.93	2.43	2.41	11.02	10.25	9.36
ST-net		10.51	6.04	3.89	11.22	6.42	3.99	30.01	17.02	13.28
Single-FT	NYC	2.72	2.06	1.76	2.84	2.75	1.89	11.71	10.62	7.76
	DC	3.90	2.62	2.05	4.17	2.19	2.15	14.39	12.32	9.19
	Porto	2.57	1.87	1.60	2.87	2.03	1.74	14.12	11.56	7.63
Multi-FT		2.18	1.89	1.60	2.20	2.08	1.69	9.90	9.41	6.68
RegionTrans	NYC	2.53	2.01	1.69	2.83	2.56	1.72	11.12	10.49	7.49
	DC	3.87	2.51	2.04	3.95	2.16	2.03	13.98	11.83	8.87
	Porto	2.45	1.83	1.60	2.85	1.98	1.73	13.05	10.97	7.43
MAML		2.01	1.78	1.52	2.10	1.92	1.66	8.13	7.59	5.88
MetaST		**1.95****	**1.70****	**1.48****	**2.04****	**1.79****	**1.65***	**7.48****	**7.15****	**5.67****

** (*) means the result is significant according to Student's T-test at level 0.01 (0.05) compared to MAML

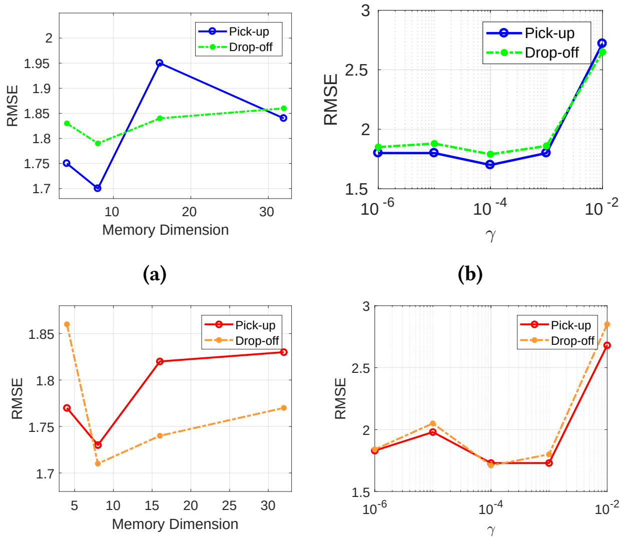
Table 3: Comparing with baselines for Bike Volume Prediction.

Bike Data		Chicago					
		Pick-up			Drop-off		
		1-day	3-day	7-day	1-day	3-day	7-day
HA		4.97	3.69	3.64	4.96	3.67	3.63
ARIMA		4.86	4.89	4.79	4.83	4.97	4.86
ST-net		7.61	5.57	3.83	8.03	5.45	3.51
SFT[1]	NYC	2.52	2.49	1.95	2.49	2.41	1.87
	DC	1.88	1.99	1.69	2.03	2.20	1.67
Multi-FT		1.97	2.05	1.62	1.90	1.98	1.59
RT[1]	NYC	2.50	2.23	1.87	2.36	2.18	1.76
	DC	1.86	1.95	1.66	1.98	2.09	1.63
MAML		1.85	1.87	1.62	1.85	1.79	1.56
MetaST		**1.76****	**1.73****	**1.45****	**1.83****	**1.71****	**1.46****

[1] Due to the space limitation, in this table, SFT, RT mean single-FT and RegionTrans respectively.

** (*) means the result is significant according to Student's T-test at level 0.01 (0.05) compared to MAML.

5.1.6 Parameter Sensitivity. MetaST involves a number of hyper-parameters. In this subsection, we evaluate how different selections of hyper-parameters impact our model's performance. Specifically, we analyze the impacts of two key parameters of spatial-temporal memory, i.e., the dimension d of pattern representation and the trade-off factor γ of two losses in the joint objective. All other hyperparameters are set as introduced in Section 5.1.4. We use the scenario of 3-day data for sensitivity analysis.



(a)

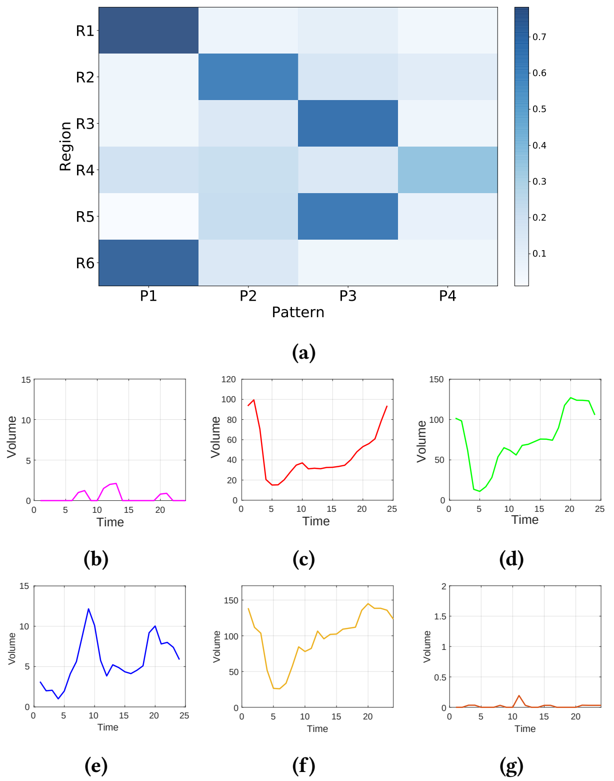
(b)

(c)

(d)

Figure 5: (a) (c) RMSE with respect to the dimension of spatial-temporal memory in Chicago taxi/bike volume prediction, respectively; (b) (d) RMSE with respect to the value of γ in Chicago taxi/bike volume prediction, respectively.

For the the dimension d of pattern representation, we change the dimension of pattern representation from 4 to 32 in spatial-temporal memory. The performance of Chicago taxi and bike volume prediction are shown in Figure 5a and Figure 5c, respectively. We find that the performance increases in the beginning but decreases later. One potential reason is that the spatial-temporal memory provides too little information when the dimension is too small, while it can include too much irrelevant information when the dimension is too large. Both of the scenarios hurt the performance. Another experiment of trade-off factor γ also demonstrates similar assumption. We change the parameter γ in Eq. (14) from 10^{-6} to 10^{-2}. Higher value of γ means higher importance of spatial-temporal memory. The results of Chicago taxi and bike volume prediction are shown in Figure 5b and Figure 5d, respectively. We can see the similar change of the performance, increasing at first but decreasing later.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 6: Spatial-temporal patterns detected by the memory $\mathcal{M}$ in Boston Taxi data. (a): probability of 4 pattern values of 6 regions; (b),(c),(d),(e),(f),(g) denote actual patterns of R1, R2, R3, R4, R5, R6, respectively.

5.1.7 Case Study: Visualization of Detected Patterns. To intuitively demonstrate the efficacy of the usage of the Spatial-temporal memory, we visualize patterns detected from Boston taxi pick-up volume prediction. We also use 3-day data for this case study. We randomly select six regions and show the similarity values with respect to each pattern category in the memory $\mathcal{M}$. The results are shown in Figure 6a. The corresponding actual patterns of each region are shown in Figure 6b, Figure 6c, Figure 6d, Figure 6e, Figure 6f, and Figure 6g respectively. We can see that the taxi pick-up volume of R1 and R6 is almost zero and their attention weights are also similar (Pattern 1 is activated). The volume distribution in R2, R3 and R5 have one peak. The peak time of R2 is around 1:00am - 2:00am (Pattern 2 is activated). The pattern and attention weights of R3 and R5 are similar and the peak time is around 8:00pm - 9:00pm (Pattern 3 is activated). In R4, the volume distribution has two peaks (Pattern 4 is activated). One peak is around 9:00am - 10:00am, another one is 8:00pm - 9:00pm. The results indicate that the memory can distinguish regions with different patterns.

5.2 Application-II: Water Quality Prediction

5.2.1 Problem Overview. The second application studied in this work is water quality prediction task. We also conduct a water quality prediction experiment. In this scenario, the water quality is represented by pH value of water, because pH value is easier to measure than other chemical metrics of water quality. We aim at predicting pH value in a specific location of next month (i.e., the time interval of water quality prediction is one month), as the changing of pH indicates the relative changes in the chemical composition in water. RMSE is still used as the evaluation metric in this task.

5.2.2 Dataset. The data used in this experiment is collected from the Water Quality Portal[6] of the National Water Quality Monitoring Council. It spans about 52 years from 1966 to 2017. Each record represents one surface water sample with longitude, latitude, date and pH value in a standard unit. The continental U.S. area is roughly split to six areas: Pacific, West, Midwest, Northeast, Southwest, South. Note that, in the water quality prediction task, the areas are treated as the cities in previous descriptions.

In addition, due to the sparsity of sampling points, we split each area into a grid region map, the size of each grid being 0.5°of latitude by 0.5°of longitude. Thus, the map sizes of all the six areas are 25×50, 30×25, 35×25, 30×25, 50×25, 45×25, respectively. The pH value of each region is represented by the median value of all sampling points in this region. We select Pacific, West, Midwest as source areas, and Northeast, Southwest, South as target areas.

5.2.3 Baselines. We use the same baselines as in the experiments for traffic prediction. Both non-transfer methods and transfer methods are used to evaluated our algorithm. Note that, when calculating HA, the relative time is monthly. For example, if we want to predict the water quality at May, HA is the average value of all pH values at May in training data.

5.2.4 Experimental Settings.

Hyperparameter Setting. Similar as the traffic prediction application, we do have external features in the water quality prediction task. For the learning process of spatial-temporal framework, we set the maximum of iterations as 5000, the number of filters in CNN as 32, the dimension of hidden states of LSTM as 64 and the size of memory representation as 4. Other hyperparameters are the same as traffic prediction.

Spatial-temporal Clustering. By analyzing the data, pH in the current month is strongly correlated with pH in the same month of previous year. Thus, we use the 12-month periodic pattern of each region. Similar as traffic prediction task, K-means is also used to cluster these patterns to 3 groups. DTW distance is used to measure the distance of K-means.

[6] https://www.waterqualitydata.us/

Table 4: Comparing with baselines for water quality prediction

pH Data		Northeast			Southwest			South		
		1-year	3-year	7-year	1-year	3-year	7-year	1-year	3-year	7-year
HA		2.302	2.112	2.016	3.051	2.811	2.770	2.402	2.141	2.028
ARIMA		2.309	2.328	2.212	3.153	3.178	3.103	2.372	2.363	2.209
ST-net		4.536	3.806	1.850	2.694	2.094	1.008	4.237	3.951	1.662
Single-FT	West	1.236	1.128	0.862	0.935	0.716	0.683	1.138	1.043	0.837
	Midwest	1.048	1.004	0.806	0.791	0.653	0.622	0.970	0.951	0.793
	Pacific	1.249	1.140	0.854	0.928	0.711	0.671	1.172	1.031	0.835
Multi-FT		1.010	0.987	0.798	0.706	0.587	0.567	0.909	0.898	0.730
RegionTrans	West	1.233	1.115	0.853	0.924	0.698	0.682	1.121	0.993	0.826
	Midwest	1.047	0.988	0.796	0.783	0.651	0.619	0.965	0.938	0.769
	Pacific	1.243	1.098	0.851	0.916	0.693	0.652	1.128	1.012	0.813
MAML		0.997	0.955	0.784	0.701	0.579	0.559	0.907	0.897	0.710
MetaST		**0.903****	**0.898****	**0.758****	**0.649****	**0.541****	**0.514****	**0.820****	**0.803****	**0.650****

** (*) means the result is significant according to Student's T-test at level 0.01 (0.05) compared to MAML

5.2.5 Results. We implement our model and compare with baselines on the water quality prediction task. We run 20 testing tasks and report the average values in Table 4. Most experiment results and their explanation are similar to traffic prediction. Besides, from this table, we observe that:

- Comparing with Multi-FT model, the performance of MAML only slightly improves in most cases. The potential reason is that the regions in the source areas may be homogeneous in short-term performance. Thus, compared to MAML which learning a initialization, simply mixing all samples (i.e., Multi-FT) may not significantly hurt the performance significantly.
- MetaST achieves the best performance compared with all baselines with the averaged relative improvement as 7.7%. Since MetaST provides more detailed long-term information by spatial-temporal memory and distills the long-term information to target city, which explicitly increases the diversity of regions in source domain.

5.2.6 Parameter Sensitivity. Following the same step of traffic prediction task, we investigate the effect of the dimension d of pattern representation and the trade-off factor γ of two losses in the joint objective on MetaST performance. The performance of d from 2 to 32 and γ from 10^{-7} to 10^{-2} on water quality value prediction is shown in Figure 7a and Figure 7b, respectively. Both Figure 7a and Figure 7b are evaluated on Northeast water quality prediction with 3-year training data. Accordingly, MetaST achieves the best performance when $d = 4$ and $\gamma = 10^{-4}$. Similar as the reasons in traffic prediction, the results indicate that suitable selection of d and γ lead to the best performance.

6 CONCLUSION AND DISCUSSION

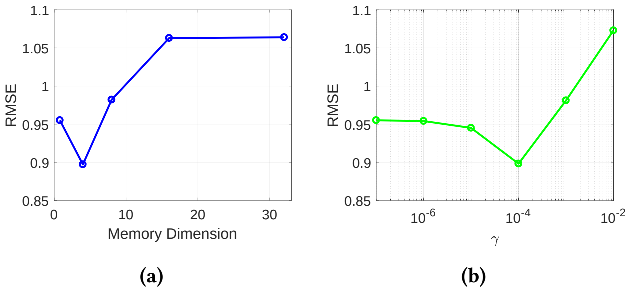


Figure 7: (a) RMSE with respect to the dimension of spatial-temporal memory; (b) RMSE with respect to the value of γ.

In this paper, we study the problem of transferring knowledge from multiple cities for spatial-temporal prediction. We propose a novel MetaST model which leverages learned knowledge from multiple cities to help with the prediction in target data-scarce cities. Specifically, the proposed model learns a well-generalized initialization of spatial-temporal prediction model for easier adaptation. Then, MetaST a global pattern-based spatial-temporal memory from all source cities. We test our model on two spatial-temporal prediction problem from two different domains: traffic prediction and environment prediction. The results show the effectiveness and of our proposed model.

For future work, we plan to investigate from two directions: (1) We plan to further consider network structure (e.g., road structure) and combine it with our proposed model. For example, a simple extension is that we can use graph convolutional network as our base model; (2) We plan to explain the black-box transfer learning framework, and analyze which information is transferred (e.g., spatial correlation, region functionality).

ACKNOWLEDGMENTS

We thank Xiuwen Yi for feedback on an early draft of this paper. The work was supported in part by NSF awards #1652525, #1618448, and #1639150. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Wenyuan Dai, Ou Jin, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2009. Eigentransfer: a unified framework for transfer learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 193–200.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*. 1126–1135.

[3] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*. IEEE, 2066–2073.

[4] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2011. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 999–1006.

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[6] Tsuyoshi Idé and Masashi Sugiyama. 2011. Trajectory regression on road networks. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 203–208.

[7] Jintao Ke, Hongyu Zheng, Hai Yang, and Xiqun Michael Chen. 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies* 85 (2017), 591–608.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[9] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *Sixth International Conference on Learning Representations*.

[10] Marco Lippi, Matteo Bertini, and Paolo Frasconi. 2013. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE TIST* 14, 2 (2013), 871–882.

[11] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, and Yi Yang. 2018. Transductive Propagation Network for Few-shot Learning. *arXiv preprint arXiv:1805.10002* (2018).

[12] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment* 10, 10 (2017), 1010–1021.

[13] Yiding Liu, Kaiqi Zhao, and Gao Cong. 2018. Efficient Similar Region Search with Deep Metric Learning. In *SIGKDD*. ACM, 1850–1859.

[14] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[15] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. 2013. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1393–1402.

[16] Devang K Naik and RJ Mammone. 1992. Meta-neural networks that learn by learning. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, Vol. 1. IEEE, 437–442.

[17] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[18] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a Model for Few-Shot Learning. *ICLR* (2016).

[19] Shashank Shekhar and Billy Williams. 2008. Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record: Journal of the Transportation Research Board* 2024 (2008), 116–125.

[20] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. 4077–4087.

[21] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, and Jieping Ye. 2017. The Simpler The Better: A Unified Approach to Predicting Original Taxi Demands on Large-Scale Online Platforms. In *KDD*. ACM.

[22] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *CVPR*.

[23] Bao Wang, Duo Zhang, Duanhao Zhang, P Jeffery Brantingham, and Andrea L Bertozzi. 2017. Deep learning for real time crime forecasting. *arXiv preprint arXiv:1707.03340* (2017).

[24] Dong Wang, Wei Cao, Jian Li, and Jieping Ye. 2017. DeepSD: Supply-Demand Prediction for Online Car-Hailing Services Using Deep Neural Networks. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 243–254.

[25] Hongjian Wang and Zhenhui Li. 2017. Region representation learning via mobility flow. In *CIKM*. ACM, 237–246.

[26] Hongjian Wang, Huaxiu Yao, Daniel Kifer, Corina Graif, and Zhenhui Li. 2017. Non-Stationary Model for Crime Rate Inference Using Modern Urban Data. *IEEE Transactions on Big Data* (2017).

[27] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2018. Crowd Flow Prediction by Deep Spatio-Temporal Transfer Learning. *arXiv preprint arXiv:1802.00386* (2018).

[28] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *SIGKDD*. ACM, 2496–2505.

[29] Ying Wei, Yu Zheng, and Qiang Yang. 2016. Transfer knowledge between cities. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1905–1914.

[30] Fei Wu, Hongjian Wang, and Zhenhui Li. 2016. Interpreting traffic dynamics using ubiquitous urban data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 69.

[31] Jianpeng Xu, Pang-Ning Tan, Lifeng Luo, and Jiayu Zhou. 2016. Gspartan: a geospatio-temporal multi-task learning framework for multi-location prediction. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 657–665.

[32] Jun Yang, Rong Yan, and Alexander G Hauptmann. 2007. Adapting SVM classifiers to data with shifted distributions. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE.

[33] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2019. Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. *AAAI Conference on Artificial Intelligence* (2019).

[34] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. *AAAI Conference on Artificial Intelligence* (2018).

[35] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. 2018. Deep Distributed Fusion Network for Air Quality Prediction. In *KDD*.

[36] Wei Ying, Yu Zhang, Junzhou Huang, and Qiang Yang. 2018. Transfer learning via learning to transfer. In *International Conference on Machine Learning*. 5072–5081.

[37] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.

[38] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[39] Rose Yu, Yaguang Li, Ugur Demiryurek, Cyrus Shahabi, and Yan Liu. 2017. Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. In *Proceedings of SIAM International Conference on Data Mining*.

[40] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. 2018. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. *arXiv preprint arXiv:1803.07294* (2018).

[41] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. *AAAI* (2017).

[42] Xiangyu Zhao and Jiliang Tang. 2017. Modeling temporal-spatial correlations for crime prediction. In *CIKM*. ACM, 497–506.

[43] Xiangyu Zhao, Tong Xu, Yanjie Fu, Enhong Chen, and Hao Guo. 2017. Incorporating Spatio-Temporal Smoothness for Air Quality Inference. In *ICDM*. IEEE, 1177–1182.

[44] Jiangchuan Zheng and Lionel M Ni. 2013. Time-dependent trajectory regression on road networks via multi-task learning. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 1048–1055.

[45] Ali Zonoozi, Jung-jae Kim, Xiao-Li Li, and Gao Cong. 2018. Periodic-CRN: A Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns.. In *IJCAI*. 3732–3738.